Complete the Look: Scene-based Complementary Product Recommendation

Wang-Cheng Kang[3]*, Eric Kim[1], Jure Leskovec[1,2], Charles Rosenberg[1], Julian McAuley[3]
[1]Pinterest, [2]Stanford University, [3]UC San Diego
{wckang,jmcauley}@ucsd.edu, {ekim,jure,crosenberg}@pinterest.com

Abstract

Modeling fashion compatibility is challenging due to its complexity and subjectivity. Existing work focuses on predicting compatibility between product images (e.g. an image containing a t-shirt and an image containing a pair of jeans). However, these approaches ignore real-world 'scene' images (e.g. selfies); such images are hard to deal with due to their complexity, clutter, variations in lighting and pose (etc.) but on the other hand could potentially provide key context (e.g. the user's body type, or the season) for making more accurate recommendations. In this work, we propose a new task called 'Complete the Look', which seeks to recommend visually compatible products based on scene images. We design an approach to extract training data for this task, and propose a novel way to learn the scene-product compatibility from fashion or interior design images. Our approach measures compatibility both globally and locally via CNNs and attention mechanisms. Extensive experiments show that our method achieves significant performance gains over alternative systems. Human evaluation and qualitative analysis are also conducted to further understand model behavior. We hope this work could lead to useful applications which link large corpora of real-world scenes with shoppable products.

1. Introduction

Visual signals are a key feature for fashion analysis. Recent advances in deep learning have been adopted by both academia [18, 27, 45] and industry [16, 53, 56] to realize various fashion-related applications, ranging from clothing recognition to fashion retrieval. Fashion images can be categorized into scene images (fashion images in the wild) and product images (fashion item images from shopping websites, usually containing a single product on a plain background). Generally speaking, the former (e.g. selfies, street photos) predominate on image sharing applications, whereas the latter are more common on online shopping websites.

Figure 1: A comparison of the use cases of product-based and scene-based complementary product recommendation. Our approach (bottom) seeks to recommend compatible fashion items based on a real-world scene, while product-based approaches (top) consider compatibility between products.

We seek to bridge the gap between these two types of images, via a new task called *Complete the Look* (CTL), in which we seek to recommend fashion products from various categories that complement (or 'go well with') the given scene (Figure 1). Compared to existing approaches, this setting corresponds more closely to real-world use-cases in which users might seek recommendations of complementary items, based on images they upload 'in the wild.'

Fashion compatibility has been studied previously [42, 11], though existing approaches mainly consider only product images (Figure 1). In comparison, our scene-based CTL task has three significant features: 1) scene images contain not only the fashion items worn by the subject (or user), but also rich context like their body type, the season, etc. By exploiting this side-information, we can potentially provide more accurate and customized recommendations; 2) our system can be adopted by users to give fashion advice (e.g. shoes that go well with your outfit) simply by uploading (e.g.) a selfie; 3) our system can be readily adapted to existing platforms to recommend products appearing in fashion images.

Learning scene-product compatibility is at the core of

*Work done while intern at Pinterest.

the CTL task. However, constructing appropriate ground-truth data to learn the notion of compatibility is a significant challenge. Existing large-scale fashion datasets are typically labeled with clothing segments, attributes, or landmarks [27, 2, 57], which are absent of any information regarding compatibility. Product-based methods have adopted (for example) Amazon's co-occurrence data [42, 29] or Polyvore's outfit data [11, 41, 38] to learn product-to-product compatibility. However, these datasets can not be used for our CTL task, as they lack images from real-world scenes. In addition to the problem of data availability, another challenge is to estimate the compatibility between product images and real-world fashion images, whose characteristics can differ significantly.

As mentioned above, existing studies typically consider compatibility of product images [42, 11], meaning that new data and techniques must be introduced for our CTL task. Another line of related work considers a cross-scenario fashion retrieval task called Street2Shop (also known as *Shop the Look*, or STL) [18, 26] which seeks to retrieve similar-looking (or even identical) products given a scene image and a bounding box of the query product. Human-labeled datasets have been introduced to estimate cross-scenario similarity [18], though our CTL task differs from STL in that we seek to learn a notion of complementarity (instead of similarity), and critically the desired complementary products typically *don't* appear in the given scene (Figure 2).

In this paper, we design an approach to generate CTL datasets based on STL data via cropping. In addition to leveraging existing datasets from the fashion domain, we also consider the domain of interior design (Section 3). We learn global embeddings from scene and product images and local embeddings from local regions of the scenes, and measure scene-product compatibility in a unified style space with category-guided attention (Section 4). We evaluate both the overall and Top-K ranking performance of our method against various baselines, quantitatively and qualitatively analyze the attended regions, and perform a user study to measure the difficulty of the task (Section 5).

2. Related Work

Visual Fashion Understanding. Recently, computer vision for fashion has attracted significant attention, with various applications typically built on top of deep convolutional networks. Clothing 'parsing' is one such application, which seeks to parse and categorize garments in a fashion image [51, 24, 52]. Since clothing has fine-graind style attributes (e.g. sleeve length, material, etc.), some works seek to identify clothing attributes [4, 19, 3], and detect fashion landmarks (e.g. sleeve, collar, etc.) [45, 27]. Another line of work considers retrieving fashion images based on various forms of queries, including images [27, 36], attributes [8, 1], occasions [25], videos [6], and user preferences [17]. Our work is closer to the 'cross-scenario' fashion retrieval setting (called street2shop) which seeks to retrieve fashion products appearing in street photos [26, 18], as the same type of data can be adapted to our setting.

