Metric-Aware Explainable Graph Network for Fashion Compatibility Recommendation

Jiali Duan
University of Southern California
jialidua@usc.edu

Xiaoyuan Guo
Emory University
xiaoyuan.guo@emory.edu

Son Tran
Amazon A9
sontran@amazon.com

C.-C. Jay Kuo
University of Southern California
cckuo@sipi.usc.edu

Abstract

In the task of fashion compatibility prediction, the goal is to pick an item from a candidate list to complement a partial outfit in the most appealing manner. Existing fashion compatibility recommendation work comprehends clothing images in a single metric space and lacks detailed understanding of users' preferences in different contexts. To address this problem, we propose a novel Metric-Aware Explainable Graph Network (MAEG). In MAEG, we leverage a Latent Semantic Extraction Network (LSEN) to obtain representations of items in the metric-aware latent semantic space. Then, we develop a graph filtering network and Pairwise Preference Attention (PPA) module to model the interactions between users' preferences and contextual information. With MAEG, we can provide recommendation to users as well as explain how each item and factor contribute to the final prediction. Extensive experiments on two large-scale real world datasets reveal that MAEG not only outperforms the state-of-the-art methods, but also provides interpretable insights by highlighting the role of semantic attributes and contextual relationships among items.

1. Introduction

The prevalence of online shopping has led to increasing demands for exploiting the user's preferences. Fashion compatibility prediction, being one form among them, requires to recommend the most "suitable" item to complete a partial outfit. For instance, an outfit with two pairs of shoes are clearly not suitable. Moreover, clothes usually consists of different attributes (e.g., color, shape, style, etc.) while the significance of each attribute usually changes as outfit changes. For example in Figure 1, the color of the sunglasses plays a crucial role when fitting it with the men's wear on the left while less so when it's paired with the women's wear on the right. The blue and pink ellipsoid indicates two semantic metric space in which items are compared.

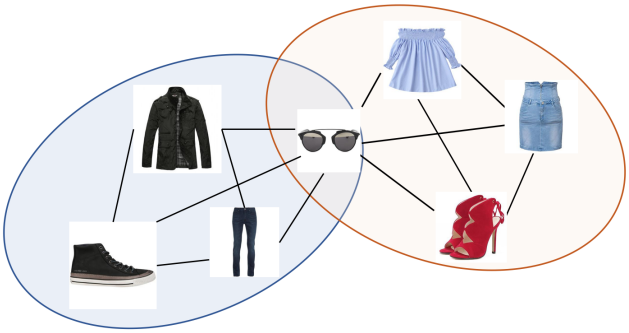

Figure 1. An illustration of interactions between a product's semantic attributes and its context: the blue and pink ellipsoids indicate two latent semantic metric space in which items are compared (e.g., color, shape, style, etc.). The sunglasses in the middle is shared by two different outfits for different reasons.

In previous works, the performance of fashion recommendation has been enhanced via various metric learning [25, 24, 21]. Veit *et al.* [25] adopted Siamese-network for comparing a pair of heterogeneous co-occurring instances and later proposed to leverage the triplet network [18] to account for different similarity conditions [24]. Vasileva et al. [22] attempted to get a better comparison between a pair of items by enumerating all possible combinations of category subspace. Although these models are able to perform multi-faceted pairwise comparisons, they fail to dive deep into exploring the interactions between clothing semantic aspects and their contexts.

In general, there are two challenges involved in current fashion compatibility recommendation. First, when choosing the most compatible item, people usually care both the role of each item in the entire outfit but also its detailed semantic attributes. For instance in Figure 1 we may ask our-

selves questions such as *"Does the sunglasses play a more important role than the bottom in outfit compatibility?"* and *"Given the black men's wear, is 'color' of the sunglasses more informative than its 'shape'?"* Second, clothing attributes are not static and usually evolve given the user's aesthetic tastes or change of social occasions (i.e., context). Moreover, it is difficult to collect fine-grained attribute annotations manually. These two aspects make it hard to generate explainable compatibility recommendations with current models.

To address the above challenges, we propose a novel Metric-Aware Explainable Graph Network (MAEG) for fashion compatibility recommendation. In MAEG, we introduce a factorized Latent Semantic Space, where each latent space represents a semantic attribute space in which items are compared. We then project items in this factorized latent semantic space to capture the user's fine-grained preferences and generate explainable compatibility recommendations. Specifically, we first develop a Latent Semantic Extraction Network (LSEN), which is used to extract latent metric-aware attribute representations in a weakly-supervised manner. Then to model the interactions between clothing semantic attributes and their contexts, we leverage a graph filtering network and design a Pairwise Preference Attention (PPA) module to automatically match the user's preference for each semantic attribute given contextual information, and aggregate all attributes with different weights. Finally, we optimize MAEG by predicting item linkage and overall outfit quality in a multi-task learning setting. Extensive experiments on two large-scale real world datasets reveal that MAEG not only outperforms existing state-of-the-art, but also provides interpretable insights by highlighting the role of semantic attributes and contextual relationships among items.

2. Related work

Generally, the related work can be grouped into the following categories: metric-aware recommendation, explainable recommendation and graph convolutional networks.

