Learning Color Compatibility in Fashion Outfits

Heming Zhang\textsuperscript{1}, Xuewen Yang\textsuperscript{2}, Jianchao Tan\textsuperscript{3},
Chi-Hao Wu\textsuperscript{1}, Jue Wang\textsuperscript{4}, C.-C. Jay Kuo\textsuperscript{1}

\textsuperscript{1}University of Southern California
\textsuperscript{2}Stony Brook University
\textsuperscript{3}Kwai Inc. \textsuperscript{4}Megvii

Abstract. Color compatibility is important for evaluating the compatibility of a fashion outfit, yet it was neglected in previous studies. We bring this important problem to researchers’ attention and present a compatibility learning framework as solution to various fashion tasks. The framework consists of a novel way to model outfit compatibility and an innovative learning scheme. Specifically, we model the outfits as graphs and propose a novel graph construction to better utilize the power of graph neural networks. Then we utilize both ground-truth labels and pseudo labels to train the compatibility model in a weakly-supervised manner. Extensive experimental results verify the importance of color compatibility alone with the effectiveness of our framework. With color information alone, our model’s performance is already comparable to previous methods that use deep image features. Our full model combining the aforementioned contributions set the new state-of-the-art in fashion compatibility prediction.

1 Introduction

While many factors contribute to the compatibility of a set of fashion items, the color compatibility plays a pivotal role. When verifying whether a set of fashion items form a compatible outfit, the first check that comes to one’s mind is whether their colors are compatible or not. While each human body may fit to a different silhouette, color compatibility is more universal and applicable to a wider audience. Learning color compatibility for fashion will help us better identify compatible outfits and provide recommendation with maximum color synergy. Previous fashion studies largely neglected the importance of color compatibility. They either focus on the impact of each individual color \cite{1} or mix colors together with other information \cite{2}. Instead, we would like to bring the importance of color compatibility to audience’s attention and propose a novel solution to it.

The colors of an outfit can be compatible for different reasons, e.g., similar colors versus contrasting colors. Identifying what colors are compatible in the fashion industry will help us choose compatible colors for outfits. Therefore, instead of making a simple yes or no prediction on the compatibility of a given set of items, we would like to go one step further to learn the different templates of color compatibility in fashion. Existing methods are not necessarily
applicable to real world problems, especially to the fashion problems. The color compatibility templates developed by Matsuda et al. [3,4] as shown in Figure 1a were later adopted for various purposes, e.g., natural image color harmonization [5,6]. Nevertheless, O’Donovan et al. [7] noticed that this theory can deviate significantly from the color data collected from human users. Moreover, those methods for natural images do not specifically apply to fashion domain. Besides, the fashion trend is notoriously volatile and changes quickly with time. Instead of using hand-weaved rules on color, it would be more preferable to learn color compatibility templates directly from fashion data.

For the reasons above, we aim to learn compatibility in fashion outfits and apply it to fashion colors for the purpose of analyzing color compatibility templates in fashion problems. Our proposed data-driven approach for learning compatibility in fashion is illustrated in Figure 1b. Given fashion outfit data, three consecutive problems are studied using a compatibility model: 1) compatibility prediction 2) compatibility templates extraction 3) compatibility template-based recommendation. The compatibility model is the key to learn the compatibility from fashion outfits data. In our work, we propose a novel compatibility model, and an innovative learning scheme to train this model.

Fig. 1: (a) Hue templates proposed by Matsuda [3] are not designed specifically for fashion problems and deviated from real-world data. Figure modified from [6]. (b) We propose to learn color compatibility in fashion outfits from the data, which can be utilized to answer the compatibility related questions.

To better learn the compatibility of fashion outfits, we first model the outfits with graphs. Most previous works did not utilize graph models as they focus on learning the pairwise compatibility between two items rather than the compatibility of entire outfits [8,9,10,11,12]. They obtained the outfit compatibility by simple aggregation (e.g., averaging) over all the pairwise compatibility scores. However, the pairwise compatibility does not necessarily guaranty compatibility across a entire set of items. A few attempts have been made to model the entire outfits. A Recurrent Neural Network (RNN) was used to model the entire outfits in [12], which requires a predefined order to convert each set of items into a sequence.
A graph representation is more suitable in modelling of fashion outfits. Each graph is orderless ensemble of nodes and their interaction (i.e. edges), thus offering a natural abstraction of the compatibility among items of a fashion outfit. Although a graph model was used to model the entire dataset in [10], the outfit compatibility was still obtained from pairwise item compatibility. Graphs are used to model outfits in the work of Cui [13]. Both of them [10,13] took a straightforward approach of modelling each item as a node in the graphs and fed such graphs into Graph Neural Networks (GNN) or Graph Convolutional Networks (GCN). However, most GNNs/GCNs are designed for tasks such as node classification or recommendation [14][15][16], thus focus on node information aggregation. On the contrary, in the compatibility problem one should focus on the relation among items, which is represented by the edge information in the graphs in previous studies. Such mismatch leads to the poor results in [10,13], comparing to the works without graphs. Instead of modelling fashion items as nodes, we model each pairwise relation between items as a node to utilize the full power of GNN methods to aggregate the outfit compatibility from pairwise compatibilities. This novel graph construction is more suitable for compatibility learning using GNNs/GCNs.