Complementary Item Recommendation. Some recent works seek to identify whether two products are complementary, such that we can recommend complementary products based on the user's previous purchasing or browsing patterns [28, 55, 46]. In the fashion domain, visual features can be useful to determine compatibility between items, for example in terms of pairwise compatibility [42, 38, 41, 29], or outfit compatibility [23, 11, 13, 40]. The former setting takes a fashion item as a query and seeks to recommend compatible items from different categories (e.g. recommend jeans given a t-shirt). The latter seeks to select fashion items to form compatible outfits, or to complete a partial outfit. Our method retrieves compatible products based on a real-world scene containing rich context (e.g. garments, body shapes, occasions), which can also be viewed as a form of complementary recommendation. However this differs from existing methods which seek to model product-product compatibility from pairs of images containing products. In addition to retrieving existing products, one recent approach uses generative models to generate compatible fashion items [35].

Attention Mechanisms. 'Attention' has been widely used in computer vision tasks including image captioning [49, 5], visual question answering [34, 48], image recognition [15, 43], and generation [50, 54]. Attention is mainly used to 'focus' on relevant regions of an image (known as 'spatial attention'). To identify relevant regions of fashion images, previous methods have adopted pretrained person detectors to segment images [26, 39]. Another approach discovers relevant regions by attribute activation maps (AAMs) [58], generated using labels including clothing attributes [1] and descriptions [10]. Recently, attention mechanisms have achieved strong performance on visual fashion understanding tasks like clothing categorization and fashion landmark detection [45]. Our work is the first (to our knowledge) to apply attention to discover relevant regions guided by supervision in the form of compatibility.

Deep Similarity Learning. A variety of methods have been proposed to measure similarity with deep neural networks. Siamese Networks are a classic approach, which seek to learn an embedding space such that similar images have short distances, and have been applied to face verification and dimensionality reduction [7, 9]. Recent methods tend to use triplet losses [33, 44] by considering an anchor image, a positive image (similar to the anchor), and a negative image (randomly sampled), such that the distance from the anchor to the positive image should be less than that of the negative. Recent studies have found that better sampling strategies (e.g. sampling 'hard' negatives) can aid performance [33, 47]. In our method, we seek to learn a unified

style space where compatible scenes and products are close, as they ought to represent similar styles.

3. Datasets

We first introduce datasets for the *Shop the Look* (STL) task, before describing how to convert STL data into a format that can be used for our *Complete the Look* (CTL) task.

3.1. Shop the Look Datasets

As shown in Figure 2, the *Shop the Look* (aka Street2Shop) task consists of retrieving visually similar (or even identical) products based on a scene image and a bounding box containing the query product. This application is useful, for example, when a user sees a celebrity wearing an item (e.g. a purse), allowing them to easily search and buy the product (or a similar one) by taking a photo and selecting a bounding box of the item.

The main challenge here arises due to the difference between products (e.g. clothing) in real-world scenes versus that of online shopping images, where the latter are typically in a canonical pose, on a plain background, adequately lit, etc. To tackle the problem, a recent study sought to collect human-labeled datasets which include bounding boxes of products in scene images, the associated product images, as well as the category of each product [18] (Figure 2). We describe three datasets that can be used for the *Shop the Look* task as follows:

Exact Street2Shop[1] Kiapour *et al.* introduced a first human-labeled dataset for the street2shop task [18]. They crawled data from ModCloth,[2] an online fashion store where people can upload photos of themselves wearing products, indicating the exact items they are wearing. ModCloth also provides category information for all products. However, since bounding boxes are not provided, the authors used Amazon's Mechanical Turk to label the bounding box of products in scene images.

Pinterest's Shop The Look We obtained two STL datasets from Pinterest[3], containing various scene images and shoppable products from partners. STL-Fashion contains fashion images and products, while STL-Home includes interior design and home decor items. Both datasets have scene-product pairs, bounding boxes for products, and product category information, all of which are labeled by internal workers. Unlike the Exact Street2Shop dataset [18] where users only provide product matches, here workers also label products that have a *similar* style to the observed product and are compatible with the scene. Furthermore, the two datasets are much larger in terms of both the number of images and scene-product pairs (Table 1).

[1] http://www.tamaraberg.com/street2shop/
[2] https://www.modcloth.com/
[3] https://www.pinterest.com/

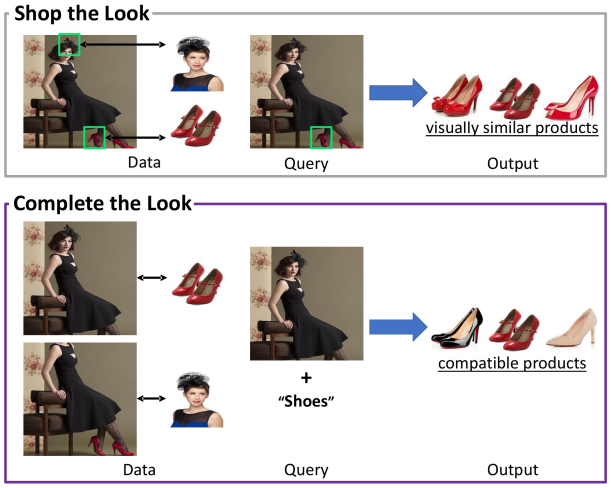


Figure 2: A comparison of data formats and tasks between STL and CTL. STL focuses on retrieving similar products while CTL seeks to recommend compatible items that don't appear in the scene.