Metric-aware Recommendation. Deep metric learning aims to learn useful representations by distance comparisons [7, 27]. In [9, 25], Siamese network was leveraged to learn the notion of similarity between a pair of items. This framework allowed CNN to train in two parallel branches with the same 'copy' and joined by a contrastive loss function. To further explore the information contained in the clothing image, Veit et al. [24] proposed to learn different notions of similarity and developed a CSN network that encodes similarity conditions in different subspace. In [22], different subspace were constructed for each pair of clothing categories to perform metric learning and most recently Tan *et al.* [21] proposed to learn these similarity condition masks in a weak-supervised manner. Unfortunately, their representation ability is often limited by pairwise relationship and they fail to consider contextual information inherent in an outfit.

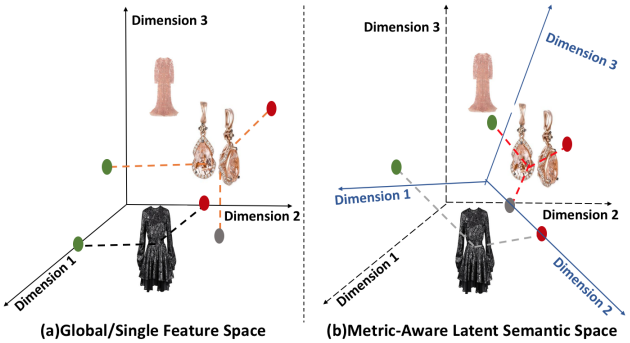

Figure 2. Difference between the conventional (a) Global/Single Feature Space and our (b) Metric-Aware Latent Semantic Space. Each latent space represents a semantic attribute space that is learned automatically from data. To make comparisons between items, different weights are assigned to these metric space to account for different preferences over corresponding attributes.

Explainable Recommendation Interpretability is one of the most important aspects of deep learning [29, 12, 5] and it has also become a very important research direction in fashion recommendation systems [16, 30]. McAuley et al. [16] made an interpretation about collaborative filtering by leveraging explicit factor based matrix factorization. Singh *et al.* [20] developed an attribute ranking module that utilizes a spatial transformer network to discover the most informative image region for the attribute. In [1, 8], a weakly supervised fashion recommendation model was built by localizing attribute regions in an image through Class Activation Maps (CAM) [31]. In this paper, we provide interpretable insights by adopting Grad-CAM [19] to highlight the most informative latent semantic attribute being attended to during feature aggregation.

Graph Convolutional Network Recently, a powerful structural learning paradigm, which takes the form of graph convolution, has shown great promise in handling complex relational reasoning [11, 14, 17]. Veličković *et al.* [26] presented a graph attention model that is able to discriminate neighboring nodes by computing attention weights between each pair of graph nodes. In contrast, we perform an outfit-level attention mechanism to explain the contribution of each item in the set with $O(N)$ complexity. Most recently, graph convolution have also been applied to recommendation systems [2, 28]. In [4], an outfit was represented as an undirected graph to consider context information and fashion compatibility prediction was cast as a link prediction problem. However, these work comprehended the clothing image as a global content representation and were unable to model the interactions between the user's preferences and contexts. In this paper, we take a further step to explain the user's preferences given varying contexts.