Popular fashion datasets only provide binary compatibility information, i.e. compatible or not compatible. It is not a trivial task to learn distinct compatibility templates from such binary data. To learn compatibility templates from such data, we apply a clustering algorithm to the compatible outfits to segregate outfits with distinct compatibility templates. However, a simple application of some clustering operation to the compatibility templates may not yield the segregation meaningful for human interpretation we expected. Therefore, we propose to integrate the clustering process into the compatibility learning through an iterative process, which ensures that the obtained outfit clusters are meaningful for compatibility interpretation. The obtained compatibility model is also improved, as demonstrated in the experimental results.

To summarize, our contributions are three-fold:

1. We raise the important problem of color compatibility in fashion outfits, which was largely neglected in previous work in fashion. We take a data-driven approach, which overcomes the drawbacks of previous color compatibility studies in natural images and is more applicable to the fashion domain.

2. We propose a new way to model the compatibility of outfits. We utilize graphs to model the entire fashion outfits with a novel graph construction method that is more suitable for compatibility learning. Our compatibility model is not limited to color compatibility, but applies to various compatibility problems.

3. We propose an innovative learning scheme to integrate compatibility prediction and outfit clustering, which helps the interpretation of the fashion compatibility as well as improves the prediction performance.
2 Related Work

2.1 Color Compatibility

Most previous studies in color compatibility are not suitable for our problem. Color compatibility templates such as the hue templates proposed by Matsuda et al. in [3] and other variants were frequently used by later work [4][5]. However, experiments on large-scale color theme datasets [7] have demonstrated an obvious mismatch between the templates and real-world data. Instead, ODonovan et al. [7] proposed to learn a regression model for color compatibility using color theme data collected from human users. The limitation of this model is that it only evaluates fixed number of colors. Besides, those color themes are for general purposes and do not necessarily apply to fashion problems.

2.2 Fashion Outfit Compatibility

Most approaches only modelled pairwise compatibility between fashion items, whereas the compatibility of an entire outfit was either not considered or predicted as simple average of the pairwise compatibilities. Some previous work used the distances of item embeddings as the measure of pairwise compatibility [17][18][19][20][21] and [9] is a generalization of this approach. Furthermore, [22][23][24][25][26][27][10][28] directly modelled the pairwise compatibility. Jagadeesh et al. [29] used color representations as clues to retrieve fashion items that are matched to the query.

It is challenging to model the compatibility of entire outfits with various number of items. In [30], item features were fused by global pooling. Similarly, in [31], items are divided into six categories and the six features are concatenated. RNNs were utilized in [12][32], but a pre-defined order is required to model a set of items as a sequence. Although [10] directly modelled the pairwise compatibility, context information from other items were used. The entire dataset is modelled as a graph, where each node represents an item and nodes that appear in the same outfit are connected. During inference, the neighbors of each node contains items that appear in other outfits and provide rich context information. Such setting is different from previous work where each outfit is evaluated separately and no information from other outfits is provided. Experiments indicate that the method in [10] performs poorly in the common setting.

Closely related to our work, Cui [13] proposed to model each outfit as a graph where each node represents an item with a specific category. However, the proposed GNN model [13] learned on such graphs focuses more on item category information and neglected the relation among items. Instead, we propose a novel graph construction that is more suitable for the compatibility problem.

2.3 Deep Learning on Graphs

Graph neural networks (GNN) and graph convolutional networks (GCN) have been evolved over the years [33][34][35][36][14][37]. A common drawback of the ex-
isting models is that they only utilized the node features. In their work, edge features were either ignored or only the 1-D edge weights were used.

A few attempts were made to overcome this drawback. In [38], an edge-conditioned convolution was proposed where edge information was utilized to filter the nodes. Moreover, edge features were directly utilized for node feature aggregation functions in [39]. In these works, edge features are only considered as an auxiliary for node feature aggregations and do not get updated throughout the network inference. Monti et al. [40] proposed a Dual-Primal GCN framework, which alternated between the original graph (primal) and its line graph (dual). The dual convolutional layer produces features on the edges of the primal graph, and the primal edge features were used in the primal convolutional layer to produce primal node features. In [41], 2-D edge features defined as (dis)similarities between pairs of nodes were updated following each update of nodes. Five options for updating edge features were discussed in [42]. The edge features in these works are only used for node feature updates. For fashion compatibility, however, direct aggregation and update of edge features are desired.

3 Learning Color Compatibility in Fashion Outfits

The fashion compatibility prediction problem takes a set of fashion items \( x \) as inputs and a binary compatibility label \( y \) as the prediction target. For each fashion item, an image is provided, and additional information such as category annotations or text descriptions may also be available.

We formulate the compatibility template learning problem as a clustering problem. The compatible outfits are grouped into clusters, where each cluster obtained should identify an outfit compatibility template.