3.2. Can STL Data be Used for CTL?

Estimating scene-product compatibility is at the core of the CTL task. Although existing STL datasets provide abundant scene-product pairs, directly using them to learn a notion of compatibility (i.e., viewing each pair as a compatible scene and product) is flawed.

For example, suppose we wanted to learn a CTL model based on STL data. The model might be trained to predict a high compatibility score for each scene/product pair (s / p^+) in the STL data, and predict a low compatibility score for the negative product p^- (e.g. via random sampling). That is, the product p^+ appears in the scene s while the product p^- doesn't. Here it is possible that the model will merely learn to *detect* whether the product appears in the scene (i.e., give a high compatibility score if it appears, and a low score otherwise), instead of measuring compatibility. In this case, the model would fail on the CTL task which seeks to recommend compatible products which *don't* appear in the scene. Empirical results also show that such an approach leads to inferior performance.

The above issue arises mainly because the model 'sees' the product in the scene image. To address it, we propose a strategy to adapt STL datasets for the CTL task. The core idea is to remove the product by cropping the scene image, which forces the model to learn the compatibility between the remaining part of the scene image and the product.

3.3. Generating CTL Datasets

To generate CTL datasets based on STL data, and overcome the issue mentioned above, we propose to crop scene images to exclude their associated products. Given a scene

Name	Source	#Scene	#Product	#Pair	Product Categories (in descending order of quantity)
Fashion-1	Exact Street2Shop [18]	10,482	5,238	10,608	footwear, tops, outerwear, skirts, leggings, bags, pants, hats, belts, eyewear
Fashion-2	STL-Fashion (Pinterest)	47,739	38,111	72,198	shoes, tops, pants, handbags, coats, sunglasses, shorts, skirts, earrings, necklaces,
Home	STL-Home (Pinterest)	24,022	41,306	93,274	rugs, chairs, light fixtures, pillows, mirrors, faucets, lamps, sofas, tables, decor, curtains

Table 1: Data statistics (after preprocessing). Each pair contains a compatible scene and product.

image I_s and a bounding box B for a product, we consider four candidate regions (i.e., top, bottom, left, and right) that don't overlap with B, and select whichever has the greatest area as the cropped scene image. Specifically, we perform the following procedure to crop scene images:

(i) In some cases, the bounding box doesn't fully cover the product. As we don't want the model to see even a small piece of the product (which might reveal e.g. its color), we slightly expand the bounding box B to ensure that the product is likely to be fully covered. Specifically, we expand all bounding boxes by 5% of the image length.

(ii) Calculate areas of the candidate regions, and select the one whose area is largest. For fashion images, we observe that almost all subjects are in a vertical pose, so we only consider the 'top' and 'bottom' regions (as the left/right regions often exclude the human subject); for home images we consider all four candidates.

(iii) Finally, as the cropped scene should be reasonably large so as to include the key context, we discard scene-product pairs for which the area of the cropped image is smaller than a threshold (we use 1/5 of the full area). If the cropped image is large enough, the pair containing the cropped scene and the corresponding product is included in the CTL dataset.

Following our heuristic cropping strategy, we manually verify that in most cases the cropped image doesn't include the associated product, and the cropped image contains a meaningful and reasonable region for predicting complements. In practice we find that discarded instances are largely due to dresses which often occupy a large area. We find that for complement recommendation it is generally not practical to apply a cropping strategy for objects that occupy a large portion of the image; therefore we opted simply to discard dresses from our dataset (note that we can still recommend other fashion items based on scenes in which people wear dresses). Figure 2 shows a comparison between STL and CTL data; CTL data statistics are listed in Table 1.

4. Method

In the *Complete the Look* task, we are given a dataset containing compatible pairs consisting of a scene image I_s and a product image I_p (as shown in Figure 2), and seek to learn scene-product style compatibility. To this end, we design a model which measures the compatibility globally in addition to a more fine-grained approach that matches relevant regions of the scene image with the product image.

4.1. Style Embeddings

We adopt ResNet-50 [12] to extract visual features from scene and product images. Based on the scene image I_s, we obtain a visual feature vector $\mathbf{v}_s \in \mathbb{R}^{d1}$ from the final layers (e.g. `pool5`), and a feature map $\{\mathbf{m}_i \in \mathbb{R}^{d2}\}_{i=1}^{w \times h}$ from intermediate convolutional layers (e.g. `conv4_6`). Similarly, the visual feature for product image I_p is denoted as $\mathbf{v}_p \in \mathbb{R}^{d1}$. Such ResNet feature vectors have shown strong performance and transferability [12, 21], and the feature maps have been shown to be able to capture key context from local regions [31, 49].