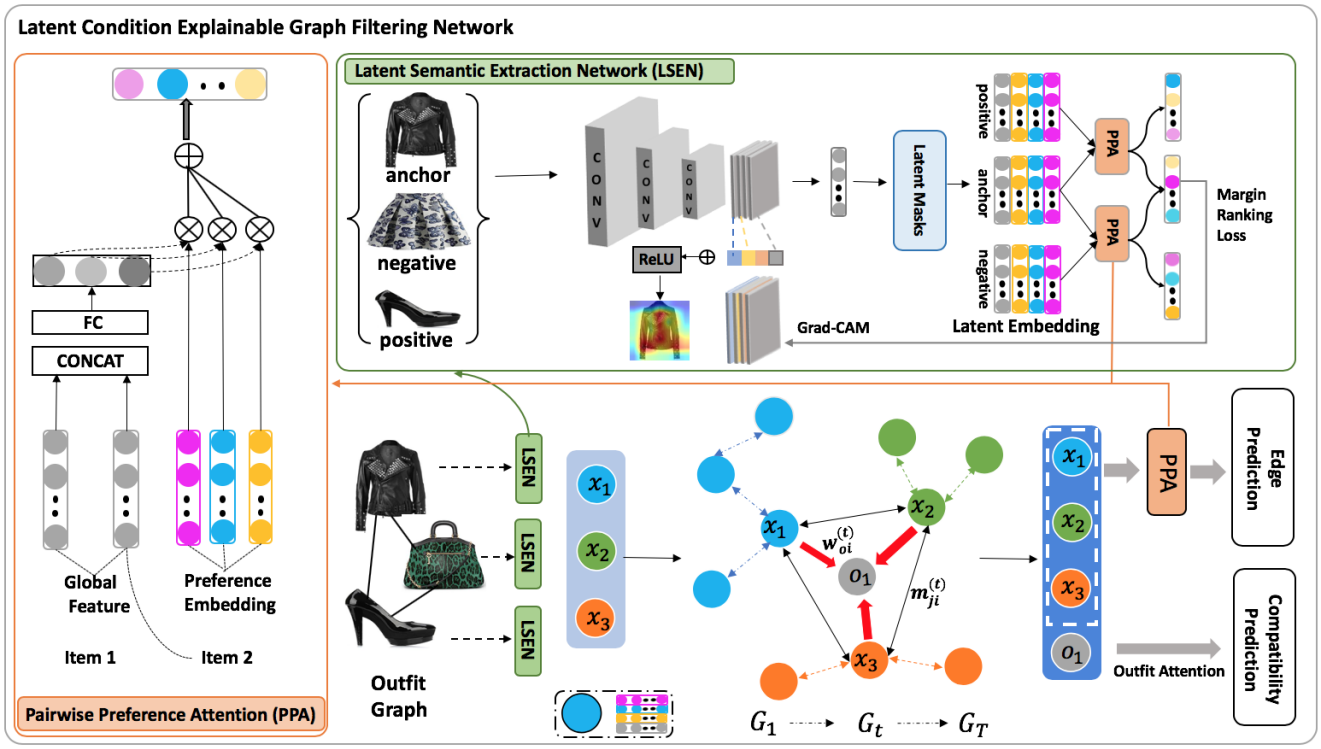

Figure 3. The architecture of Metric-Aware Explainable Graph Network (MAEG) for Fashion Compatibility Recommendation.

3. Metric-Aware Explainable Graph Networks

In this section, we introduce our proposed MAEG for addressing the fashion compatibility recommendation.

Most previous work represent clothing items in a global/single feature space as shown in Figure 2 (a), but comparisons can be ambiguous when context is present. For example, the gold dress and earring are close in terms of color whereas the two dresses are close in terms of category in Figure 2 (b). As a result, the incompatible gold earring and black dress are also close to each other. To solve this problem, MAEG utilizes a factorized Metric-Aware Latent Semantic Space, where each latent space represents a semantic attribute space in which items are being compared. As demonstrated in Figure 2 (b), items are represented jointly by several latent semantic space.

An overview of MAEG architecture is shown in Figure 3, which consists of two main components, i.e., Latent Semantic Extraction Network (LSEN) and Pairwise Preference Attention (PPA). With LSEN, we first obtain fashion item projections in the latent semantic feature space. Next we construct a graph representation for the outfit and perform message passing to consider contextual information. Then, we jointly train the item representation via latent semantic embedding and graph filtering network in a multi-task setting. Finally, with attribute preference inference, we can generate explainable recommendations.

3.1. Projecting Item into Latent Semantic Space

In this subsection, we describe how to project items into latent semantic space to obtain metric-aware semantic attributes in different subspace.

To get item embedding with respect to K different attributes, a standard supervised-learning framework requires annotating each item with these K attributes. Whereas, most real-world E-commerce datasets lack attribute annotations to learn semantic attribute representation directly. To this end, we resort to a weakly-supervised approach by learning from pseudo-labels similar to [13]. However, instead of minimizing conditional entropy of unlabeled class probabilities, we minimize the margin-ranking loss and recover internal metric in the latent space in a data-driven manner.

Specifically, we first construct triplets by mining the *Polyvore Outfits* [22]. Each triplet is in the form (a, p, n), where a and p are two compatible clothing items in outfit $\mathcal{I}_a^+$, and n is a mined negative from $\mathcal{I} \setminus \mathcal{I}_a^+$ but belonging to the same category as p. Then we use a CNN Φ to extract global feature representations for each triplet, yielding (x_a^g, x_p^g, x_n^g). Next we assign a pseudo-label l_k $(k = 1...K)$ to each triplet, by projecting its element to the corresponding semantic space:

$$x^{p_k} = \text{Proj}(x^g, E_k), \quad k = 1, 2, ..., K \tag{1}$$

where E_k is a learned parameter for the k-th semantic space. Here, each x^{p_k} is a semantic attribute representation of x

under space k. Consequently, we get the latent semantic feature representation extracted for item i by LSEN: $\hat{x} = [x_i^{p_1}; ...; x_i^{p_K}; x_i^g]$, which will then go through two branches, PPA (described below) and our Graph Filtering Network 3.2.

Actually, each x^{p_k} is a vector of dimension D residing in its corresponding latent semantic space and captures certain semantic attribute of the item. However, as discussed in introduction, a clothing item can have a variety of different attributes and each attribute may be given different priorities when paired with different objects. Therefore inspired by [23], we propose an attribute-level pairwise attention module (namely Pairwise Preference Attention (PPA)), which outputs a preference-aware feature representation for a pair of nodes by assigning adaptive weights to features in different latent semantic space. Concretely, as illustrated in left part of Figure 3, given a pair of inputs $(\hat{x}_i, \hat{x}_j)$, the attention weight for each latent semantic space is computed as:

$$\alpha_i = \text{softmax}(ReLU(W^T[x_i^g || x_j^g])) \tag{2}$$

where matrix $W \in R^{2D \times K}$ is a learned parameter to transfer the concatenated feature to dimension designated by the latent semantic space. The weighted preference embedding for item i can thus calculated as:

$$x_i^{wp} = \sum_{k=1}^{K} \alpha_{ik} x_i^{p_k} \tag{3}$$

Finally, to learn a useful metric for the latent semantic space, we adopt the Margin Ranking Loss [18], which has been widely used in metric learning:

$$\mathcal{L}_{\text{rank}} = \max(0, -(||x_a^{wp} - x_n^{wp}||_2 - ||x_a^{wp} - x_p^{wp}||_2) + \mu) \tag{4}$$

where μ is the maximum margin and $x_a^{wp}, x_p^{wp}, x_n^{wp}$ denote the weighted preference embedding for the anchor, positive and negative element in the triplet respectively.