An overview of our color compatibility learning framework is shown in Figure 2. We first extract color features from the items, which will be described in Section 3.1. As introduced in Section 3.2, we then embed the outfit into an embedding feature, which will be used for compatibility prediction. We will explain our proposed learning scheme for the compatibility model in Section 3.3.

Fig. 2: An overview of our proposed method for learning color compatibility in fashion outfits.
3.1 Feature Extraction

For the problem of color compatibility learning, we propose to use a color palette, collection of dominant colors, as the color feature for an item. Previous work on fashion study usually use color histogram as features. Here we use the color palette instead for several reasons: First, our goal is to study the compatibility among colors, rather than the statistics of color distribution; Second, noises such as illumination variation, are encoded in the color histogram but color palette is more invariant to such variations; Third, color palettes are more abstractive, from which we can better learn the compatibility templates.

To extract a color palette from a fashion item image, we convert the image color space from RGB to Lab. In the Lab color space, the distance between two colors indicates their visual difference in humans perception. For each image, we apply the K-means algorithm to cluster the pixels with their Lab colors as features. Each cluster centroid represent a dominant color in the image and the cluster centroids together form the palette of the corresponding fashion item. The color feature of the image is obtained by concatenating the cluster centroids. In our work, we cluster the colors into 3 clusters. The resulting color feature has 9 elements (the details are in the supplemental material).

Although we use color features for the purpose of color compatibility learning, our proposed compatibility learning framework can be applied to other representations as well. In section 4, we demonstrate the effectiveness of our framework with other fashion item representations such as image features extracted from ResNet.

3.2 Outfit Embedding

Our outfit embedding consists of two steps. We first construct an outfit graph to model the given set of items, then embed the outfit graph into an outfit embedding.

As explained in previous sections, we would like to overcome the mismatch between outfit graphs and existing GNN models. Rather than designing a network structure specific to our problem, we choose to design a new graph construction method that fits to most existing models that aggregate the node information. Since the outfit compatibility is the global relation among outfit items, we propose to use nodes to represent pairwise item relations instead of individual items. Thus the global outfit relation can be aggregated from local pairwise relations.

For a given outfit of \( N \) items, we construct a graph \( G = (V, E) \) with \( \binom{N}{2} \) nodes where each node represents the pairwise relation between two items in the outfit. For each pair of nodes with shared item, we connect them with an edge. An illustration is shown in Figure 3. Comparing to the graph construction method in [13], our proposed graph can be viewed as its line graph (also known as the edge-to-node dual graph).

For the node \( \mathcal{V}_{(i,j)} \in V \) that represents the relation between the \( i \)th and the \( j \)th items, the node feature is obtained from features of these two items as

\[
f_{\mathcal{V}_{(i,j)}} = Z(f_i) \odot Z(f_j),
\]

(1)
Learning Color Compatibility in Fashion Outfits

Fig. 3: (a) The graph construction step by modelling each pairwise item relation as a node. The node features are obtained by embedding two item features. (b) The graph embedding step with GNN.

where $Z(\cdot)$ is an embedding module, $\odot$ denotes the Hadamard product and $f$ denotes the feature.

For an outfit graph constructed from an outfit, we then embed the outfit graph into an embedding of length $d$, where $d$ is a hyper-parameter. We can apply various models to aggregate the node information and embed the graph. In this work, we use GraphSage [14] for demonstration. At $l$-th layer, the hidden representation of node $V_{(i,j)}$ in $G$ is updated as

$$
\begin{align*}
    m^{l+1}_{V_{(i,j)}} &= \text{MEAN} \left( \{ h^l_{V'}, \forall V' \in \mathcal{N}(V_{(i,j)}) \} \right), \\
    h^{l+1}_{V_{(i,j)}} &= \sigma \left( W[m^l_{V_{(i,j)}}, m^{l+1}_{V_{(i,j)}}] + b \right),
\end{align*}
$$

(2)

where $\mathcal{N}(V_{(i,j)})$ is the neighborhood of $V_{(i,j)}$, $[\cdot,\cdot]$ denotes concatenation, $W$ is the weight matrix, $b$ is the bias term and $\sigma$ is a nonlinear activation function. We first obtain the new message $m^{l+1}_{V_{(i,j)}}$ by aggregating information from the neighbors of node $V_{(i,j)}$, then the updated hidden representation $h^{l+1}_{V_{(i,j)}}$ is obtained from the new message and the previous hidden representation of node $V_{(i,j)}$. The node features at the last layer are then globally pooled into a graph embedding.

Given the graph embedding, we can apply a fully connected layer and take a softmax to predict the compatibility from the outfit graph embedding. To train this layer together with the GNN model, we propose a novel learning scheme.

3.3 Joint Learning Scheme

There are different reasons for a set of colors to be considered compatible, e.g. similar colors or different colors with a synergy. To further understand such differences and explore compatibility templates, we cluster the compatible outfits
in the embedding space such that the outfits in the same cluster should be close in
the embedding space. Note that the incompatible samples are all simply assigned
to a single “incompatible” cluster. We do not care why a set is not compatible
so there is no need of finer clustering for them. (Detailed discussion of a single
incompatible cluster is in the supplemental material).

