Interpretable Fashion Matching with Rich Attributes

Xun Yang
National University of Singapore
xunyang@nus.edu.sg

Xiangnan He
University of Science and Technology of China
xiangnanhe@gmail.com

Xiang Wang
National University of Singapore
xiangwang1223@gmail.com

Yunshan Ma
Fuli Feng
National University of Singapore
yunshan.ma@u.nus.edu
fulifeng93@gmail.com

Meng Wang
Department of Computer Science
Hefei University of Technology
eric.mengwang@gmail.com

Tat-Seng Chua
National University of Singapore
dcscts@nus.edu.sg

ABSTRACT

Understanding the mix-and-match relationships of fashion items receives increasing attention in the fashion industry. Existing methods have primarily utilized the visual content to learn the compatibility and performed matching in the latent visual space. Despite their effectiveness, these methods work like a black box and cannot reveal the reasons that two items match well. The rich attributes associated with fashion items, e.g., *off-shoulder dress* and *black skinny jean*, which describe the **semantics** of items in a human-interpretable way, have been largely ignored. In this work, we address the interpretable fashion matching task, aiming to inject interpretability into the compatibility modeling of fashion items. Specifically, given a corpus of matched pairs of items, we not only can predict the compatibility score of unseen pairs, but also learn interpretable patterns that lead to a good match, e.g., *white T-shirt* matches with *black trouser*. We propose a new solution named **A**ttribute-based **I**nterpretable **C**ompatibility (AIC) method, which consists of three modules: 1) a tree-based module that extracts decision rules on matching prediction, 2) an embedding module that learns vector representation for a rule by accounting for the attribute semantics in the rule, and 3) a joint modeling module that unifies the visual embedding and rule embedding to predict the matching score. To justify our proposal, we contribute a new Lookastic dataset with fashion attributes available. Extensive experiments show that AIC not only outperforms several state-of-the-art methods, but also provides reasonable interpretability on matching decisions.

CCS CONCEPTS

• **Information systems** → *Specialized information retrieval.*

KEYWORDS

Multimedia recommendation, Clothing matching, Fashion compatibility learning

SIGIR '19, June 21–25, 2019, Paris, France

ACM ISBN 978-1-4503-9999-9/18/06…$15.00
https://doi.org/10.1145/1122445.1122456

ACM Reference Format:
Xun Yang, Xiangnan He, Xiang Wang, Yunshan Ma, Fuli Feng, Meng Wang, and Tat-Seng Chua. 2019. Interpretable Fashion Matching with Rich Attributes. In *SIGIR '19: The 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, June 21–25, 2019, Paris, France.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION

Fashion is a rapidly growing industry, which has motivated various research topics in the fashion domain, such as recommendation [38, 39], search [23], and dialogue systems [22], etc. In this paper, we focus on a newly-emerged topic of *Mix-and-match*-based fashion recommendation [11, 20, 28–30, 35], for which the goal is to predict the compatibility between fashion items. For example, when a user views/buys an item (e.g., a red floral maxi dress), the system matches it with the compatible fashion items from a complementary category (e.g., high-heel sandals). The key to solving this problem is how to effectively model the item-item compatibility relationships.

Existing methods have primarily leveraged the images of fashion items to model the notion of visual compatibility and performed matching in a latent visual space [5, 14, 24, 29, 31]. A common assumption is that a pair of compatible items should stay close with each other in the latent space . Then, the matching problem is solved under a metric learning paradigm: first collect a corpus of matched/unmatched item pairs, and then train a parameterized similarity function that enforces matched pairs have a higher similarity score than unmatched pairs. Despite their effectiveness, existing methods mainly exploit the visual information that comprises of low-level signals, forgo modeling the rich attributes associated with fashion items, e.g., *off-shoulder dress* and *black skinny jean*. They just work like a black box and cannot interpret the reasons that two items match well, being insufficient to support downstream applications. We argue that the rich attributes, which describe the **semantics** of items in a human-interpretable way, should be carefully taken into account to improve both the matching accuracy and interpretability.

Recent works have tried to alleviate the above-mentioned limitations by augmenting the visual features of items with textual descriptions [29], or refining pairwise visual compatibility with category-category complementary relationships [30, 35]. However, the textual description of items is directly encoded as a dense vector without language parsing, making it hard to reveal which attributes contribute most to a match. The category-category relationships only use coarse-grained categories to bridge two items from complementary categories, which results in limited interpretability. In summary, the semantics of rich attributes associated with fashion items have not been fully explored in fashion matching.

This paper addresses the *interpretable fashion matching* task, which is a new topic in this field. Our aim is to inject *interpretability* into the compatibility modeling of fashion items by leveraging the rich fashion attributes. Specifically, given a corpus of matched pairs of items, we learn the interpretable matching patterns that lead to a good match, e.g., *white T-shirt* matches with *black trouser*, which is termed as *attribute cross* (analogous to feature cross [7]) in this work. Towards this end, we propose a new solution named **A**ttribute-based **I**nterpretable **C**ompatibility (AIC) method, which discovers informative attribute crosses in an explicit and interpretable way. Specifically, we first automatically extract decision rules on matching prediction by using a decision tree method. Then, we design an embedding module to explicitly learn the vector representation for each rule by preserving the semantics of attributes in the rule. We further propose a joint modeling module that unifies the visual embedding and attribute-based rule embedding to predict the matching score. To enhance the interpretability, we design an attention network to select the most informative matching patterns, making the overall prediction process easy-to-interpret. To the best of our knowledge, this is the first time to develop an interpretable fashion matching framework that can explicitly learn attributed-based matching patterns.

Our contributions are summarized as follows.

- We present an attribute-based interpretable compatibility framework that not only can predict the compatibility score of unseen pairs, but also learn interpretable matching patterns that lead to a good match.
- We propose to capture the semantics of decision rules by modeling attribute interaction, and unify the strengths of visual embedding and attribute-based rule embedding.
- We contribute a dataset with fashion attributes available to justify the effectiveness of AIC on interpretable fashion matching. Extensive experiments show that AIC not only outperforms several state-of-the-art methods, but also provides reasonable interpretability on matching decisions.

2 PROBLEM FORMULATION

Given a corpus of fashion items $\mathcal{X} = \{x_i\}_i^{|\mathcal{X}|}$, and the binary pair labels $\mathcal{Y} = \{y_{ij}\}$, defined by

$$y_{ij} = \begin{cases} 1 & \text{if}(x_i, x_j) \in C, \\ 0 & \text{Otherwise,} \end{cases} \tag{1}$$

where C denotes the pairwise compatibility relationship (i.e., if x_i is compatible with x_j, then $y_{ij} = 1$), the basic goal of fashion matching is to build a predictive model that estimates the compatibility score between x_i and x_j:

$$\hat{y}_{ij} = f(x_i, x_j), \tag{2}$$

where f denotes the predictive model, and $\hat{y}_{ij}$ denotes the predicted compatibility score of a pair of items.

Figure 1: An illustration of the *mix-and-match* relationship (Left) and rich fashion attributes associated with fashion items (Right). Fashion items are usually described by a diverse set of attributes that carry rich semantics of items, which have been largely ignored by existing fashion matching methods.