3.2. Graph Filtering Network

In this subsection, we investigate how to take into account contextual information inherent in the outfit and how to model the interactions between the user's preference over semantic attributes and context.

To this end, we aim to output a context-aware representation $x_{i,T}^{ctx}$ for each object i conditioned on metric-aware latent semantic features associated with the object. This is obtained with iterative message passing over T iterations with our graph filtering network as shown in bottom part of Figure 3.

We use a fully-connected graph over the outfit, where each node represents a clothing item in the outfit and there's a directed edge $i \rightarrow j$ between every pair of item i and j. Each node i is represented by a metric-aware semantic feature $x_{i,t}^{ctx}$ that is updated during each iteration t.

Semantic Attribute-Conditioned Message Passing To consider contextual information in the factorized Latent Semantic Space at iteration $t+1$, our graph filtering takes three steps:

Step 1. We update the initial features for all nodes with the updated embedding parameter $M_{k,t}$ at time t ($k = 1, ..., K$) using Eqn 1. The resulting latent semantic attribute representation for node i is:

$$x_{i,t}^{ctx} = [x_{i,t}^{p_1}; ...; x_{i,t}^{p_K}; x_{i,t}^g], i = 1, ..., N \tag{5}$$

Step 2. For node i, we compute message vector $m_{j,i}^t$ from its neighbor j, by learning a function f_R parameterized by θ_R as:

$$m_{j,i}^t = f_R(x_{i,t}^{ctx}, x_{j,t}^{ctx}; \theta_R) \tag{6}$$

$$\theta_R = \begin{cases} \theta_0 & i = j \\ \theta_1 & i \neq j,\ 1 \leq i \leq N,\ j \in \mathcal{N}_i \end{cases} \tag{7}$$

where θ_0 and θ_1 are learned parameters. The idea is to differentiate between the effects from self-loop connections and neighboring connections.

Step 3. For node i, we gather messages propagated from its neighbors and update node representation by learning a function f_O parameterized by θ_O.

$$x_{i,t+1}^{ctx} = f_O(x_{i,t}^{ctx}, \sum_{j \in \mathcal{N}_i} m_{ji}^t; \theta_O) \tag{8}$$

Note that f_R and f_O are shared among all edges and all nodes, therefore generalizable to unseen data. Finally, we get final representation for each node $x_{i,T}^{ctx}, i = 1, 2, ..., N$, which can be used as input to subsequent Edge Prediction module 3.3.

Permutation-Invariant Outfit Embedding In deciding the quality of an outfit, a user usually places different emphasis on different items in the outfit. In fact, this provides an intuitive way to interpret the behavior of graph convolution. In order to capture this behavior, we propose to build an outfit embedding that is order-agnostic. This can be achieved without extra burden due to our contextualized graph representation. Specifically, we first initialize the outfit-embedding as $o_{\text{raw}} = \sum_{i=1}^{N} x_i$, then we perform dot-product attention to compute the affinity between it with each item i in the set:

$$\beta_i = \text{softmax}(< o_{\text{raw}}, x_{i,T}^g > / \sqrt{D}),\ \ i = 1, ..., N \tag{9}$$

where D is the dimension of x^g. The updated outfit-embedding can thus be computed as:

$$o = \sum_{i=1}^{N} \beta_i \cdot x_{i,T}^g \tag{10}$$

The computation complexity of this step is $O(N)$, where N is the number of objects in the outfit.

3.3. Multi-Task Learning

We collect context-aware node embedding with attribute information in different latent semantic space $x_i^{ctx}(i = 1, ..., N)$ and permutation-invariant outfit embedding $o_i(i = 1, ..., M)$ after T graph iterations for training. N is the number of nodes and M is the number of outfits and we ignore subscript T for brevity.

For edge prediction, given a pair of node representations (x_i^{ctx}, x_j^{ctx}), we apply Pairwise Preference Attention (PPA) introduced in 3.1, followed by a prediction function:

$$\hat{e}_{ij} = \mathcal{P}_e(\text{PPA}(x_i^{ctx}, x_j^{ctx})) \tag{11}$$

To train $\mathcal{P}_e$, we randomly sample equal number of positive and negative edges based on adjacency information and perform random edge-dropout with probability 0.15 at each iteration for robustness. Similarly, we learn an outfit grading function $\mathcal{P}_o$ with permutation-invariant outfit embedding as input:

$$\hat{s}_i = \mathcal{P}_o(o_i) \tag{12}$$

To train $\mathcal{P}_o$, we sample negatives by randomly replacing the groundtruth item at the blank position with one of the negative candidates from the same categories. The final loss is a weighted sum of two binary cross-entropy over edge and outfit:

$$\mathcal{L} = \mathcal{L}_\text{e} + \lambda_1 \mathcal{L}_\text{o} + \lambda_2 ||\Theta||^2 \tag{13}$$

where λ_1 controls the weight between the two losses and λ_2 is a regularization hyper-parameter. Θ includes all model parameters.

3.4. Attribute Preference Inference

We project clothing items into a new latent semantic space. In this space, the user's preferences for different semantic attributes can be calculated, making it possible to generate explainable recommendations. Specifically, when we calculate the preference over semantic attributes (Eqn 2), we can further identify which region of the image is being attended for this decision and where this particular semantic attribute gets activated.