We expect each outfit cluster to represent a distinct and tangible reason for
the containing outfits to be considered compatible. Therefore, each cluster can
be interpreted as a compatibility template. However, it is not trivial to cluster
the outfits such that compatibility templates could be interpreted. If we first
train the embedding with outfit compatibility labels and then cluster the outfits
using their trained embeddings, the clusters obtained may not be meaningful
for compatibility prediction and thus may not help us to interpret compatibility
templates.

Therefore, we propose a joint learning scheme that brings the clustering pro-
cess into the training loop of the compatibility model. It is an iterative learning
scheme that alternatively generates pseudo labels from the outfit clusters and
updates the network with pseudo labels.

**Pseudo label generation.** Originally a sample $i$ contains the outfit input $x_i$
and a binary outfit compatibility label $y_i \in \{0, 1\}$. For each sample $i$, we generate
a multi-class pseudo label $\tilde{y}_i = c$, where $c \in \{0, 1, \cdots, C\}$ and $C$ is the number
of compatible clusters. The pseudo label generation procedure consists of four
steps and is illustrated in Figure 4.

1. **Collecting graph embeddings.** We collect the outfit embeddings of all
outfit samples in the training set using the current model.
2. Clustering. The current embeddings of outfits are clustered to $C + 1$ clusters and the clusters are indexed from 0 to C. For cluster $c \in \{0, 1, \cdots, C\}$, we count the number of compatible outfits $m_{c,pos}$ and the number of incompatible outfits $m_{c,neg}$ it contains. Then we calculate the compatible ratio as $r_c = m_{c,pos}/m_{c,neg}$ for each cluster $c$.

3. Assigning cluster labels. In this step, we assign a cluster label $z_c \in \{0, 1\}$ to cluster $c \in \{0, 1, \cdots, C\}$. These labels are only assigned to the clusters and do not represent for the outfit sample they contain. The obtained clusters are ranked by the ratios $r_c$ obtained in the previous step. Then the top $C$ clusters are assigned with cluster labels of 1, whereas the rest cluster is assigned with a cluster label of 0.

4. Generating pseudo sample labels. For each data sample $(x_i, y_i)$, we assign a pseudo multi-class label $\tilde{y}_i \in \{0, 1, \cdots, C\}$. The Euclidean distance between the $i$th sample and the centroid of cluster $c$ in the embedding is denoted as $d(i, c)$. The pseudo label is assigned as

$$\tilde{y}_i = \arg \min_c d(i, c) \text{ s.t. } y_i = z_c,$$

which is the index of the closest cluster that has the same label.

The unsupervised learning method DeepCluster [47] has a similar concept which iteratively clusters the features obtained from the network, and update the network using the clusters. In our method, we make use of the available compatibility labels and obtain the pseudo labels by fusing the ground-truth compatibility labels with the cluster assignments.

**Network update with pseudo labels.** We combine two loss functions to update the network with both the pseudo and ground-truth labels. For each compatible outfit, we randomly sample an incompatible outfit to pair with it.

The first loss is a global loss that is applied to entire outfits. We use the categorical cross-entropy loss to update the network parameters with pseudo labels:

$$l_1 = \sum_i [l_{CCE}([g(x_{i,pos})]_{\tilde{y}_{i,pos}}, \tilde{y}_{i,pos}) + l_{CCE}([g(x_{i,neg})]_{\tilde{y}_{i,neg}}, \tilde{y}_{i,neg})],$$

where $l_{CCE}(\cdot, \cdot)$ is the categorical cross-entropy loss, $g[\cdot]_c$ is the predicted softmax score for the class $c \in \{0, 1, \cdots, C\}$, and the subscriptions $i,pos$ and $i,neg$ denote the positive and the negative outfits of the $i$th pair of outfits, respectively.

Additionally, if a positive/negative pair of outfits share common items, we apply a local loss to subsets of the outfits. It helps the network learn more subtle differences between item relations. For each positive/negative pair of outfits that share items, we remove their common items from both outfits and obtain a pair of outfit subsets. Then for this pair of outfit subsets, we apply a binary cross-entropy loss with the compatibility labels:

$$l_2 = \sum_i [l_{BCE}(g(\bar{x}_{i,pos}), y_{i,pos}) + l_{BCE}(g(\bar{x}_{i,neg}), y_{i,neg})],$$

where $\bar{x} = x \setminus \text{common items}$.
where $l_{BCE}(\cdot, \cdot)$ denotes the binary cross-entropy loss and $\hat{x}$ denotes the subset of $x$ obtained by removing the common items. The intuition is that a subset of a compatible outfit should also be compatible, and a subset of an incompatible outfit is most likely to be incompatible.

The final loss function takes the weighted sum of $l_1$ and $l_2$ with a hyper-parameter $\lambda$

$$l = l_1 + \lambda l_2.$$  

(6)

More details of the loss functions are in the supplemental material.