Due to the limited size of our datasets, we freeze the weights of ResNet-50 (pretrained on Imagenet) and apply a two-layer feed forward network $g(\Theta; \cdot)$[4] to transform the visual features to a d-dimensional metric embedding (with unit length) in a unified style space. Specifically, we have:

$$\begin{aligned} \mathbf{f}_s &= g(\Theta_g; \mathbf{v}_s), \mathbf{f}_p = g(\Theta_g; \mathbf{v}_p), \\ \mathbf{f}_i &= g(\Theta_l; \mathbf{m}_i), \hat{\mathbf{f}}_i = g(\Theta_{\hat{l}}; \mathbf{m}_i), \end{aligned} \tag{1}$$

where $\mathbf{f}_s$ and $\mathbf{f}_p$ are the style embedding for the scene and the product respectively, and $\mathbf{f}_i, \hat{\mathbf{f}}_i$ are embeddings for the i-th region of the scene image. ℓ_2 normalization is applied on embeddings to improve training stability, an approach commonly used in recent work on deep embedding learning [33, 47].

4.2. Measuring Compatibility

We measure compatibility by considering both global and local compatibility in a unified style space.

Global compatibility. We seek to learn style embeddings from compatible scene and product images, where nearby embeddings indicate high compatibility. We use the (squared) ℓ_2 distance between the scene embedding $\mathbf{f}_s$ and the product embedding $\mathbf{f}_p$ to measure their global compatibility:

$$d_{\text{global}}(s, p) = \|\mathbf{f}_s - \mathbf{f}_p\|^2, \tag{2}$$

where $\|\cdot\|$ is the ℓ_2 distance.

[4] The network architecture is `Linear-BN-Relu-Dropout-Linear-L2Norm`, and parameterized by Θ.

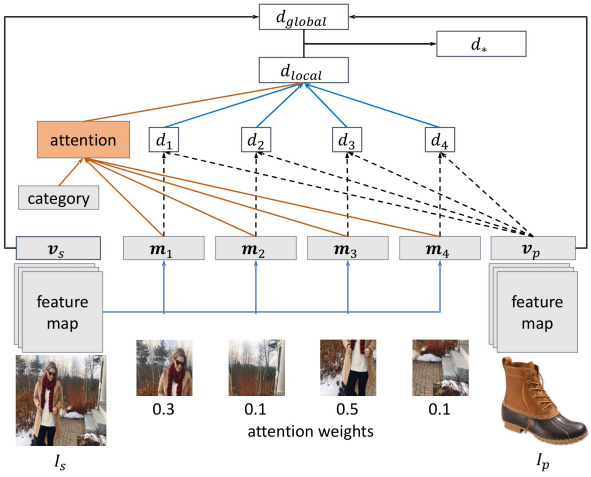


Figure 3: An illustration of our hybrid compatibility measurement. We simplify the size of the attention map to 2×2.

Local Compatibility. As the scene image typically contains a large area including many objects, considering only global compatibility may overlook key details in the scene. Hence we match every region of the scene image with the product image to achieve a more fine-grained matching procedure. Moreover, not all regions are equally relevant, and relevance may vary when predicting complementarity from different categories. Thus, we first measure the compatibility between every scene patch and the product, and then adopt category-aware attention to assign weights over all regions:

$$d_{\text{local}}(s,p) = \sum_{1\le i\le w\times h} a_i \|\mathbf{f}_i - \mathbf{f}_p\|^2, \\ \hat{a}_i = -\|\hat{\mathbf{f}}_i - \hat{\mathbf{e}}_c\|^2,\ \mathbf{a} = \text{softmax}(\hat{\mathbf{a}}), \tag{3}$$

where c is the category of product p, and $\hat{\mathbf{e}}_c \in \mathbb{R}^d$ is an ℓ_2-normalized embedding for category c. Here, we use the distance between $\hat{\mathbf{f}}_i$ and $\hat{\mathbf{e}}_c$ to measure the relevance of the i-th region of the scene image when predicting complements from category c. Note the 'attentive distances' in eq. 3 can be viewed as an extension of attention for metric embeddings, as if we replace the ℓ_2 distance with an inner product we recover the conventional attention form $(\sum_i a_i \mathbf{f}_i)^T \mathbf{f}_p$.

Finally, we measure compatibility by defining a hybrid distance that combines both global and local distances:

$$d_*(s,p) = \frac{1}{2}\left[d_{\text{global}}(s,p) + d_{\text{local}}(s,p)\right]. \tag{4}$$

Figure 3 illustrates our scene-product compatibility measuring procedure. Recall that all the embeddings we used are normalized to have unit length, and attention weights are also normalized (i.e., $\sum_i a_i = 1$). Note that all the distances (d_*, d_{glocal}, and d_{local}) range from 0 to 4.

Figure 4: A sample of the binary questions in our testing sets. Given a scene and two products, the model must predict which product is more compatible with the scene. Correct predictions are labeled in green, incorrect in red.