Traditional methods primarily leverages the visual content of item images to learn compatibility in a latent visual space. However, item images just describe the implicit and low-level visual content. Actually, in addition to the item image, a fashion item on most fashion e-commercial websites is usually described by a diverse set of attributes, which have been largely ignored by most existing methods. For example, the item of ID 001 in Figure 1(a) has diverse categorical attributes about *category* (midi-dress), *pattern* (floral), *color* (natural-white), *neckline* (V-Neck), *style* (causal), etc. The attributes not only provide good semantic description of items, but also have the potential to explicitly reveal the intra-connectivity between items. They can help to explain why two fashion items can be grouped together for a fashionable outfit by a set of attribute crosses[7, 32], such as [*Fullbody: pattern=floral*] & [*Fullbody: category=Midi-dresses*] & [*Footwear: category=Sandals*]. Each attribute cross reflects a particular matching pattern.[1]

This paper aims to address the task of *interpretable fashion matching*. We denote a and $\mathcal{A} = \{a_k\}_{k=1}^{|\mathcal{A}|}$ as an item attribute and the whole attribute set. For a given item x_i, we construct its attribute set as $\mathcal{A}_i \subset \mathcal{A}$. Then, we can formally define this new task as:

- **Inputs:** A corpus of fashion items with rich attributes and pairwise matching relationships $\{\mathcal{X}, \mathcal{A}, \mathcal{Y}\}$.
- **Outputs:** (1) A pairwise ranking function for each pair of items (x_i, x_j), i.e., $f : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ which maps a pair of items to a compatibility score value by jointly considering the visual correlations and attribute correlations, and (2) a set of second-order attribute crosses $\{a_p \& a_q\}$ or higher-order attribute crosses: $\{a_p \& a_q \& \cdots a_l\}$ that explicitly reveals which attributes in x_i and x_j dominate the matching process.

3 OUR PROPOSED APPROACH

This paper proposes to address the interpretable fashion matching task, aiming to inject interpretability into the compatibility modeling of fashion items. The key to tackling such interpretable fashion matching task is 1) how to extract the self-interpretable attribute crosses from data, 2) how to learn the representation of the derived

[1]Note that in this work we express the matching pattern as a *attribute cross*, which is a combination of multiple attributes. We use them exchangeable without specification.

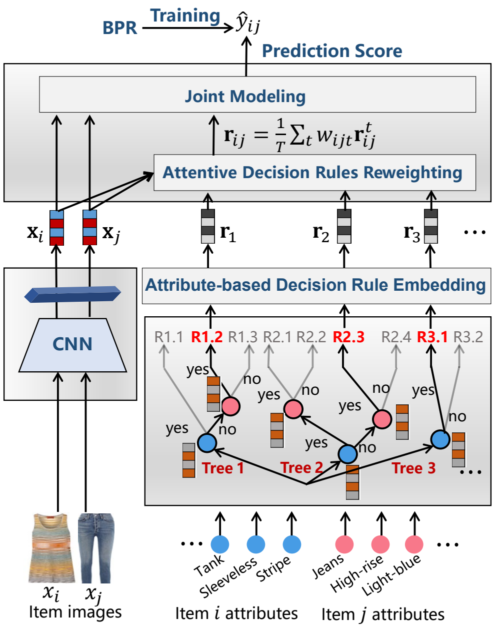

Figure 2: An illustration of our Attribute-based Interpretable Fashion Compatibility (AIC) framework.

attribute crosses, and 3) how to unify the strengths of attribute crosses and item images for joint prediction.

We address the three problems by developing an attribute-based interpretable compatibility (AIC) framework, as shown in Figure 2, which mainly consists of three modules:

- A Tree-based decision rule extraction module that automatically derive a set of self-interpretable decision rules, in which each decision rule can be seen as a high-order attribute cross or a set of second-order attribute crosses.
- An embedding module that learns vector representation for a decision rule by accounting for the attribute semantics.
- A joint modeling module that unifies the visual embedding and attribute-based rule embedding in the same space to predict the compatibility score.

3.1 Tree-based Decision Rule Extraction

The main goal of the interpretable fashion matching framework is to infer attribute-based matching patterns, i.e., attribute crosses. Then, the first problem is how to extract the attribute crosses. A popular solution in industry is to manually craft all the feature crosses, and learn the weight of all feature crosses. Obviously, such straightforward solution is not scalable when we model higher-order attribute interactions on a large scale attribute set. Another solution is to manually define a set of matching rules [22, 26, 28] based on item attributes, such as *White shirt & black trousers*. However, manually defining matching rules usually needs strong domain knowledge and may not be expressive enough to capture complex matching patterns. It is highly desired to infer the rich matching patterns from data automatically.

Motivated by recent works[32, 41] in recommendation domain, we propose to leverage Tree-based models, e.g., CART[3], GBDT[8],

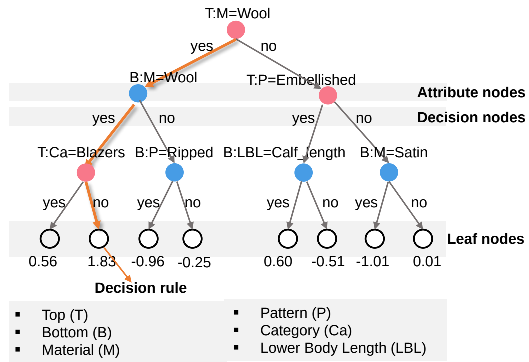

Figure 3: A simple decision tree for a Top-Bottom matching.

XGBoost [6], for automatically constructing self-interpretable attribute crosses from categorical item attributes, due to the self-interpretability and scalability. As shown in Figure 3, a simple decision tree with binary node splits can be represented as $Q = \{\mathcal{V}, \mathcal{D}, \mathcal{E}\}$, where $\mathcal{V}$ denotes two types of *nodes*: one is internal/root nodes that represent features (attributes) and the other is leaf nodes that represent outcomes for prediction, $\mathcal{D}$ denotes binary *decision* nodes (*yes*, or *no*), and $\mathcal{E}$ denotes the edge connecting two nodes. The paths from root to leaf represent decision rules, revealing the reasoning procedure. By training the decision tree using one-hot-encoded categorical attributes as inputs, each derived decision rule can be seen as a high-order *attribute cross* (i.e., matching pattern). As shown in Figure 3, the path from root node to the second leaf node on the left side represents a three-order attribute cross [*Top:Material=Wool*]&[*Bottom:Material=Wool*] & [*Top:Category≠Blazers*]. When the last *decision* is changed from *yes* to *no*, the rule [*Top:Material=Wool*] & [*Bottom:Material=Wool*] & [*Top:Category=Blazers*] still has high prediction score. It uncovers that sometimes the most dominant matching pattern may be a second-order attribute cross.

In this work, we adopt the boosted tree model, e.g., GBDT [8], which is defined as an ensemble of T decision trees $\sum_{t=1}^{T} Q_t$. Then, given the one-hot-encoded categorical attributes $\mathcal{A}_{ij} = (\mathcal{A}_i, \mathcal{A}_j)$ of (x_i, x_j) as inputs, the boosted tree module will return T decision rules $\{r_{ij}^1, \cdots, r_{ij}^t, \cdots, r_{ij}^T\}$, where r_{ij}^t $(1 \leq t \leq T)$ denotes the t-th decision rule returned by its corresponding decision tree. Since a decision rule is directed and has different decision states between two attribute nodes, for clarity, we describe a decision rule in a path-like form

$$r_{ij}^t : a_1^t \xrightarrow{s_1^t} a_2^t \xrightarrow{s_2^t} \cdots a_Z^t \xrightarrow{s_Z^t}, \tag{3}$$

where a_z^t $(1 \leq z \leq Z)$ denotes the z-th attribute in the rule r_{ij}^t, s_z^t denotes the binary decision state of attribute a_z^t, and Z denotes the number of attributes and decisions in the rule r_{ij}^t. The leaf node is not shown in Eq. (3).