4 Experimental Analysis

4.1 Datasets

Polyvore Outfits [8]. It contains 53,306, 5000, and 10,000 outfits for training, validation, and testing, respectively. It provides one image per fashion item and some items also come with text descriptions. Two tasks are provided for benchmarking. One task is the compatibility prediction (CP), in which the compatibility predicted for given outfits are evaluated using area-under-curve (AUC). The other task is the fill-in-the-blank (FITB), where one fashion item out of four candidates that fits a given outfit subset best is chosen and the evaluation metric is the accuracy.

Maryland Polyvore [12]. This dataset is similar to the Polyvore Outfits dataset, but much smaller in size. It has 17,316, 1,407, and 3076 outfits for training, validation and testing, respectively. It also has the two tasks (CP and FITB) for benchmarking. The test set provided by [12] is not challenging enough as it contains negative samples that can be easily identified using category information. We follow the previous studies [8,24] and evaluate our model on a more challenging test set proposed in [8].

4.2 Implementation Details

At the preprocessing step, we first separate fashion items from image backgrounds using [48]. Then we extract a 3-color palette from each fashion item.

We use one fully connected layer for node feature embedding and a 3-layer GraphSage [14] with ReLU activation and global max-pooling for graph embedding. The hidden representations are 60-D and the final embedding is 20-D. We utilize K-means with cosine similarity for clustering and re-assign the pseudo labels to outfits in every other 25 epochs of training. We tuned the hyperparameters and chose $C = 4$ and $\lambda = 0.5$ for the best performance. Detailed analysis can be found in the supplemental material.

Since our compatibility model also applies to other representations, we follow [8,24] and extract image features from ResNet18 [46], as well as text features from HGLMM Fisher vectors [49] of word2vec [50] in some experiments.
4.3 Comparison with Previous Work

We compare with the results of the following methods:

**Bi-LSTM** [12]. It models the outfits as sequences and the orders are predefined according to item categories. The image features are obtained from Inception V3 [51]. It utilizes the text descriptions as regularization during training.

**Siamese Network** [17]. It models the pairwise compatibility. We compare with its results reported in [8].

**Type-aware** [8]. It considers the category differences when learning the pairwise compatibility, which additionally requires the item category annotation. Same as Bi-LSTM, the text features are used as regularization during training.

**Outfit generation** [27]. It learns the pairwise compatibility. BERT features [52] are extracted from text descriptions and used for both training and testing.

**Context-aware** [10]. It models the entire dataset as a graph. Its best results are obtained by utilizing context information from other outfits, which is a different setting from most previous works. We only compare to its results obtained without extra context information from other outfits. Its image features are extracted from ResNet50 [46].

**NGNN** [13]. It models each outfit as an graph where each item is modelled as a node. Item category annotation is required for graph construction. The authors only reported the results on the easy test set of the small Maryland Polyvore dataset. We conduct the experiments using their original source code on the challenging test sets provided by [8].

**SCE-Net** [24]. It models pairwise compatibility and requires text features for both training and testing.

**CSA-Net** [28]. It models pairwise compatibility and its category-based subspace attention network requires item category annotation.

**Ours.** We extract color features as described in Section 3.1. To further verify the effectiveness of our compatibility system, we also utilize ResNet18 features and HGLMM features following most previous works. Our framework does not
Table 2: Comparison with previous methods when text information is additionally utilized.

| Methods         | Image / text representation | Needs category | Needs text in testing | P. Outfits | Maryland P. |
|-----------------|-----------------------------|----------------|-----------------------|------------|-------------|
|                 |                             |                |                       | CP         | FITB        |
| Bi-LSTM         | InceptionV3/HGLMM           | Yes            | -                     | 0.65       | 0.94        |
|                 |                             |                |                       | 39.7       | 64.9        |
| Type-aware      | ResNet18/HGLMM              | Yes            | -                     | 0.86       | 0.90        |
|                 |                             |                |                       | 55.3       | 61.0        |
| Outfit gen.     | ResNet18/BERT               | -              | Yes                   | 0.93       | 0.93        |
|                 |                             |                |                       | 66.1       | 69.9        |
| SCE-Net         | ResNet18/HGLMM              | -              | Yes                   | 0.91       | 0.90        |
|                 |                             |                |                       | 61.6       | 60.8        |
| Ours            | ResNet18/HGLMM              | -              | -                     | 0.90       | 0.91        |
|                 |                             |                |                       | 65.8       | 67.1        |

require item category annotation and the text features are only used for training as regularization.

The comparisons are conducted in two groups. In the first group, the images are used as inputs for both training and testing. The image representations used are either color palettes extracted using our method, or deep image features extracted from convolutional neural networks. Among the first group, Type-aware, NGNN and CSA-Net also require item categories as inputs. The results of the first group are listed in Table 1 With color palettes of 3 colors as image representation alone, our results already outperform most previous methods with deep features on the large-scale Polyvore Outfits dataset. This observation verifies our claim of the importance of color compatibility. Using the same deep features further boosts our results. Only the CSA-Net, which requires extra category annotation, outperforms ours. It demonstrates the effectiveness of the proposed compatibility learning method. The other graph method NGNN suffers from the bias on item categories. Its performance on the Maryland Polyvore drop 30%-40% on both tasks from the easy test set to the hard test set. Our graph method better utilizes individual item characteristics.