4.3. Objective

Following [33], we adopt the hinge loss to learn style embeddings by considering triplets $\mathcal{T}$ of a scene s, a positive product p^+, and a negative product p^-:

$$\mathcal{L} = \sum_{(s,p^+,p^-)\in\mathcal{T}} \left[d_*(s,p^+) - d_*(s,p^-) + \alpha\right]_+, \tag{5}$$

where α is the margin, which we set to 0.2 as suggested in [33]. Storing all possible triplets in $\mathcal{T}$ is intractable; we use mini-batch gradient descent to optimize the model, where training triplets for each batch are dynamically generated: we first randomly sample (s,p^+) from all compatible pairs, and then sample a negative product p^- from the same category of p^+. We do not sample negatives from different categories, as during testing we rank products from the same category, which is what the adopted sampling strategy seeks to optimize. For more detail refer to Section 5.2, and to our supplementary material.

5. Experiments

We first compare our approach against existing methods that are designed for predicting fashion compatibility between products. We then study the behavior of the attention mechanism. Finally, we perform a human study and show practical use-cases of our approaches.

5.1. Baselines

Popularity: A simple baseline which recommends products based on their popularity (i.e., the number of associated $(scene, product)$ pairs).

Imagenet Features: We directly use visual features from ResNet pretrained on Imagenet, which have shown strong performance in terms of retrieving visually similar images [30, 16]. The similarity is measured via the ℓ_2 distance between embeddings.

IBR [29]: Image-based recommendation (IBR) measures product compatibility via a learned Mahalanobis distance between visual embeddings. Essentially IBR learns a linear transformation to convert visual features into a style space.

Siamese Nets: Veit et al. [42] adopt Siamese CNNs [7] to learn style embeddings from product images, and measure their compatibility using an ℓ_2 distance. As suggested in [42], we fine-tune the network based on a pretrained model.

BPR-DAE [38]: This method uses autoencoders to extract representations from clothing images and textual descriptions, and incorporates them into the BPR recommendation framework [32]. Due to the absence of textual information in our datasets, we only use its visual module.

Since the baselines above are designed for measuring product compatibility, we adapt the baselines to our problem by treating all images as product images and apply the same sampling strategy as used in our method.

5.2. Implementation Details

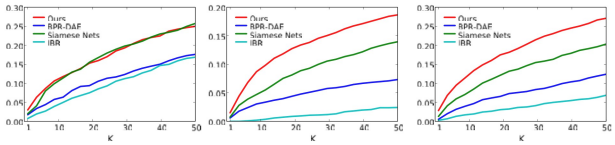
For a fair comparison, we implemented all methods using ResNet-50[5] (pretrained on Imagenet), as the underlying network, where the layer `pool5` (2048d) is used for the visual vectors and the layer `block3` (7×7×1024) is used as the feature map. We use an embedding size of 128, and we did not observe any performance gain with larger d (e.g. $d = 512$). All models are trained using *Adam* [20] with a batch size of 16. As suggested in [33], visual embeddings are normalized to have a unit length for metric embedding based methods, and the margin α is set to 0.2. For all methods, horizontal mirroring and random 224×224 crops from 256×256 images are used for data augmentation, and a single center crop is used for testing. We randomly split the scenes (and the associated pairs) into training (80%), validation (10%) and test (10%) sets. We train all methods for 100 epochs, examine the performance on the validation set every 10 epochs, and report the test performance for the model which achieves the best validation performance.

5.3. Recommendation Performance

As shown in Figure 4, given a scene s, a category c, a positive product p^+, and a negative product p^- (randomly sampled from c), the model needs to decide which product is more compatible with the scene image. The accuracy of these binary choice problems is used as a metric for performance evaluation. Note the accuracy here is equivalent to the AUC which measures the overall ranking performance.

Table 2 lists the accuracy of all methods. First, we note that the first group of methods (learning-free) perform poorly. Imagenet features perform similarly to random guessing, which indicates that visual compatibility is different from visual similarity, and thus it is necessary to learn the notion of compatibility from data. Even the naïve popularity baseline achieves better (though still poor) performance. Second, we found training with cropped images is effective as it can generally boost the performance compared with using full images. Compared to other baselines, our method has the most significant performance drop with full images, presumably because our method is the only one which is aware of local appearance, which makes it easier to erroneously match scene patches with the product rather than leaning compatibility (as discussed in Section 3.2). The performance gap between Fashion-1 and Fashion-2 possibly relates to their sizes. Finally, our method achieves the best performance on all datasets for both the fashion and home domains.

[5]We use the implementation from TensorFlow-Slim: https://github.com/tensorflow/models/tree/master/research/slim. The architecture is slightly different from the original paper (e.g. different strides).



(a) **Fashion-1** (b) **Fashion-2** (c) **Home**

Figure 5: Top-K Accuracy on all datasets (i.e., how often the top-K retrieved items contain the ground-truth product).

Method	Fashion-1	Fashion-2	Home
Random	50.0	50.0	50.0
Popularity	52.1	57.5	55.6
Imagenet Feature	49.4	51.6	48.1
Train w/ full images			
IBR [29]	56.5	58.5	57.0
Siamese Nets [42]	63.0	67.1	72.4
BPR-DAE [38]	59.3	61.1	64.2
Ours	63.1	70.0	75.0
Train w/ cropped images			
IBR [29]	54.5	55.9	58.0
Siamese Nets [42]	64.0	69.0	73.1
BPR-DAE [38]	59.6	61.1	65.8
Ours	**68.5**	**75.3**	**79.6**

Table 2: Accuracy of binary comparisons on all datasets.