Note that we only utilize the GBDT model to automatically extract the decision rules and do not use its prediction scores on the leaf nodes for prediction, since it suffers from poor generalization ability[32]. For unseen attribute vector inputs $\mathcal{A}_{ij}$, it would return a decision rule with all *no* decisions, such as the path from the root node to the first leaf node on the right side in Figure 3.

3.2 Attribute-based Decision Rule Embedding

After extracting a set of decision rules via the boosted tree model, the next question is how to transform the decision rules to vector representations for predicting compatibility score. Since each rule has a unique leaf node which corresponds to a unique ID, prior work [32] proposed to encode rule ID as a vector, while ignoring the semantics of decision rules. To be more specific, such ID embedding method fails to model the semantic correlation between similar rules. To address this problem, we propose to embed the semantics of each rule into a low-dimensional vector by taking the attribute interactions into consideration. We elaborate this solution as follows:

Attribute and Decision Embedding. Recall that each rule is composed by attributes, decisions, and edges connecting two nodes, as show in Figure 3 and 4. To represent the attribute, we first set up a lookup layer to transform the one-hot encodings of all the attributes $\{a_k\}_{k=1}^{|\mathcal{A}|}$ into low-dimensional dense embedding vectors $\{\mathbf{a}_k\}_{k=1}^{|\mathcal{A}|} \in \mathbb{R}^{d\times|\mathcal{A}|}$. While, as shown in Figure 3, each attribute has two *decision* states (*yes* and *no*) in two mutually exclusive decision edges. How to model such decision states into the attribute representation? A simple way is to directly treat the attribute (e.g., [*Top:Material=Wool*]) and its opposite [*Top:Material≠Wool*]) as two independent attributes. Then, we need to optimize $2\times|\mathcal{A}|$ attribute embedding vectors. While, such a solution directly ignores the exclusive relationship between [*Top:Material=Wool*] and [*Top:Material≠Wool*]. We propose to embed the two decision states as the same dimensional vector representations $\mathbf{s}_k \in \mathbb{R}^d$ with the attribute embeddings $\mathbf{a}_k$ via the look up operation. To model the exclusive relationship, we propose to combine the attribute embedding and its corresponding decision embedding by a simple vector translating operation[2]:

$$\overrightarrow{\mathbf{a}}_k = \mathbf{a}_k + \mathbf{s}_k, \tag{4}$$

where $\overrightarrow{\mathbf{a}}_k$ denotes the translated embedding vector of $\mathbf{a}_k$. For simplicity, we use $\overrightarrow{a}_k$ to denote the attribute a_k with decision state s_k, then $\overrightarrow{\mathbf{a}}_k$ denotes its vector representation. In this way, we only need to optimize $(2+|\mathcal{A}|)$ embedding vectors and preserve the exclusive relationship between attribute and its opposite.

Rule Embedding. After injecting the embedding vectors of binary decision states into the attribute embeddings by Eq. (4), we can reformulate Eq. (3) as $r_{ij}^t : \overrightarrow{a}_1^t \rightarrow \overrightarrow{a}_2^t \rightarrow \cdots \overrightarrow{a}_Z^t$, which is a sequence of inner-connected attributes. Then, the popular pooling operation, such as max-pooling or average-pooling, can be used to compute the embedding vector of decision rules based on the attribute embeddings. But this way does not explicitly model the second-order or higher-order attribute interactions, and then cannot identify which attribute cross in the decision rule is the most informative one.

We propose to learn the representation of decision rule based on the interaction of attribute crosses in the rule. As shown in Figure 4, the second-order and higher-order attribute crosses in the rule are respectively described and represented by

- Second-order attribute cross $\overrightarrow{a}_z^t$ & $\overrightarrow{a}_{z+1}^t$, which is represented by $\mathbf{v}_z^{2(t)} = \overrightarrow{\mathbf{a}}_z^t \otimes \overrightarrow{\mathbf{a}}_{z+1}^t$,

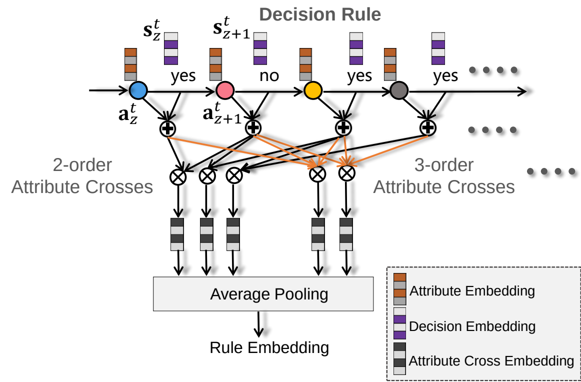

Figure 4: An illustration of the proposed attribute-based decision rule Embedding.

- Higher-order attribute cross $\overrightarrow{a}_z^t$ & $\overrightarrow{a}_{z+1}^t$& $\cdots$ &$\overrightarrow{a}_{z+O-1}^t$, which is represented by $\mathbf{v}_z^{O(t)} = \overrightarrow{\mathbf{a}}_z^t \otimes \overrightarrow{\mathbf{a}}_{z+1}^t \otimes \cdots \otimes \overrightarrow{\mathbf{a}}_{z+O-1}^t$,

where $\mathbf{v}_z^{O(t)} \in \mathbb{R}^d$ $(2 \le O \le Z)$ denotes the embedding vector of the z-th high-order attribute cross. The $\mathbf{v}_z^2$ is a specific form of $\mathbf{v}_z^O$ when $O=2$. The $\otimes$ denotes the element-wise multiplication, i.e., Hadamard Product. Finally, the embedding of the rule r^t is defined as the linear aggregation of all the attribute crosses embedding with an average pooling operation

$$\mathbf{r}_{ij}^t = \frac{1}{N}\sum_{o=2}^{O}\sum_{z=1}^{Z+o-1}\mathbf{v}_z^{o(t)}, \tag{5}$$

where $\mathbf{r}_{ij}^t \in \mathbb{R}^d$, and N is the number of all attribute crosses in the decision rule.

3.3 Visual-Rule Joint Modeling

In this section, we describe how to jointly model visual embedding of item images and attribute-based rule embedding for predicting fashion compatibility. It mainly consists of three submodules: 1) learning low-dimensional visual embeddings of item images with a pretrained CNN, 2) reweighting the embeddings of decision rules with an attention network, 3) jointly leveraging visual embedding and attribute-based rule embedding for compatibility prediction.

Deep Visual Embedding of Items. The deep visual embedding learning module on the left down side of Figure 2 has been widely used in existing visual compatibility learning models due to the strong transferability of *deep* features. This work adopts a pretrained deep CNN (e.g., ResNet-50[12]) to extract visual features from item images. Given an image of item x_i, the output of a pretrained CNN is $\mathbf{x}_i^{cnn} \in \mathbb{R}^{d^{cnn}}$ where $\mathbf{x}_i^{cnn}$ is a high-dimensional visual feature representation of item x_i. Then we apply a one-layer feed forward network to transform the high-dimensional output of CNN into a d-dimensional visual embedding $\mathbf{x}_i \in \mathbb{R}^d$:

$$\mathbf{x}_i = g\left(\mathbf{x}_i^{cnn}\right) = \mathbf{W}^g\mathbf{x}_i^{cnn} + \mathbf{b}^g, \tag{6}$$

where $g(\cdot)$ is a one-layer feed forward network with weight parameters $\mathbf{W}^g \in \mathbb{R}^{d\times d^{cnn}}$ and $\mathbf{b}^g \in \mathbb{R}^d$. The visual embedding module enables our framework generalize to unseen fashion items.