In the second group, the text descriptions are also used for the network training. Unlike Bi-LSTM, Type-aware and ours, Outfit gen. and SCE-Net also need text descriptions during testing. Similar conclusion can be drawn from the results of the second group listed in Table 2 When switching from the small-scale Maryland Polyvore dataset to the large-scale Polyvore Outfits dataset, the performances of Bi-LSTM and Type-aware drop dramatically, indicating potential overfitting. On the contrary, the proposed method gives consistent good results on both datasets. Although SCE-Net utilizes text description during inference, our method still shows comparable results on the CP task, while improving the FITB task by a significant margin of 4%-6%. Our results are only outperformed by Outfit gen., which uses a more powerful text features both for training and inference. Although the text descriptions provide more useful information during inference, it is less applicable to real scenarios where text descriptions are unavailable.
4.4 Ablation Studies

We compare the following modules as ablation studies to evaluate the contribution of each proposed module: NG denotes the node graph where each node represents an item in the outfit. LG denotes our proposed line graph where each node represents a pairwise relation between items. $l_{BCE}$ is the binary cross-entropy loss. $l_1$ and $l_2$ are the two losses proposed in (4) and (5), respectively. The ablation results are evaluated on the validation set of the large-scale Polyvore Outfits dataset and listed in Table 3. All methods compared use color palettes.

| Methods      | Compat AUC | FITB Acc |
|--------------|------------|----------|
| NG+$l_{BCE}$ | 0.76       | 42.0     |
| LG+$l_{BCE}$ | 0.84       | 52.7     |
| LG+$l_1$     | 0.84       | 54.7     |
| LG+$l_1$+$l_2$ | 0.84 | 58.8     |

Table 3: Ablations studies on how each proposed module contributes to the final performance. The experiments are conducted on the validation set of Polyvore Outfits with color palettes as fashion representations.

Each proposed module contributes to the final performance. Replacing the node graph with our proposed line graph, the overall performance is dramatically improved, supporting our argument that our graph better represents the fashion outfits for compatibility learning. By replacing the ground-truth binary labels to generated multi-class pseudo labels, the accuracy of the FITB task is increased by 2%. It shows that incorporating outfit clustering into the training loop not only provides us meaningful clusters but also helps the model better understand the compatibility. With the entire joint learning scheme, the accuracy of the FITB task is boosted by 6%. It verifies that our joint learning scheme helps the network learn the subtle difference of item relations.

4.5 Outfit Cluster Visualization

To gain more insights of our color compatibility model, we visualize the compatible clusters obtained from learning the fashion outfits data. For each outfit sample in the validation set of the Polyvore Outfits dataset, we obtain its predicted cluster using our model. Due to the limited space, only two clusters with 35 outfit samples per cluster are displayed in Figure 5. More visualization results are found in the supplemental material. For each cluster, two outfit samples are enlarged and displayed on the right.

Each cluster has its own color compatibility template. The visualization in Figure 5a suggests that the items in each outfit share similar colors. The cluster in Figure 5c demonstrates the combination of two major colors that are distinct.
(a) Cluster 1: The visualization suggests that the items of the outfits in this cluster have similar colors.

(b) Cluster 2: The visualization suggests that the outfits in this cluster have the combination of two major colors that distinct besides of the neutral colors.

Fig. 5: Compatible outfit clusters predicted using color palettes as fashion representation. Due to limited space, we only display 35 outfits per cluster (one outfit per column) and at most 10 items per outfit. More results can be found in supplementary materials. Two outfit samples in the bounding boxes are enlarged and shown on the right.

Fig. 6: A compatible outfit cluster predicted using deep image features that reveals a color template. The items in the same column belong to the same outfit. The visualization suggests that the items of the outfits in this cluster have similar color.

besides of the neutral colors with low chroma such as black or white. As illustrated via the visualization, our model does not simply group outfits together that share the similar colors, such as one cluster with green colors and the other with yellow colors. The outfits sharing similar color compatibility templates are grouped together through our model, which is the purpose of integrating the clustering process into the training loop.

It is not surprising that we can also find some color compatibility templates in the visualization of the model trained on deep image features. One example of
the clusters is shown in Figure 6 which displays the color compatibility template of similar colors. It again verifies our claim that color is an essential impact factor for outfit compatibility.