Therefore, we can use the attention weight calculated in Eqn 2 as classification score for the K latent semantic space and compute the gradient of the score for class c with respect to the last convolutional layer, where c is determined by the class that maximizes the classification confidence, similar to [19]:

$$\gamma_t^c = \frac{1}{Z} \sum_m \sum_n \frac{y^c}{\partial F_{mn}^t} \tag{14}$$

where γ_t^c is the neuron importance weight of class c for channel t. F_{mn}^t indicates the (m, n) spatial location of the t-th channel of feature map F. Z is the normalization term. Then we perform a weighted combination of forward activation maps, followed by ReLU:

$$\text{Mask}^c = \text{ReLU}(\sum_t \gamma_t^c F^t) \tag{15}$$

where Mask^c indicates semantic attribute class c's contribution to the activation.

Accordingly, the explanation consists of three parts: (1) MAEG can highlight which part of the clothing image is being attended by the network during preference selection of semantic attributes; (2) MAEG provides the relative importance of each item in the outfit when making outfit compatibility prediction and compatibility score between each pair of items; (3) MAEG provides t-SNE visualization of the learned embedding. We will demonstrate these recommendation results in the experiment section.

4. Experiments

Datasets. *Maryland Polyvore* [6] is a real-world dataset created based on users' preferences of outfit configurations on an online website named *polyvore.com*: items within the outfits that receive high-ratings are considered compatible and vice versa. There are in total 164,379 items constituting 217,899 different outfits, where each outfit consists of 8 items at maximum and 6.5 items in average. In the original provided test set, negative candidates are sampled randomly without taking category into consideration (*e.g.*, the "shoe" may have already appeared in the given outfit, which makes exclusion of current candidate easier). We also notice that not all items are softlines (*e.g.*, there are occasionally "lamps" or "wardrobes" in the dataset that are hardlines). As such, we evaluate our model on resampled version of Maryland Polyvore as [4, 21].

Polyvore Outfits [22] is much larger than *Maryland Polyvore* with a total of 365,054 items and 68,306 outfits. Items that don't belong to clothing are discarded. Most importantly, items in the candidate list are hard-mined to belong from the same category, giving rise to more challenges. The maximum number of items per outfit is 19. We use the split provided by the authors and finally have 53,306 outfits for training, 10,000 for testing, and 5,000 for validation.

Evaluation Protocol. During inference, we use the edge prediction result for Fill In the Blank (FITB) and Outfit Compatibility Prediction (Compat AUC) as in [4]. The goal of FITB is to complete a given partial outfit $S = \{s_i\}(i = 1, ..., N)$ with the best item j from a candidate set T. We choose the item j^* that satisfies:

$$j^* = \underset{j}{\text{argmax}}(\sum_{i=1}^{N} e_{ij}), \ \ j = 1, ..., T \tag{16}$$

where e_{ij} is the edge prediction score between node i and node j. The evaluation metric for this task is the prediction

accuracy.

For evaluation outfit compatibility, a score close to 1 represents a compatible outfit while 0 an incompatible outfit. For an outfit k, we take the sum of two terms as the final prediction:

$$\frac{2}{N(N-1)} \sum_{\substack{i,j \\ (i \neq j)}} e_{ij}, \ \ i, j = 1, 2, ..., N \tag{17}$$

The evaluation metric for this task is the *Area Under the Roc Curve* (AUC) [3] to measure how much our model is capable of distinguishing between compatible and incompatible outfit.

Implementation Details. For fair comparisons, we follow the configurations specified in [22] and use ResNet-18 as global feature extractor Φ for all approaches, which yields a 64-dimensional feature. For [4], we use the default setting (k=0 for wo/ ctx and k=1 for w/ ctx case) and retrain it with 3 hidden layers of size 64 using ResNet-18 feature. For our MAEG framework, we use $K = 4$, $\mu = 0.3$ and $D = 64$ for our LSEN module and $T = 3$ for our graph filtering network, with hidden size of 64 each. We use dropout ratio of 0.5 and batch normalization between each graph layer for regularization. We use Adam [10] with a learning rate of 0.001 with $10,000$ iterations for optimization and $\lambda_1 = 0.5$, $\lambda_2 = 1e-5$. We perform model-parallelism on two RTX 2080 for training.

5. Results

In this section, we perform comprehensive analysis of our proposed MAEG. First, we evaluate our model on two large-scale real world datasets against state-of-the-art methods from the latest papers in Section 5.1. Then, we conduct extensive ablation study to investigate how each component of our framework affects the overall performance in Section 5.2. To do so, we dive deep into details of our two core modules LSEN and PPA, to compare them with their variants in Section 5.3 and Section 5.4 respectively. Finally, we interprete our model and show qualitative results in Section 5.5.