In addition to an overall ranking measurement, the Top-K accuracy (the fraction of times that the first K recommended items contain the positive item) [1] might be closer to a practical scenario. Figure 5 shows Top-K accuracy curves for all datasets. We see that our method significantly outperforms baselines on the last two datasets, and slightly outperforms the strongest baseline on the first dataset. Performance analysis on additional model variants is included in our supplementary material.

5.4. Analysis of Attended Regions

We visualize the attended areas to intuitively reveal what parts of a scene image are important for predicting complementarity, and quantitatively evaluate whether the attention focuses on meaningful regions.

Figure 6 shows test scene images (after cropping), the corresponding attention map from our model, and the saliency

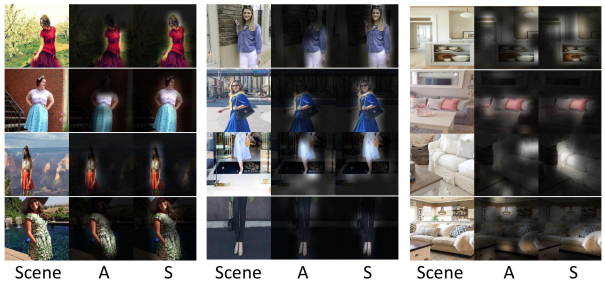


Figure 6: Visualization of attention maps ('A') from our method, and saliency maps ('S') from DeepSaliency [22].

Method	Fashion-1	Fashion-2	Home
Top-1 region			
Random	13.2	12.3	16.4
Attention (Ours)	22.4	24.4	18.9
DeepSaliency [22]	24.8	25.0	17.8
Top-3 regions			
Random	32.1	29.9	37.0
Attention (Ours)	43.3	45.0	38.3
DeepSaliency [22]	49.2	47.8	36.8

Table 3: Fraction of successful hits on meaningful regions.

map generated by DeepSaliency[6] [22]. DeepSaliency is trained to detect salient objects while our attention mechanism discovers relevant areas by learning the compatibility between scene and product images. For the first two fashion datasets, the two approaches both successfully identify the subject of the image (i.e., the person) from various backgrounds. Interestingly, our attention mechanism tends to ignore human faces and more focus only on clothing, which means our model discovers that the subject's clothing is more relevant than the appearance of the subject themselves when recommending complements.[7] In contrast, the scenes in the interior design domain are much more complex (critically, they contain many objects rather than a single subject). Although some meaningful objects (e.g. pillows, lamps, etc.) are discovered in some cases, it appears to be harder for either attention or saliency to detect key objects.

In addition to qualitative examples, we also quantitatively measure whether attention focuses on meaningful areas of scene images. Here we assume that areas corresponding to labeled products are relevant. Specifically, we divide the image into 7×7 regions, and a region is considered 'relevant' if it significantly overlaps (i.e. larger than half the area of a region) with any product's bounding box. We then calculate the attention map (7×7) for all test scene images and rank the 49 regions according to their scores. If the top-1 region (or one of the top-3 regions) is a 'relevant region', then we deem the attention map as a successful hit.

Table 3 shows the fraction of successful hits of our attention map, random region ranking, and the saliency map from DeepSaliency [22]. On the fashion datasets, our method's performance is close to that of DeepSaliency, and both methods are significantly better than random. This shows that our attention mechanism can discover and focus on key objects (without knowing what area is relevant during training) guided by the supervision of complementarity. This is similar to a recent study which shows that spatial attention seems to be good at extracting key areas for clothing category prediction [45]. However, for the 'home' domain, both our method and DeepSaliency perform only slightly better than (or similar to) random. This indicates that scene images in the home domain are more complex than fashion images, as shown in Figure 6. This may also imply that a more sophisticated method (e.g. object detection) might be needed to extract local patterns for the home domain.

Method	Dataset	Overlap	Human
Popularity	56.5	60.7	56.0
IBR [29]	56.3	56.2	52.9
Siamese Nets [42]	72.1	72.6	62.1
BPR-DAE [38]	62.5	63.8	58.3
Ours	**75.8**	**77.3**	**65.0**
Human	75.0	100	100

Table 4: Accuracy on sampled data using dataset labels, human labels, and overlap labels as ground truth (respectively).

5.5. Human Performance

To assess how well the learned models accord with human fashion sense, we conduct a human subject evaluation, in which four fashion experts are asked to respond to binary choice questions (as shown in Figure 4). Specifically, each fashion expert is required to label 20 questions (randomly sampled from the test set) for each dataset, and performance is then evaluated based on the 240 labeled questions.

An important observation is that our model achieves 'human-level' performance: the second column of Table 4 ("Dataset") shows that fashion experts achieve 75.0% accuracy, while our model achieves 75.8% accuracy.

Considering that fashion experts sometimes disagree with the ground-truth, we use the fashion experts' judgments as the ground-truth labels to evaluate the consistency with human fashion sense, and our model outperforms other methods (fourth column of Table 4, "Human"). We also observe that better performance on the dataset typically implies a better consistency with human fashion sense.