4.6 Fashion Recommendation based on Color compatibility Templates

While we can select one single overall best fit, as with the existing methods. Our innovative clustering method enables more versatile, customized recommendations from different compatibility templates. We made use of the outfit queries from the FITB task in the Polyvore Outfits dataset, and randomly sampled 2000 items as candidates. For each outfit query, we compare all the candidates and generate recommendation based on the cluster prediction. Figure 7 shows the recommendation from clusters 1 and 2. More results are in the supplemental material. For each outfit query, three recommendations per cluster are displayed. When the query outfit has similar colors, recommendation from cluster 1 also suggests a similar color, whereas cluster 2 suggests a distinct color. When the query outfit has mixed colors, cluster 1 matches one of those colors, whereas cluster 2 either matches one of those colors or suggests a neutral color.
5 Conclusion

In this paper, we addressed an important problem, namely learning color compatibility in fashion outfits. It plays an essential role in outfit compatibility, yet it has been neglected in previous work. We proposed a compatibility learning method including a novel compatibility model and an innovative learning scheme. Our approach provides more insights and possibility for customized recommendation. The proposed framework is not only suitable for color compatibility learning task, but also extendable to general outfit compatibility learning tasks. We conducted comprehensive experiments to demonstrate the importance of color compatibility, and the effectiveness of the proposed method.

References

1. Tangseng, P., Okatani, T.: Toward explainable fashion recommendation. arXiv preprint arXiv:1901.04870 (2019)
2. Yang, X., He, X., Wang, X., Ma, Y., Feng, F., Wang, M., Chua, T.S.: Interpretable fashion matching with rich attributes, SIGIR (2019)
3. Matsuda, Y.: Color design. Asakura Shoten 2 (1995) 10
4. Tokumaru, M., Muranaka, N., Imanishi, S.: Color design support system considering color harmony. In: 2002 IEEE World Congress on Computational Intelligence. 2002 IEEE International Conference on Fuzzy Systems. FUZZ-IEEE'02. Proceedings (Cat. No. 02CH37291). Volume 1., IEEE (2002) 378–383
5. Cohen-Or, D., Sorkine, O., Gal, R., Leyvand, T., Xu, Y.Q.: Color harmonization. ACM Transactions on Graphics (TOG) 25 (2006) 624–630
6. Tan, J., Echevarria, J.I., Gingold, Y.I.: Palette-based image decomposition, harmonization, and color transfer. CoRR abs/1804.01225 (2018)
7. O’Donovan, P., Agarwala, A., Hertzmann, A.: Color compatibility from large datasets. ACM Transactions on Graphics (TOG) 30 (2011) 63
8. Vasileva, M.I., Plummer, B.A., Dusad, K., Rajpal, S., Kumar, R., Forsyth, D.: Learning type-aware embeddings for fashion compatibility. In: Proceedings of the European Conference on Computer Vision (ECCV). (2018) 390–405
9. Yang, X., Ma, Y., Liao, L., Wang, M., Chua, T.S.: Transfcm: translation-based neural fashion compatibility modeling. In: Proceedings of the AAAI Conference on Artificial Intelligence. Volume 33. (2019) 403–410
10. Cucurull, G., Taslakian, P., Vazquez, D.: Context-aware visual compatibility prediction. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2019) 12617–12626
11. Lin, Y., Ren, P., Chen, Z., Ren, Z., Ma, J., De Rijke, M.: Explainable outfit recommendation with joint outfit matching and comment generation. IEEE Transactions on Knowledge and Data Engineering (2019)
12. Han, X., Wu, Z., Jiang, Y.G., Davis, L.S.: Learning fashion compatibility with bidirectional lstms. In: Proceedings of the 25th ACM international conference on Multimedia, ACM (2017) 1078–1086
13. Cui, Z., Li, Z., Wu, S., Zhang, X.Y., Wang, L.: Dressing as a whole: Outfit compatibility learning based on node-wise graph neural networks. In: The World Wide Web Conference, ACM (2019) 307–317
14. Hamilton, W., Ying, Z., Leskovec, J.: Inductive representation learning on large graphs. In: Advances in Neural Information Processing Systems. (2017) 1024–1034
Learning Color Compatibility in Fashion Outfits

15. Garcia, V., Bruna, J.: Few-shot learning with graph neural networks. In: Proceedings of the 6th International Conference on Learning Representations. (2018)

16. Ying, R., He, R., Chen, K., Eksombatchai, P., Hamilton, W.L., Leskovec, J.: Graph convolutional neural networks for web-scale recommender systems. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. (2018) 974–983

17. Veit, A., Kovacs, B., Bell, S., McAuley, J., Bala, K., Belongie, S.: Learning visual clothing style with heterogeneous dyadic co-occurrences. In: Proceedings of the IEEE International Conference on Computer Vision. (2015) 4642–4650

18. Chen, L., He, Y.: Dress fashionably: Learn fashion collocation with deep mixed-category metric learning. In: Thirty-Second AAAI Conference on Artificial Intelligence. (2018)

19. Kang, W.C., Kim, E., Leskovec, J., Rosenberg, C., McAuley, J.: Complete the look: Scene-based complementary product recommendation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2019) 10532–10541

20. Chen, W., Huang, P., Xu, J., Guo, X., Guo, C., Sun, F., Li, C., Pfadler, A., Zhao, H., Zhao, B.: Pog: Personalized outfit generation for fashion recommendation at alibaba ifashion. (2019)

21. Cui, Z., Li, Z., Wu, S., Zhang, X.Y., Wang, L.: Dressing as a whole: Outfit compatibility learning based on node-wise graph neural networks. In: The World Wide Web Conference, ACM (2019) 307–317