5.1. Recommendation Performance

We present the recommendation performance for Fill In The Blank (FITB) and Outfit Compatibility Prediction (Compat AUC) on two datasets in Table 1 and Table 2 respectively. As can be seen, our model obtains consistent improvement in both tasks across two datasets. In particular, our model outperforms the state-of-the-art graph-convolution based approach CA-GCN [4] by over 5% under with-context case in FITB task, demonstrating that it can better differentiate candidate items by utilizing our proposed latent semantic space. In fact, the significance of our latent semantic representation is more obvious under the without-context case, in which our model achieves a gain of over 19% against [4]. In addition, when compared with strongly-supervised metric-learning approaches such as [22, 25], which require category label information during training, our model learns metric aware representation with pseudo-label but achieves comparable performance when no context is used and much higher performance (over 24% on both datasets) when context is used. *To the best of our knowledge, we made the first attempt to demonstrate the effectiveness of leveraging metric-aware latent embedding in graph neural networks.*

Method	FITB Acc	Compat. AUC
Siamese Net [25]	54.4%	0.85
Bi-LSTM [6]	64.9%	0.94
TA-CSN [22]	65.0%	0.93
SCE-Net [21]	60.8%	0.90
CA-GCN (wo /ctx) [4]	41.7%	0.71
CA-GCN (w /ctx) [4]	83.1%	0.99
Ours (wo/ ctx)	62.1%	0.93
Ours (w/ ctx)	87.3%	0.99
Ours + Outfit(w/ ctx)	**89.3%**	**0.99**

Table 1. Comparisons on FITB/Compatibility task over Resampled *Maryland Polyvore*.

Method	FITB Acc	Compat. AUC
Siamese Net [25]	52.9%	0.81
TA-CSN [22]	55.3%	0.86
SCE-Net [21]	61.6%	0.91
CA-GCN (wo/ ctx) [4]	43.3%	0.75
CA-GCN (w/ ctx) [4]	82.4%	0.99
Ours (wo/ ctx)	63.1%	0.93
Ours (w/ ctx)	86.7%	0.99
Ours+Outfit (w/ ctx)	**88.0%**	**0.99**

Table 2. Comparisons on FITB/Compatibility task over *Polyvore Outfits*.

5.2. Ablation Study

In this subsection, we perform ablation study on our proposed MAEG. As shown in Table 3, we report the FITB Acc and Compat AUC on *Polyvore Outfits* [22] for different combinations of our components.

For model #1, we use graph filtering network, but with 4 random subspace (without LSEN). In this case, there is no guarantee that the four subspace can provide useful metric measure for feature comparison. The performance is only around 37.7% for FITB and 0.69 for Compat AUC. The situation is the same for model #2, with outfit-embedding as supervision. In model #3 and 4, LSEN is used to learn useful metric-aware latent semantic space while average-weighting mechanism across latent subspace is in

Model Version	LSEN	PPA	Graph Filtering	Outfit Embedding	ACC ↑	AUC ↑
1			✓		37.7	0.69
2			✓	✓	38.8	0.70
3	✓		✓		57.0	0.84
4	✓		✓	✓	57.7	0.87
5	✓	✓	✓		86.7	0.99
6	✓	✓	✓	✓	88.0	0.99

Table 3. Ablation study of our model on *Polyvore Outfits* [22].

place of PPA. In this case, compared with model #2, the performance of model #4 increases from 38.8% to 57.7% for FITB and from 0.70 to 0.87 for Compat AUC. Therefore, we conclude that LSEN serves as a crucial component for providing useful metric representation for downstream components such as graph filtering network or PPA. To demonstrate the effectiveness of PPA component, we compare model #3 with #5, where the differences lie at how we perform feature selection during triplet-training and graph aggregation. As can be seen, the performance gap due to PPA module is around 29% for FITB and 0.15 for Compat AUC. One possible explanation is that adaptive feature selection provides our framework with more modelling capacity whereas average aggregation of feature from different semantic space may impose difficulty for graph filtering network in learning useful node representations. Finally, model #6 shows extra 1.3% improvement for FITB by leveraging outfit-embedding component. It also helps interpret the model's emphasis on different items when making compatibility predictions.

With Variants of **Graph Filtering Network** (wo/w ctx)		
Method	Compat AUC	FITB ACC
1 hidden layer	0.89/0.84	59.6/57.7
2 hidden layer	0.90/0.94	61.2/72.6
3 hidden layer	0.93/0.99	63.1/88.0

Table 4. Variants of T for Graph Filtering Network on *Polyvore Outfits*.

We further study the influence of propagation step T on the recommendation performance. All hidden layers have size of 64 since the feature dimension from ResNet-18 extractor is 64. From Table 4, we observe consistent improvement of performance by increasing hidden layer number from 1 to 3. We notice that when there's only 1 hidden layer, the modeling capacity seems limited when context is involved. Instead, two or more are better options as they increase the modelling capacity.

5.3. Variants of LSEN

In this subsection, we discuss the necessity of using Latent Semantic Embedding extracted by LSEN. We construct

With Variants of **LSEN** (wo/w ctx)		
Method	Compat AUC	FITB ACC
Random 4-Subspace	0.52/0.70	30.4/38.8
Siamese-Net [25]	0.80/0.93	52.0/69.9
LSEN w/ (1 subspace)	0.85/0.95	54.3/72.4
LSEN w/ (3 subspace)	0.92/0.99	62.4/87.0
LSEN w/ static M_k	0.92/0.99	61.8/77.0
LSEN (Ours)	**0.93/0.99**	**63.1/88.0**

Table 5. Variants of Latent Semantic Extraction Network (LSEN) on *Polyvore Outfits*.