Attentive Decision Rules Re-weighting. Given inputs $(\mathcal{A}_i, \mathcal{A}_j)$ of (x_i, x_j), our boosted tree module (GBDT) returns T decision rules

$[r_{ij}^1, \cdots, r_{ij}^t, \cdots, r_{ij}^T]$. Note that not every rule has the equal contribution to (x_i, x_j), and some rules may also be invalid. Therefore, it is necessary to design an attention module to modulate the contribution of each rule. Inspired by the recent work [4, 28, 32], we apply a multi-layer perceptrons (MLPs) to learn the attentive weight of each derived rule:

$$w'_{ijt} = \mathbf{w}^T \sigma(\mathbf{W}([(\mathbf{x}_i + \mathbf{r}_{ij}^t) \otimes \mathbf{x}_j, \mathbf{r}_{ij}^t]) + \mathbf{b}), \tag{7}$$

$$w_{ijt} = \frac{\exp(w'_{ijt})}{\sum_t^T \exp(w'_{ijt})} \tag{8}$$

where w_{ijt} denotes the weight of the t-th rule corresponding to (x_i, x_j), $\mathbf{W} \in \mathbb{R}^{d\times 2d}$ and $\mathbf{b}$ denotes the weight matrix and bias vector of the hidden layer in our attention module, and $\mathbf{w} \in \mathbb{R}^{d\times 1}$ is the weight vector of the regression layer. The $[\cdot, \cdot]$ denotes the concatenation operation of two vectors. The σ is the non-linear activation function *ReLu*. In Eq. (7), we project $(\mathbf{x}_i + \mathbf{r}_{ij}^t) \otimes \mathbf{x}_j$ into the attention module, which aims to directly capture the interaction $\mathbf{x}_i \otimes \mathbf{x}_j$ and $r_{ij}^t \otimes \mathbf{x}_j$ in the same embedding space. Note that Eq. (7) is implemented as an asymmetrical form by considering the directed matching order (e.g., Top-Bottom[28, 29] and Top-Footwear) in the fashion matching task. Then, we can obtain a unified vector representation of all the derived decision rules corresponding to (x_i, x_j):

$$\mathbf{r}_{ij} = \frac{1}{T} \sum_{t=1}^{T} w_{ijt} \mathbf{r}_{ij}^t \tag{9}$$

The attention module enables the first-time message passing between visual space and attribute-based rule space for learning the importance of decision rules. Note that the attention module endows our framework with interpretability. For each matching pair, we can return the most informative decision rule to explain the matching result.

Joint Prediction. Given the visual embedding vectors $\mathbf{x}_i$ and $\mathbf{x}_j$ of items x_i and x_j, and the unified rule embedding vector $\mathbf{r}_{ij}$, we design a joint modeling solution that can enable the visual part and rule part perform separately and mutually. The complete predictive function is defined by

$$f(x_i, x_j, \mathcal{A}_{ij}) = \underbrace{\mathbf{h}_1^T (\mathbf{x}_i \otimes \mathbf{x}_j)}_{Visual} + \underbrace{\mathbf{h}_2^T \mathbf{r}_{ij}}_{Rule} + \underbrace{\mathbf{h}_3^T ((\mathbf{x}_i + \mathbf{x}_j) \otimes \mathbf{r}_{ij})}_{Visual-Rule}, \tag{10}$$

where $\mathbf{h}_1 \in \mathbb{R}^{d\times 1}$, $\mathbf{h}_2 \in \mathbb{R}^{d\times 1}$, and $\mathbf{h}_3 \in \mathbb{R}^{d\times 1}$ denote the weight parameters of three regression layers, respectively, which yields compatibility predictions from three parts: the first is visual compatibility ($\mathbf{h}_1$), the second is rule-based compatibility ($\mathbf{h}_2$), and the third is visual-rule joint compatibility ($\mathbf{h}_3$). To identify the contribution of each attribute cross in a decision rule, the second term can be rewritten as

$$\mathbf{h}_2^T \mathbf{r}_{ij} = \frac{1}{T} \sum_{t=1}^{T} w_{ijt} \mathbf{h}_2^T \mathbf{r}_{ij}^t = \frac{1}{T \times N} \sum_{t=1}^{T} \sum_{o=2}^{O} \sum_{z=1}^{Z+o-1} w_{ijt} \mathbf{h}_2^T \mathbf{v}_z^{o(t)}, \tag{11}$$

where the $w_{ijt}(\mathbf{h}_2^T \mathbf{v}_z^{o(t)})$ is the prediction score contributed by the attribute cross $\overrightarrow{a}_z^t \,\&\, \overrightarrow{a}_{z+1}^t \& \cdots \,\&\, \overrightarrow{a}_{z+o-1}^t$ in the t-th decision rule.

The third part is equal to $\mathbf{h}_3^T(\mathbf{x}_i \otimes \mathbf{r}_{ij}) + \mathbf{h}_3^T(\mathbf{x}_j \otimes \mathbf{r}_{ij})$, which transforms the interaction of $\mathbf{r}_{ij}$ and $\mathbf{x}_i$ and the interaction of $\mathbf{r}_{ij}$ and $\mathbf{x}_j$ to the compatibility scores, respectively. The third part aims to capture the complex interaction between low-level visual concept and high-level semantic concept (i.e., attributes) in a joint space. It refines the item-item visual compatibility with the intra-connectivity between two items, which enables the second-time message passing between visual space and attribute-based rule space in a mutually enhanced way.

3.4 Learning

We formulate the fashion matching task as a ranking problem, and minimize the Bayesian Personalized Ranking (BPR) objective [25] which forces the prediction score of a matched pair $(x_i, x_j) \in C$ to be larger than that of unmatched pair $(x_i, x_k) \notin C$:

$$\mathcal{L} = \sum_{\mathcal{T}} -\ln \sigma \left(f(x_i, x_j, \mathcal{A}_{ij}) - f(x_i, x_k, \mathcal{A}_{ik})\right), \tag{12}$$

where $\sigma(\cdot)$ is the widely-used logistic sigmoid function. The regularization term has been omitted for clarity. $\mathcal{T}$ denotes a training set of 5-tuples $:\{(x_i, x_j, x_k, \mathcal{A}_{ij}, \mathcal{A}_{ik}) | (x_i, x_j) \in C, (x_i, x_k) \notin C\}$. The matched pair (x_i, x_j) is extracted from the same outfit. The negative item x_k is randomly selected from a different category with x_i, which has not matched with x_i before. Note that our tree-based module is first trained and then fixed as a decision rule extractor.

3.5 Discussion

3.5.1 Interpretability. The main goal of the interpretable fashion matching task is to learn self-interpretable attribute crosses for revealing the reasons behind each matching decision. Our proposed AIC method injects interpretability into the fashion compatibility modeling, which is able to provide two levels of interpretation.

- Given a pair of items x_i and x_j from different categories, the tree module first returns a set of decision rules. Then, our attention model re-weights each rule embedding and selects informative decision rules by the importance w_{ijt} to x_i and x_j as the first-level interpretation. (Rule-based)
- Given a selected decision rule r_{ij}^t, our predictive model in Eq. (11) can identify which attribute cross in the rule dominates this matching. (Attribute cross-based)

In summary, we not only can yield a decision rule to explain the matching process, but also can identify the most dominant attribute cross in the rule. We have conducted a case study in section 4.4 on the interpretability of AIC.