22. Iwata, T., Watanabe, S., Sawada, H.: Fashion coordinates recommender system using photographs from fashion magazines. In: Twenty-Second International Joint Conference on Artificial Intelligence. (2011)

23. Song, X., Feng, F., Liu, J., Li, Z., Nie, L., Ma, J.: Neurostylist: Neural compatibility modeling for clothing matching. In: Proceedings of the 25th ACM international conference on Multimedia, ACM (2017) 753–761

24. Tan, R., Vasileva, M.I., Saenko, K., Plummer, B.A.: Learning similarity conditions without explicit supervision. In: Proceedings of the IEEE International Conference on Computer Vision. (2019)

25. Polania, L.F., Gupte, S.: Learning fashion compatibility across apparel categories for outfit recommendation. In: 2019 IEEE International Conference on Image Processing (ICIP). (2019)

26. Feng, Z., Yu, Z., Yang, Y., Jing, Y., Jiang, J., Song, M.: Interpretable partitioned embedding for customized fashion outfit composition. In: Proceedings of the 2018 ACM on International Conference on Multimedia Retrieval (ICMR). (2018) 143–151

27. Li, K., Liu, C., Forsyth, D.: Coherent and controllable outfit generation. arXiv preprint arXiv:1906.07273 (2019)

28. Lin, Y.L., Tran, S., Davis, L.S.: Fashion outfit complementary item retrieval. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2020) 3311–3319

29. Jagadeesh, V., Piramuthu, R., Bhardwaj, A., Di, W., Sundaresan, N.: Large scale visual recommendations from street fashion images. In: Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining. (2014) 1925–1934

30. Li, Y., Cao, L., Zhu, J., Luo, J.: Mining fashion outfit composition using an end-to-end deep learning approach on set data. IEEE Transactions on Multimedia 19 (2017) 1946–1955
31. Tangseng, P., Yamaguchi, K., Okatani, T.: Recommending outfits from personal closet. In: Proceedings of the IEEE International Conference on Computer Vision. (2017) 2275–2279
32. Nakamura, T., Goto, R.: Outfit generation and style extraction via bidirectional lstm and autoencoder. arXiv preprint arXiv:1807.03133 (2018)
33. Bruna, J., Zaremba, W., Szlam, A., LeCun, Y.: Spectral networks and locally connected networks on graphs. In: Proceedings of the 3rd International Conference on Learning Representations. (2014)
34. Henaff, M., Bruna, J., LeCun, Y.: Deep convolutional networks on graph-structured data. arXiv preprint arXiv:1506.05163 (2015)
35. Defferrard, M., Bresson, X., Vandergheynst, P.: Convolutional neural networks on graphs with fast localized spectral filtering. In: Advances in neural information processing systems. (2016) 3844–3852
36. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907 (2016)
37. Velickovic, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., Bengio, Y.: Graph Attention Networks. International Conference on Learning Representations (2018) accepted as poster.
38. Simonovsky, M., Komodakis, N.: Dynamic edge-conditioned filters in convolutional neural networks on graphs. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2017) 3693–3702
39. Gong, L., Cheng, Q.: Exploiting edge features in graph neural networks. arXiv preprint arXiv:1809.02709 (2018)
40. Monti, F., Shchur, O., Bojchevski, A., Litany, O., Günemann, S., Bronstein, M.M.: Dual-primal graph convolutional networks. arXiv preprint arXiv:1806.00770 (2018)
41. Kim, J., Kim, T., Kim, S., Yoo, C.D.: Edge-labeling graph neural network for few-shot learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2019) 11–20
42. Wang, Y., Sun, Y., Liu, Z., Sarma, S.E., Bronstein, M.M., Solomon, J.M.: Dynamic graph cnn for learning on point clouds. ACM Transactions on Graphics (TOG) 38 (2019) 146
43. Yamaguchi, K., Hadi Kiapour, M., Berg, T.L.: Paper doll parsing: Retrieving similar styles to parse clothing items. In: Proceedings of the IEEE international conference on computer vision. (2013) 3519–3526
44. Hidayati, S.C., Hua, K.L., Cheng, W.H., Sun, S.W.: What are the fashion trends in new york? In: Proceedings of the 22nd ACM international conference on Multimedia, ACM (2014) 197–200
45. Hu, Y., Yi, X., Davis, L.S.: Collaborative fashion recommendation: A functional tensor factorization approach. In: Proceedings of the 23rd ACM international conference on Multimedia, ACM (2015) 129–138
46. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2016) 770–778
47. Caron, M., Bojanowski, P., Joulin, A., Douze, M.: Deep clustering for unsupervised learning of visual features. In: Proceedings of the European Conference on Computer Vision (ECCV). (2018) 132–149
48. Bradski, G.: The OpenCV Library. Dr. Dobb’s Journal of Software Tools (2000)
49. Klein, B., Lev, G., Sadeh, G., Wolf, L.: Associating neural word embeddings with deep image representations using fisher vectors. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2015) 4437–4446
50. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in neural information processing systems. (2013) 3111–3119

51. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2016) 2818–2826

52. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)