A final question is related to the subjectivity of the task: how well does our model do on cases where there is a clear answer? To answer this question, we used the following heuristic to generate a dataset consisting of only unambigu-

[6] http://www.zhaoliming.net/research/deepsaliency

[7] Note that attention is only used when measuring local compatibility, the model can still leverage the context provided in the unattended regions via the global compatibility.

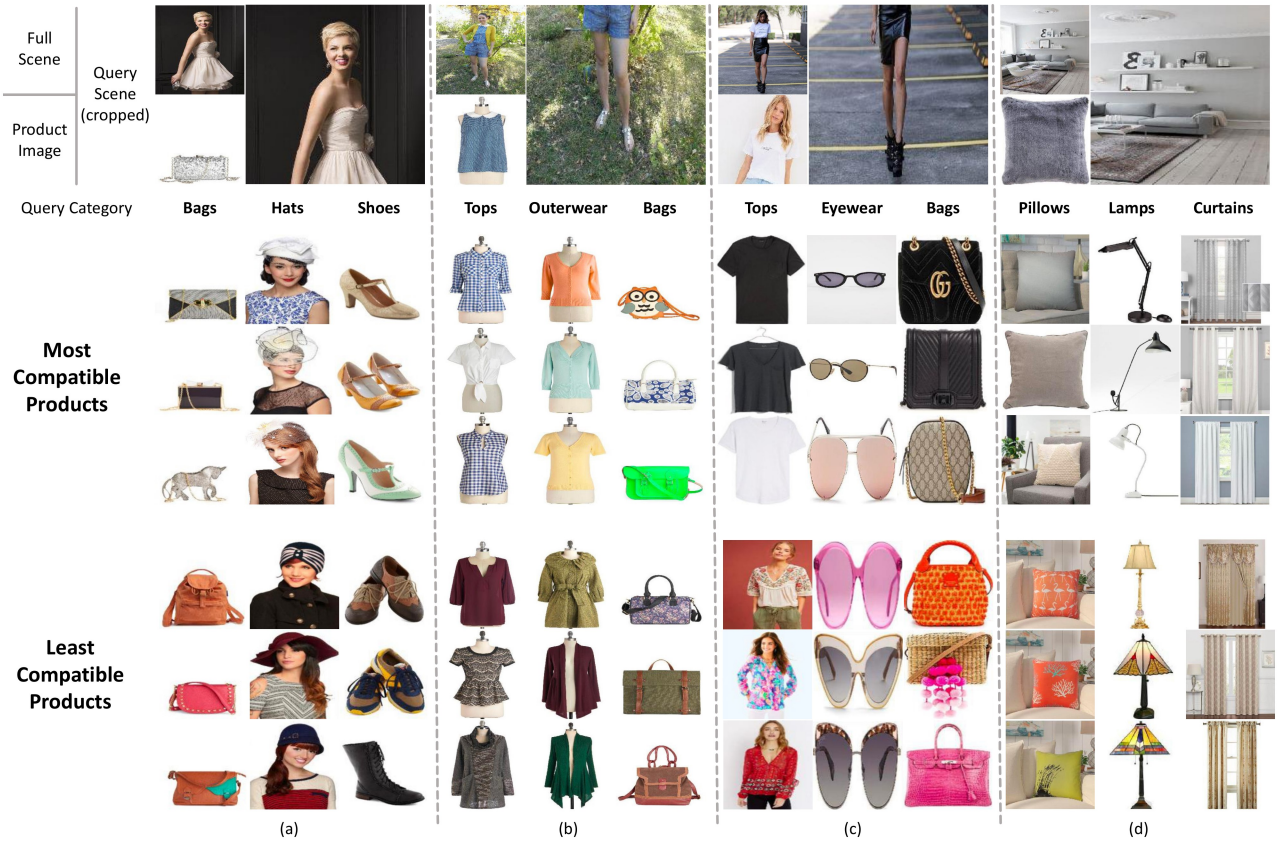


Figure 7: Qualitative results of the top-3 most and least compatible products generated by our model.

ous questions: select the test data where both fashion experts and the dataset label agree. On this data subset, our model is again the top-performer, which shows that our model indeed produces better fashion recommendations than other approaches, even when controlling for question ambiguity (third column of Table 4, "Overlap").

5.6. Qualitative Results

Figure 7 shows four examples (from the test set) that depict the original scene, the cropped product, query scene, query category, and the top-3 most and least compatible products selected by our algorithm. By comparing the removed product and the retrieved products from the first category (i.e., the same category as the removed one), it appears that the most compatible items are closer in style to the ground-truth product compared with the least compatible items. Qualitatively speaking, the generated compatible products are more compatible with the scenes. In column (a), the recommended white and transparent hats are (in the authors' opinion) more compatible than dark colors; in column (b), the yellow outerwear from the full scene is close in color and style with the recommendations. We also observe that the learned compatibility is not merely based on simple factors like color; for example, in column (d) the recommended lamps have different colors but similar style (modern, minimalist), and are quite different in style from the incompatible items. Thus the model appears to have learned a complex notion of style.