3.5.2 Relation to Tree-enhanced Embedding (TEM). Our proposed AIC has a similar two-way (embedding + tree) architecture with TEM[32]. The key difference lies in the decision rule embedding module. TEM simply encodes ID information as a dense vector to represent a rule, while ignoring the semantics of rules. To be more specific, TEM treats all rules independently and fails to explicitly model the semantic correlation between rules. Moreover, its parameter size is linear with the scale of decision rules, which easily leads to overfitting when the tree number is large (as verified in section 4.3.2). AIC overcomes the limitation of TEM by linearly modeling the attribute interactions into semantics-preserving rule embedding, thus can not only achieve better performance than

TEM, but also provides higher interpretability. Besides, AIC enforces interaction between visual embedding and rule embedding in the prediction layer, which yields better performance.

In summary, 1) AIC learns the attribute-based rule embedding while TEM only learns ID-based rule embedding, 2) AIC not only provides decision rules as an interpretation but also can identify the most informative attribute cross as the second-level interpretation, while TEM only provides rule-level interpretation, and 3) AIC models the interaction of visual embedding and rule embedding in the same embedding space.

4 EXPERIMENTS

To justify the effectiveness of AIC, we conduct extensive experiments to answer the following questions:

- **RQ1**: Can our AIC framework outperform the state-of-the-art approaches?
- **RQ2**: How do different modules of our AIC (e.g., the attribute-based rule embedding module) contribute to the performance?
- **RQ3**: How can our AIC provide easy-to-interpret fashion matching results?

4.1 Dataset Description

The most popular fashion matching dataset is the *Polywore* [11, 29, 35]. However, this dataset does not have fashion attribute annotation. *To the best of our knowledge, there is not available dataset for this fashion matching task, due to the absence of fine-grained attribute annotations.* To effectively evaluate our AIC framework, we collect a large outfit dataset from a personal outfit recommendation website *Lookastic*[2] which provides diverse and fashionable outfit collections with detailed product attribute annotations. We collected 30,790 fashionable outfits from the website, in which both male and female outfits are collected. Each outfit contains a set of items from multiple complementary categories (e.g., Top, Outwear, Bottom, Footwear).

Following the setting in [10, 29], *we extract matched item pairs that are co-occurring in the same outfit as the ground truth for training*, and filter out some improper or incomplete pairs. Finally, we obtain 124,665 matched pairs for men with 5,069 items, 158,755 matched pairs for women with 10,016 items. Apart from the attributes provided by *Lookastic*, we also use the *Visenze*[3] API to extract more item attributes and filter out overlapped attributes. This final dataset has diverse item attribute annotations consisting of 65 item *colors*, 38 *materials*, 40 *patterns*, 253 fine-grained *categories*, 11 *styles*, and 114 category-specific attributes.

We evaluate our proposed AIC with baseline methods on *Lookastic-Men*, and *Lookastic-Women*, respectively. We randomly split the dataset by 70% for training, 20% for testing, and 10% for validation. The validation set is used to tune hyper-parameters and the final comparison is conducted on the test set.

4.2 Experimental Settings

4.2.1 Evaluation Protocols. To evaluate the effectiveness of our model more fairly, we repeat the random dataset split for five times and report the average performance of all methods on the testing set with significance test. For each matched item-item pair in the training sets, we pair it with three randomly sampled negative items from a different category. Each query item and its negative items must not co-occur in the same outfit. For each matched pair in the testing set, we pair it with 500 negative items. Then each method outputs prediction scores for these 501 items. If not mentioned, all the negative items are sampled from the whole dataset but from a different category with the query item.

To evaluate the prediction performance of a ranked list, we use three widely-used information retrieval metrics: the *Mean Reciprocal Rank* (MRR), *Hit Ratio* at rank K (hit@K), and *Normalized Discounted Cumulative Gain* at rank K (ndcg@K). The MRR is the average of the reciprocal ranks of results for a sample of queries. The hit@K intuitively measures whether the test item is present on the top-K list, and the ndcg@K accounts for the position of the hit by assigning higher scores to hits at top-K list. A higher MRR, hit@K, or ndcg@K score denotes a better performance. We calculated all metrics for each test query item and reported the average score. Without special mention, we truncate the ranked list at $K = 5$ and $K = 10$ for hit@K and ndcg@K

4.2.2 Baselines. We compare our proposed AIC with the following baseline methods to justify its effectiveness:

- **Siamese Nets**[31] (**SiaNet**). It measures the visual compatibility using ℓ_2-normalized Euclidean distance. (**Image only**)
- **BPR-DAE**[29]. This work models the pairwise visual compatibility as the inner-product of item embeddings. (**Image only**)
- **TransNFCM**[35]. It is a state-of-the-art fashion matching method that leverages category-level complementary relationships to refine the item-item compatibility. (**Image + coarse category**)
- **VBPR**[13]. It is a strong baseline for visually-aware user-item interaction modeling. It fuses visual information and ID embedding to enhance the item representation. (**Image + ID**)
- **Neural Factorization Machines**[15] (**NFM**). It is a state-of-the-art embedding-based learning method that implicitly models higher-order feature interaction in a nonlinear way. We implement it by encoding all item attributes and item images with embedding vectors. (**Image + attributes**)
- **TEM**[32]. It is a state-of-the-art embedding-based learning method that combines the strength of traditional embedding-based models and the tree-based models. Different with AIC, it learns the ID-based embedding to represent rule. (**Image + attributes**)

Note that we use the same deep visual embeddings of item images for all baselines. The ID embeddings of items in TEM are replaced by visual embeddings of images for a fair comparison. We only use the visual modules of BPR-DAE and TransNFCM in our experiments, due to the absence of textual descriptions in our dataset. We implement all the baseline method, using the same BPR loss, except SiaNet[4].

4.2.3 Parameter Settings. We implement AIC by stochastic gradient descent (SGD) using Pytorch[5]. The pretrained ResNet-50[12] model is applied to extract visual feature of item images using the output of the *pool5* layer. The size of hidden layer for learning low-dimensional visual embedding is set to $d = 64$ as well as the

[2]https://lookastic.com/
[3]https://www.visenze.com/automated-product-tagging
[4]We empirically found that SiaNet performs much better with margin ranking loss
[5]https://pytorch.org

Table 1: Overall Performance Comparison (%) with baseline methods. * and ** denote the statistical significance for $p_{value} < 0.05$ and $p_{value} < 0.01$, respectively, compared to the best baseline.

Dataset	Lookastic-Men				
Methods	MRR	hit@5	hit@10	ndcg@5	ndcg@10
BPR-DAE	23.35	30.97	30.90	23.28	26.17
Siamese	23.05	31.37	40.92	23.04	26.12
TransNFCM	26.14	34.94	44.27	26.28	29.30
VBPR	28.32	36.83	45.40	28.57	31.34
NFM	28.92	37.49	46.37	29.16	32.02
TEM	29.10	37.88	46.97	29.33	32.27
AIC	30.74**	39.51**	48.23**	31.06**	33.88**
Rel. Impro.	5.6%	4.3%	2.6%	5.8%	4.9%
Dataset	Lookastic-Women				
Methods	MRR	hit@5	hit@10	ndcg@5	ndcg@10
BPR-DAE	23.69	32.97	42.25	24.02	27.02
Siamese	24.00	33.71	44.23	24.25	27.65
TransNFCM	29.88	41.01	51.08	30.70	33.96
VBPR	29.46	39.32	48.33	30.06	32.98
NFM	30.49	40.90	50.60	31.15	34.29
TEM	31.63	42.35	52.33	32.32	35.55
AIC	33.19**	43.83*	53.09**	33.94*	37.01**
Rel. Impro.	4.9%	3.4%	1.4%	5.0%	4%

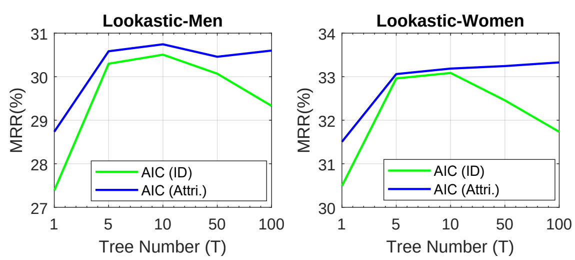

Figure 5: Comparison (MRR (%)) of the attribute-based (AIC (Attri.)) and ID-based (AIC (ID)) rule embeddings.

latent embedding size of item attributes. The mini-batch size is set to 1024 and the learning rate η is searched in {0.001, 0.01, 0.05, 0.1 } on validation set. We use XGBoost[6] to generate the tree-structure where the number of trees and the maximum depth of trees are searched in {1, 10, 30, 50, 80, 100 } and {4, 6, 8, 10} on validation set, respectively. If not mentioned, the tree number and maximum depth are fixed as 10 and 6 on testing set, respectively. We employ SGD to optimize all methods with momentum factor as 0.9. We run all methods until convergence and drop the learning rate η to $\eta/10$ every 10 epochs.