6 variants based on Siamese-Net [25] and variants of LSEN to verify the effectiveness of incorporating the metric-aware latent semantic space. As shown in Table 5, four random subspace without metric learning can only achieve 38.8% FITB and 0.70 Compat AUC. In contrast, a pre-trained Simaeset-Net [25] embedding bring the number to 69.9% and 0.93 for FITB and Compat AUC respectively. We also trained with variants of LSEN, by experimenting with different numbers of latent semantic space or by fixing latent embedding layer (static M_k) in LSEN. The result shows that 4 subspace yields slightly better performance than using 3 subspace (88.0% vs 87.0% in FITB) whereas much better than using 1 subspace alone (88.0% vs 72.4% in FITB Acc and 0.99 vs 0.55 in Compat AUC). We further increase the performance under w/ ctx case by keeping latent embedding layer M_k updated during training.

5.4. Variants of PPA

In this subsection, we investigate two alternatives for performing feature preference selection. In Table 6, we compare PPA with average-weighting mechanism and direct concatenation. The former indiscriminatively blends features from different subspace with feature dimension unchanged while the latter also yields a single feature but with longer dimension. Interestingly, they give almost the same performance under without-context situation, but training with average-weighting mechanism cannot further increase the performance when context is involved during inference. Therefore, we argue that it is important to perform careful feature selection during graph aggregation step, which would then influence the graph update step. The evaluation result shows the effectiveness of using PPA, by performing adaptive feature selection based on contextual information.

With Variants of **Preference Selection** (wo/w ctx)		
Method	Compat AUC	FITB
Average Weight	0.87/0.87	58.0/57.7
Concat	0.87/0.91	58.7/61.2
PPA (Ours)	**0.93/0.99**	**63.1/88.0**

Table 6. Necessity of Pairwise Preference Attention: Results on the *Polyvore-Outfit* test set obtained by variants of Preference Selection.

5.5. Interpretability of Our Model

In order to better understand the recommendation results, we provide interpretation of our model from different perspectives. First, we provide t-SNE [15] visualization for one of our learned latent semantic space on *Polyvore Outfits* [22], which involves items from different categories. Ideally, clothing items that are compatible in color, style, shape, etc will be embed close to each other. As shown in Figure 4, we can observe gradual transition in terms of color across categories, which demonstrates that even with only weak supervision during training, our approach is able to learn visually similar conditions explicitly defined in the dataset.

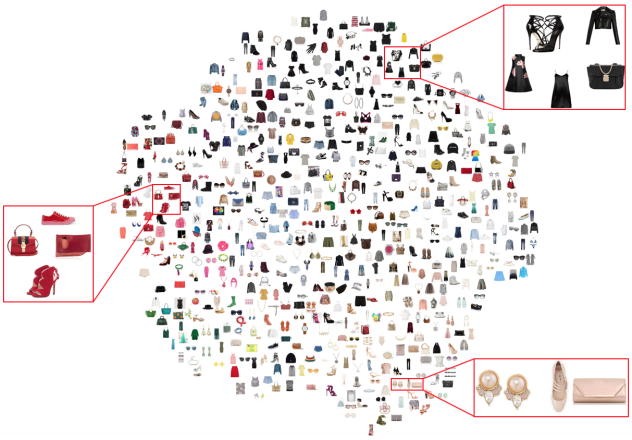

Figure 4. An example of t-SNE visualization of our Latent Semantic Space on *Polyvore Outfits*.

Second, we use Grad-CAM [19] to understand how our network makes its preference decision over latent semantic attributes. The idea is to use the gradient of the maximum preference score with respect to the global average pooling layer in LSEN to visualize which parts of the clothing image are most important for the preference selection. Visualization examples from four representative categories are presented in Figure 5.

Figure 5. Grad-CAM visualization on decision making of Pairwise Preference Attention (PPA) module.

Third, through permutation-invariant outfit embedding described in Section 3.2, we are able to provide some human interpretable results for our model's emphasis when making outfit compatibility predictions. As shown in Figure 6, items that are more conspicuous seem to get more credits for the final "contributions", which to some extent coincides with human intuition.

Figure 6. Distribution of attention weight for item in the outfit.

Finally, we provide qualitative recommendation results for FITB using our model in Figure 7. More results are available in the supplementary material.

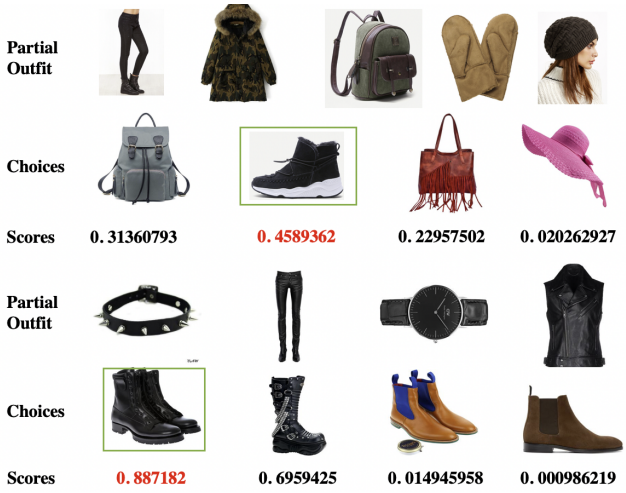

Figure 7. Qualitative results of our model for FITB prediction on *Maryland Polyvore* and *Polyvore Outfits*. Green box indicates the groundtruth and scores highlighted with red color are our predictions using Eqn 16.