4.3 Performance Comparison

We first compare the performance of all the methods. We then justify how our method can effectively learn the semantics of decision rule for enhancing the compatibility modeling.

4.3.1 Overall Comparison. (**R1**) Table 1 displays the performance comparison *w.r.t.* MRR, hit@K (K=5, 10), and ndcg@K (K=5, 10) among the baseline methods on the *Lookastic-Men* and *Lookastic-Women* datasets. We have the following findings:

- BPR-DAE and SiaNet, which merely rely on visual information, achieve poor performance. TransNFCM and VBPR perform much better, since TransNFCM exploits the category-level complementary relationship as the connection between compatible items and VBPR combines the ID embedding of items and visual embedding for feature augmentation. It indicates the necessity of exploiting the side information for modeling the complex fashion compatibility beyond the visual information, since visual embeddings of items just comprise of low-level signals, which cannot effectively capture the complex interaction patterns.
- NFM and TEM achieve competitive performance, which can be attributed to the utilization of feature interaction. NFM exploits high-order feature interaction with a multi-layer MLPs in a non-linear way, which consistently outperforms the strong baseline VBPR. While, TEM uses a tree-based model to automatically derive higher-order feature crosses with an attention mechanism. It slightly outperforms NFM on both datasets, especially the *Lookastic-Women* dataset where more diverse item-item interactions are provided. It indicates the effectiveness of modeling the high-order feature interactions.
- Our proposed AIC substantially outperforms the state-of-the-art methods, NFM and TEM, on both datasets. This demonstrates the effectiveness of AIC. It not only integrates the predictions from both visual space and attribute-based rule space in the prediction layer, but also explicitly learns the semantics of decision rules based on the attribute interaction in the rule. Such semantics-preserving rule embedding is jointly modeled with visual information in a unified space, which leads to better performance and also reveals the complex matching patterns in a more explicit way.

4.3.2 Effect of Attribute-based Decision Rules Embedding. (**R2**) One of the contributions of AIC is that it learns the semantics of decision rules by explicitly modeling the attribute interaction. While, the prior work [32] proposes to learn the ID embedding of each rule without considering the content of each rule. To justify the effect of our attribute-based rule embedding, we compare the performance of this two rule embeddings in Table 2 and Figure 5, which are termed as AIC(Attri.) and AIC(ID), respectively. Note that we fix the maximum depth of tree as 6 and vary the number of decision trees $T \in [1, 5, 10, 50, 100]$ to generate different tree structures for comparison. We have the following observations from Table 2 and Figure 5.

Overall, the attribute-based rule embedding consistently outperforms the ID-based rule embedding. When the tree number is 5 or 10, AIC (ID) performs comparable to AIC (Attri.). However, when the tree number is increased to 50 or 100, the performance of AIC (ID) drops significantly. It reflects that the AIC (ID) is sensitive to the tree numbers. It easily suffers from overfitting when the tree number is large, since its parameter size is linear with the scale of all the leaf nodes in GBDT. While AIC(Attri.) directly optimizes the attribute embedding, thus could effectively capture the semantic correlation between similar rules. The performance comparison justifies the effectiveness of AIC on the semantic encoding of rules.

4.3.3 Effect of Visual-Rule Joint Modeling. (**R2**) As shown in Eq. (10), AIC not only predicts the visual compatibility and semantic compatibility with two regression vectors ($\mathbf{h}_1$ and $\mathbf{h}_2$) respectively,

[6] https://xgboost.readthedocs.io/en/latest/

Table 2: Comparison (hit@5, ndcg@5, %) of the attribute-based (AIC (Attri.)) and ID-based (AIC (ID)) rule embeddings.

TreeNum	Datasets	Lookastic-Men		Lookastic-Women	
	Methods	hit@5	ndcg@5	hit@5	ndcg@5
T=1	AIC (Attri.)	37.16	28.92	42.05	32.22
	AIC (ID)	35.99	27.60	41.05	31.27
T=5	AIC (Attri.)	39.34	30.88	43.66	33.80
	AIC (ID)	39.05	30.59	43.57	33.69
T=10	AIC (Attri.)	39.51	31.06	43.83	33.94
	AIC (ID)	39.25	30.83	43.46	33.78
T=50	AIC (Attri.)	39.32	30.77	43.81	33.97
	AIC (ID)	38.85	30.33	42.85	33.16
T=100	AIC (Attri.)	39.45	30.90	43.87	34.06
	AIC (ID)	37.88	29.55	41.99	32.38

Table 3: Ablation study on the effect of visual-rule interaction (VRI) term.

Dataset	Lookastic-Men				
Methods	MRR	hit@5	hit@10	ndcg@5	ndcg@10
AIC (Rule only)	18.90	25.17	34.11	18.37	21.25
AIC (VRI only)	29.22	38.03	46.98	29.49	32.38
AIC (without VRI)	30.38	39.13	47.92	30.68	33.52
AIC (with VRI)	30.74	39.51	48.23	31.06	33.88
Dataset	**Lookastic-Women**				
Methods	**MRR**	**hit@5**	**hit@10**	**ndcg@5**	**ndcg@10**
AIC (Rule only)	23.40	30.97	39.82	23.30	26.16
AIC (VRI only)	33.12	43.64	53.28	33.83	36.95
AIC (without VRI)	32.73	43.19	52.62	33.43	36.49
AIC (with VRI)	33.18	43.83	53.09	33.94	37.00

but also transforms the visual-rule interaction (VRI) $(\mathbf{x}_i + \mathbf{x}_j) \otimes \mathbf{R}_{ij}$ to a compatibility score with the regression vector $\mathbf{h}_3$. This section investigates how AIC perform with/without the VRI term, and how AIC perform with the rule term or VRI term only.

The performance comparison is shown in Table 3. If only using the rule term, AIC achieves poor performance, since tree-based module has poor generalization ability [32]. If only using the VRI term, AIC achieves comparable performance to the combination of other two predictions $(\mathbf{h}_1(\cdot) + \mathbf{h}_2(\cdot))$ on the *Lookastic-Men* dataset, and even better performance on the *Lookastic-Women* dataset which has richer item-item interactions. It yields 38.03% hit@5 score and 43.64% hit@5 score, respectively, on the two datasets, which outperforms most of the baseline methods in Table 1. When the VSI term is integrated with the other two terms, it effectively improves the prediction from 39.13% to 39.51% on *Lookastic-Men* and from 43.19% to 43.83% on *Lookastic-Women* in terms of hit@5 score. On *Lookastic-Women*, the VRI term has dominated the prediction. It shows the necessity and effectiveness of modeling the interaction of visual embedding and rule embedding in a shared embedding space.

4.3.4 Effect of the Attention Network. (**R2**) As mentioned in section 3.3, we design an attention network to re-weight the decision rule embeddings. This section investigates how this attention network improve the performance. We replace the attention module with average pooling/max-pooling and then compare the performance of AIC with the two variants. As shown in Figure 6, the attention

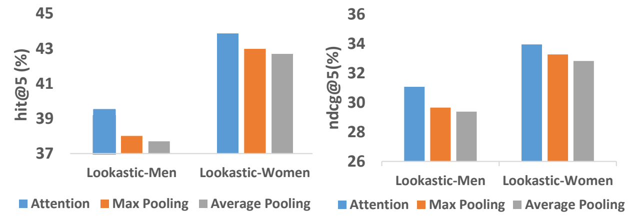

Figure 6: Ablation study on the effect of the attention network using hit@5 (Left) and ndcg@5 (Right).

network consistently outperforms the average pooling and max-pooling operations in terms of hit@5 and ndcg@5. It indicates that some derived rules are invalid. It will degrade the performance by simply aggregating all the rule embedding with average pooling. Although the max-pooling operation obtains better performance than average pooling, it is an element-wise nonlinear operation, which makes the matching process hard-to-interpret. Overall, the attention network not only makes the item-item matching easy-to-interpret but also further improves the performance.

4.4 Case Study on Interpretation (R3)

To demonstrate the interpretability of AIC, we visualize two item-item matching cases on *Lookastic-Women* in Figure 7. Figure 7 (a) is a Top-Bottom case, and Figure 7 (b) is a Fullbody-Footwear case. Each item-item matching pair is sampled on the testing set (positive). For simplicity, the maximum depth of GBDT is set to 4 and only second-order attribute crosses are computed. As shown in Figure 7, the abbreviations of attributes are shown on the right side of each decision rule and the normalized score of each second-order attribute cross is shown on the left side.

For the first case in Figure 7 (a), the input is a *navy coat* paired with *low rise gray jeans*. We observe that the first decision rule encodes some common sense matching patterns, such as *Sophisticated knee length top* doesn't match with *shorts*, and *Sophisticated knee length top* matches with *low rise bottom*. In most case, *high rise bottom* is more likely to match with *short body length top*, thus could make women's beautiful waist curve be seen clear. By our proposed AIC, we also identify the most dominant attribute cross in a decision rule. The second-order attribute cross with the highest score in the first rule is [*Bottom: Rise Type=Low rise*]&[*Top: Style=Sophisticated*]. For the second decision rule, it still cares about the clothing length. The most dominant attribute cross is [*Top: Sleeve Length=Long*] &[*Bottom: Lower Body Length=*7/8], which can be explained as *long sleeve top* goes with *long body length bottom*. For the Fullbody-Footwear case, the input is a *white sleeveless cutout dress* paired with *white heels*. The first decision rule is mainly dominated by the attribute cross [*Fullbody: Color=White*]&[*Footwear: Color=White*], which is a common matching pattern. The second decision rule is dominated by the attribute cross [*Fullbody: Pattern=Cutout*]&[*Footwear: Heel Type=Common heels*].

Overall, the derived matching patterns are consistent with the given matched pairs, and the discovered second-order attribute crosses have higher readability and are also easy-to-interpret. The two matching cases demonstrates AIC's capability of providing more informative and easy-to-interpret matching patterns.

Figure 7: Visualization of the derived decision rules and the normalized prediction score of each second-order attribute cross in the rule. The highest score is marked in red. Note that the binary *decision* state has been merged with its corresponding attribute for simplicity.

5 RELATED WORK

Fashion Matching. Existing works can be mainly classified into two groups: one is outfit creation [11, 20] aiming to automatically compose fashion outfits , and the other one is item-item compatibility [5, 14, 24, 28–30, 35], which is close to our work. Most existing methods of the second group cast fashion matching as a metric learning [36, 37] problem by assuming that a pair of matched items should be *close* to each other in a latent space. Earlier works model the pairwise compatibility with data-independent interaction functions, e.g., inner-product[29], or Euclidean distance[5, 24], which are improved by data-dependent interaction function, such as probabilistic mixtures of non-metric embeddings [14], and category-aware conditional similarity [30, 35]. Our work is related to the second direction but addresses the new and challenging task of interpretable fashion matching, where we not only predict compatibility for unseen pairs but also aim to learn self-interpretable matching patterns to uncover the reasons behind each matching decision. Our work is different with the recent work [28] which first manually constructs a set of matching rules and then use these rules to guide the item embedding learning. The main limitation of [28] is that manually constructing matching rules usually rely on strong domain knowledge, thus resulting in poor scalability.

Fashion Attributes. In recent years, substantial works [1, 10, 19, 21, 23, 27] have been devoted to extract and analyse visual descriptive attributes from fashion images or related textual descriptions for cross modal retrieval [21], interactive fashion search [9, 40], classification [23, 27], and fashion trend prediction [1]. Unlike prior work, this paper prefers to use the rich product attributes associated with fashion items to design an interpretable fashion matching framework. Current visual analysis methods can facilitate our work when the attribute annotation is unavailable.

User-item Recommendation. Our work is also related to personalized recommendation [16, 17, 33], and multimedia recommendation methods [4, 34, 38], which leverage the ID information and visual information of items to model user-item interaction. In this work, we only focus on cross-category item matching without considering the user information. While, the *user* attributes can be easily incorporated into AIC for personalized compatibility modeling, which is left for our future work.

6 CONCLUSION

In this paper, we developed an attribute-based interpretable compatibility (AIC) method, which aims to inject interpretability into the pairwise compatibility modeling. Specifically, we devised a two-way compatibility architecture. Given a matched pair of items, we automatically extract a set of decision rules from a boosted tree model and learn the semantics-preserved rule embedding by explicitly modeling the attribute interaction. Then, we introduced a joint modeling module to unify the strengths of visual information and attribute-based rule information in a shared embedding space, which facilitates the information propagation between visual space and rule space in a mutually-enhanced way. By such a two-way architecture, AIC could not only predict the compatibility score of unseen pairs, but also derive self-interpretable matching patterns to reveal the reasons behind each matching decision. In summary, this work contributes a self-interpretable way for fashion compatibility modeling by deriving the intra-connectivity between items from rich fashion attributes.

As future work, we will consider discovering informative extra-connectivity between items from a domain-specific knowledge graph which could encode richer information, e,g, designer, celebrity, fashion show, country, religion, etc, to further enrich the interpretability of AIC. We are also interested in incorporating the *user* profile, such as age, occupy, gender, city, social relationships, etc., into AIC for personalized fashion matching and personalized outfit composition. We will also try to extend AIC to facilitate other attribute-based visual matching/retrieval tasks [18, 22].

7 ACKNOWLEDGMENTS

This research is part of NExT++ research and also supported in part by the Thousand Youth Talents Program 2018, and in part by the National Natural Science Foundation of China (NSFC) under Grant 61725203 and Grant 61732008. NExT++ research is supported by the National Research Foundation, Prime Minister's Office, Singapore under its IRC@SG Funding Initiative.

REFERENCES

[1] Ziad Al-Halah, Rainer Stiefelhagen, and Kristen Grauman. 2017. Fashion Forward: Forecasting Visual Style in Fashion. In *ICCV*. 388–397.
[2] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *NIPS*. 2787–2795.
[3] Leo Breiman. 2017. *Classification and regression trees.* Routledge.
[4] Jingyuan Chen, Hanwang Zhang, Xiangnan He, Liqiang Nie, Wei Liu, and Tat-Seng Chua. 2017. Attentive collaborative filtering: Multimedia recommendation with item-and component-level attention. In *SIGIR*. ACM, 335–344.
[5] Long Chen and Yuhang He. 2018. Dress Fashionably: Learn Fashion Collocation With Deep Mixed-Category Metric Learning. In *AAAI*. 2103–2110.