6. Conclusion

In this paper, we proposed MAEG to capture how users' preferences evolve given changes of contexts and provide interpretability for fashion compatibility recommendation tasks. We first developed a Latent Semantic Extraction Network (LSEN) to project items under different semantic attribute space that's learned automatically from data. Then we introduced Pairwise Preference Attention (PPA) and Graph Filtering Network to comprehend the interactions between user's preference and context. Finally, we provide interpretable visualizations of our framework. Experimental results on real-world datasets clearly demonstrated the effectiveness and explanatory power of MAEG.

References

[1] Kenan E Ak, Ashraf A Kassim, Joo Hwee Lim, and Jo Yew Tham. Learning attribute representations with localization for flexible fashion search. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7708–7717, 2018. 2

[2] Rianne van den Berg, Thomas N Kipf, and Max Welling. Graph convolutional matrix completion. *arXiv preprint arXiv:1706.02263*, 2017. 2

[3] Andrew P Bradley. The use of the area under the roc curve in the evaluation of machine learning algorithms. *Pattern recognition*, 30(7):1145–1159, 1997. 6

[4] Guillem Cucurull, Perouz Taslakian, and David Vazquez. Context-aware visual compatibility prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 12617–12626, 2019. 2, 5, 6

[5] Jiali Duan, Jun Wan, Shuai Zhou, Xiaoyuan Guo, and Stan Z Li. A unified framework for multi-modal isolated gesture recognition. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 14(1s):21, 2018. 2

[6] 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*, pages 1078–1086. ACM, 2017. 5, 6

[7] Elad Hoffer and Nir Ailon. Deep metric learning using triplet network. In *International Workshop on Similarity-Based Pattern Recognition*, pages 84–92. Springer, 2015. 2

[8] Min Hou, Le Wu, Enhong Chen, Zhi Li, Vincent W Zheng, and Qi Liu. Explainable fashion recommendation: A semantic attribute region guided approach. *arXiv preprint arXiv:1905.12862*, 2019. 2

[9] Wang-Cheng Kang, Chen Fang, Zhaowen Wang, and Julian McAuley. Visually-aware fashion recommendation and design with generative image models. In *2017 IEEE International Conference on Data Mining (ICDM)*, pages 207–216. IEEE, 2017. 2

[10] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

[11] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016. 2

[12] C-C Jay Kuo, Min Zhang, Siyang Li, Jiali Duan, and Yueru Chen. Interpretable convolutional neural networks via feedforward design. *Journal of Visual Communication and Image Representation*, 60:346–359, 2019. 2

[13] Dong-Hyun Lee. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on Challenges in Representation Learning, ICML*, volume 3, page 2, 2013. 3

[14] Yujia Li, Daniel Tarlow, Marc Brockschmidt, and Richard Zemel. Gated graph sequence neural networks. *arXiv preprint arXiv:1511.05493*, 2015. 2

[15] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008. 8

[16] Julian McAuley and Jure Leskovec. Hidden factors and hidden topics: understanding rating dimensions with review text. In *Proceedings of the 7th ACM conference on Recommender systems*, pages 165–172. ACM, 2013. 2

[17] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1):61–80, 2008. 2

[18] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823, 2015. 1, 4

[19] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 618–626, 2017. 2, 5, 8

[20] Krishna Kumar Singh and Yong Jae Lee. End-to-end localization and ranking for relative attributes. In *European Conference on Computer Vision*, pages 753–769. Springer, 2016. 2

[21] Reuben Tan, Mariya I Vasileva, Kate Saenko, and Bryan A Plummer. Learning similarity conditions without explicit supervision. *arXiv preprint arXiv:1908.08589*, 2019. 1, 2, 5, 6

[22] Mariya I Vasileva, Bryan A Plummer, Krishna Dusad, Shreya Rajpal, Ranjitha Kumar, and David Forsyth. Learning type-aware embeddings for fashion compatibility. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 390–405, 2018. 1, 2, 3, 5, 6, 7, 8

[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. 4

[24] Andreas Veit, Serge Belongie, and Theofanis Karaletsos. Conditional similarity networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 830–838, 2017. 1, 2

[25] Andreas Veit, Balazs Kovacs, Sean Bell, Julian McAuley, Kavita Bala, and Serge Belongie. Learning visual clothing style with heterogeneous dyadic co-occurrences. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4642–4650, 2015. 1, 2, 6, 7

[26] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017. 2

[27] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. In *Advances in neural information processing systems*, pages 3630–3638, 2016. 2

[28] Rex Ying, Ruining He, Kaifeng Chen, Pong Eksombatchai, William L Hamilton, and Jure Leskovec. Graph convo-

lutional neural networks for web-scale recommender systems. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 974–983. ACM, 2018. 2

[29] Quanshi Zhang, Ying Nian Wu, and Song-Chun Zhu. Interpretable convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8827–8836, 2018. 2

[30] Yongfeng Zhang, Guokun Lai, Min Zhang, Yi Zhang, Yiqun Liu, and Shaoping Ma. Explicit factor models for explainable recommendation based on phrase-level sentiment analysis. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*, pages 83–92. ACM, 2014. 2

[31] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2921–2929, 2016. 2