[6] Tianqi Chen and Carlos Guestrin. 2016. Xgboost: A scalable tree boosting system. In *SIGKDD*. ACM, 785–794.

[7] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, et al. 2016. Wide & deep learning for recommender systems. In *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems*. ACM, 7–10.

[8] Jerome H Friedman. 2001. Greedy function approximation: a gradient boosting machine. *Annals of statistics* (2001), 1189–1232.

[9] Xiaoxiao Guo, Hui Wu, Yu Cheng, Steven Rennie, Gerald Tesauro, and Rogerio Feris. 2018. Dialog-based interactive image retrieval. In *Advances in Neural Information Processing Systems*. 678–688.

[10] Xintong Han, Zuxuan Wu, Phoenix X Huang, Xiao Zhang, Menglong Zhu, Yuan Li, Yang Zhao, and Larry S Davis. 2017. Automatic spatially-aware fashion concept discovery. In *ICCV*. IEEE, 1472–1480.

[11] Xintong Han, Zuxuan Wu, Yu-Gang Jiang, and Larry S Davis. 2017. Learning fashion compatibility with bidirectional lstms. In *ACM MM*. ACM, 1078–1086.

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

[13] Ruining He and Julian McAuley. 2016. VBPR: Visual Bayesian Personalized Ranking from Implicit Feedback.. In *AAAI*. AAAI Press, 144–150.

[14] Ruining He, Charles Packer, and Julian McAuley. 2016. Learning compatibility across categories for heterogeneous item recommendation. In *ICDM*. IEEE, 937–942.

[15] Xiangnan He and Tat-Seng Chua. 2017. Neural factorization machines for sparse predictive analytics. In *Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval*. ACM, 355–364.

[16] Xiangnan He, Zhenkui He, Jingkuan Song, Zhenguang Liu, Yu-Gang Jiang, and Tat-Seng Chua. 2018. NAIS: Neural Attentive Item Similarity Model for Recommendation. *IEEE TKDE* 30, 12 (2018), 2354–2366.

[17] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *WWW*. 173–182.

[18] R. Hong, L. Li, J. Cai, D. Tao, M. Wang, and Q. Tian. 2017. Coherent Semantic-Visual Indexing for Large-Scale Image Retrieval in the Cloud. *IEEE Transactions on Image Processing* 26, 9 (2017), 4128–4138.

[19] Wei-Lin Hsiao and Kristen Grauman. 2017. Learning the latent "look": Unsupervised discovery of a style-coherent embedding from fashion images. In *ICCV*.

[20] Wei-Lin Hsiao and Kristen Grauman. 2018. Creating capsule wardrobes from fashion images. In *CVPR*. 7161–7170.

[21] Lizi Liao, Xiangnan He, Bo Zhao, Chong-Wah Ngo, and Tat-Seng Chua. 2018. Interpretable Multimodal Retrieval for Fashion Products. In *2018 ACM Multimedia Conference on Multimedia Conference*. ACM, 1571–1579.

[22] Lizi Liao, Yunshan Ma, Xiangnan He, Richang Hong, and Tat-Seng Chua. 2018. Knowledge-aware Multimodal Dialogue Systems. In *ACM MM*. ACM, 801–809.

[23] Ziwei Liu, Ping Luo, Shi Qiu, Xiaogang Wang, and Xiaoou Tang. 2016. Deepfashion: Powering robust clothes recognition and retrieval with rich annotations. In *CVPR*. 1096–1104.

[24] Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton Van Den Hengel. 2015. Image-based recommendations on styles and substitutes. In *SIGIR*. ACM, 43–52.

[25] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian personalized ranking from implicit feedback. In *UAI*. AUAI Press, 452–461.

[26] Amrita Saha, Mitesh M Khapra, and Karthik Sankaranarayanan. 2018. Towards Building Large Scale Multimodal Domain-Aware Conversation Systems. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

[27] Edgar Simo-Serra and Hiroshi Ishikawa. 2016. Fashion style in 128 floats: joint ranking and classification using weak data for feature extraction. In *CVPR*. 298–307.

[28] Xuemeng Song, Fuli Feng, Xianjing Han, Xin Yang, Wei Liu, and Liqiang Nie. 2018. Neural Compatibility Modeling with Attentive Knowledge Distillation. In *SIGIR*. New York, USA, 5–14. https://doi.org/10.1145/3209978.3209996

[29] Xuemeng Song, Fuli Feng, Jinhuan Liu, Zekun Li, Liqiang Nie, and Jun Ma. 2017. Neurostylist: Neural compatibility modeling for clothing matching. In *ACM MM*. ACM, 753–761.

[30] Mariya I Vasileva, Bryan A Plummer, Krishna Dusad, Shreya Rajpal, Ranjitha Kumar, and David Forsyth. 2018. Learning Type-Aware Embeddings for Fashion Compatibility. In *ECCV*. 390–405.

[31] Andreas Veit, Balazs Kovacs, Sean Bell, Julian McAuley, Kavita Bala, and Serge Belongie. 2015. Learning visual clothing style with heterogeneous dyadic co-occurrences. In *ICCV*. IEEE, 4642–4650.

[32] Xiang Wang, Xiangnan He, Fuli Feng, Liqiang Nie, and Tat-Seng Chua. 2018. Tem: Tree-enhanced embedding model for explainable recommendation. In *Proceedings of the 2018 World Wide Web Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 1543–1552.

[33] Xiang Wang, Xiangnan He, Liqiang Nie, and Tat-Seng Chua. 2017. Item silk road: Recommending items from information domains to social users. In *Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval*. ACM, 185–194.

[34] Qidi Xu, Fumin Shen, Li Liu, and Heng Tao Shen. 2018. GraphCAR: Content-aware Multimedia Recommendation with Graph Autoencoder. In *ACM SIGIR*. 981–984.

[35] Xun Yang, Yunshan Ma, Lizi Liao, Meng Wang, and Tat-Seng Chua. 2019. TransNFCM: Translation-Based Neural Fashion Compatibility Modeling. In *AAAI*.

[36] Xun Yang, Meng Wang, and Dacheng Tao. 2018. Person Re-Identification With Metric Learning Using Privileged Information. *TIP* 27, 2 (2018), 791–805.

[37] Xun Yang, Peicheng Zhou, and Meng Wang. 2018. Person Reidentification via Structural Deep Metric Learning. *IEEE TNNLS* 99 (2018), 1–12.

[38] Wenhui Yu, Huidi Zhang, Xiangnan He, Xu Chen, Li Xiong, and Zheng Qin. 2018. Aesthetic-based clothing recommendation. In *WWW*. 649–658.

[39] Xishan Zhang, Jia Jia, Ke Gao, Yongdong Zhang, Dongming Zhang, Jintao Li, and Qi Tian. 2017. Trip outfits advisor: Location-oriented clothing recommendation. *TMM* 19, 11 (2017), 2533–2544.

[40] Bo Zhao, Jiashi Feng, Xiao Wu, and Shuicheng Yan. 2017. Memory-augmented attribute manipulation networks for interactive fashion search. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1520–1528.

[41] Qian Zhao, Yue Shi, and Liangjie Hong. 2017. Gb-cent: Gradient boosted categorical embedding and numerical trees. In *Proceedings of the 26th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 1311–1319.