6. Conclusion

In this paper, we proposed a novel task, *Complete the Look*, for recommending complementary products given a real-world scene. Complete the Look (or CTL) can be straightforwardly applied on e-commerce websites to give users fashion advice, simply by providing scene images as input. We designed a cropping-based approach to construct CTL datasets from STL (Shop the Look) data. We estimate scene-product compatibility globally and locally via a unified style space. We performed extensive experiments on recommendation performance to verify the effectiveness of our method. We further studied the behavior of our attention mechanism across different domains, and conducted human evaluation to understand the ambiguity and difficulty of the task. Qualitatively, our CTL method generates compatible recommendations that seem to capture a complex notion of 'style.' In the future, we plan to incorporate object detection techniques to extract key objects for compatibility matching.

Acknowledgement: The authors would like to thank Ruining He, Zhengqin Li, Larkin Brown, Zhefei Yu, Kaifeng Chen, Jen Chan, Seth Park, Aimee Rancer, Andrew Zhai, Bo Zhao, Ruimin Zhu, Cindy Zhang, Jean Yang, Mengchao Zhong, Michael Feng, Dmitry Kislyuk, and Chen Chen for their help in this work.

Supplementary Material

A. Performance Analysis

A.1. Ablation Study

We perform an ablation study to analyze the effect of the global and local compatibility measuring components. Table 5 shows the accuracy of our method and three variants on all datasets. 'L' and 'G' represents the variants with only the local component and the global component (respectively). We also exam the performance of the variant 'G+L^0' which assigns equal weights on each region (instead of using attention weights). 'G+L' is the default model which uses both components and attention weights. We can see the hybrid model 'G+L' is better than using the two components individually. Also, the attention is helpful as it can boost performance compared with 'G+L^0'.

Method	Fashion-1	Fashion-2	Home
L	67.6	73.8	77.1
G	66.9	74.2	77.8
G+L^0	66.9	74.1	78.6
G+L (Default)	**68.5**	**75.3**	**79.6**

Table 5: Ablation Study

A.2. The Effect of Local Features

As our method utilizes intermediate feature maps as local features, we exam the effect of using features from different layers and networks. In addition to different blocks of ResNet-50 [12], we also consider the VGG-16 [37] network where the `fc7` layer is used as the visual feature vector. In Table 6, we can observe that the ResNet-50 network with the `block3` feature achieves the best performance on all three datasets. Hence we choose to use the `block3` layer by default. Moreover, we find using VGG features is generally worse than using ResNet features.

A.3. Performance on Each Category

We exam the performance on each category, compared with the strongest baseline Siamese Nets [42]. Figure 8 shows the performance comparison per category, based on the STL-Fashion and STL-Home datasets. We can see that in most categories, our method outperforms Siamese Nets. Moreover, the baseline has severe performance drops on categories like skirts and curtains. This verifies the effectiveness of our method when recommending products from various categories.

Local Feature	Fashion-1	Fashion-2	Home
ResNet-50			
`block1` (28×28×256)	68.0	74.5	77.7
`block2` (14×14×512)	67.6	73.3	78.3
`block3` (7×7×1024)	**68.5**	**75.3**	**79.6**
`block4` (7×7×2048)	64.9	74.2	78.5
VGG-16			
`pool3` (28×28×512)	64.0	71.1	76.2
`pool4` (14×14×512)	63.3	71.7	76.7
`pool5` (7×7×512)	63.1	72.0	75.5

Table 6: The effect of local features

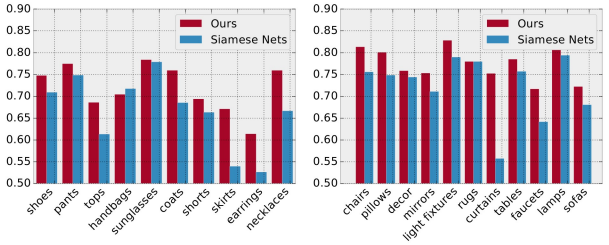


Figure 8: Accuracy per category. Left: STL-Fashion, right: STL-Home.

B. Implementation Details

The architecture of the two-layer network $g(\Theta;\cdot)$ is `Linear-BN-Relu-Dropout-Linear-L2Norm`, where the dimensionality is set to $4\times d$ for the first linear layer, and d for the last linear layer. The dropout rate is set to 0.5, and the learning rate is set to 0.001. For all methods, we update the statistics of batch normalization [14] layers in ResNet during training, and find it generally improves the performance.

C. Human Study Interface

Figure 9 shows screenshots of the interface when conducting the user study. We first provide a description of the task and a small sample of tests to the fashion experts, and then ask them to answer the questions (i.e., choose one of the two products that is more compatible with the given scene).

References

[1] K. E. Ak, A. A. Kassim, J. H. Lim, and J. Y. Tham. Learning attribute representations with localization for flexible fashion search. In *CVPR*, 2018.

[2] K. E. Ak, J.-H. Lim, J. Y. Tham, and A. A. Kassim. Efficient multi-attribute similarity learning towards attribute-based fashion search. In *WACV*, 2018.

[3] Z. Al-Halah, R. Stiefelhagen, and K. Grauman. Fashion forward: Forecasting visual style in fashion. In *ICCV*, 2017.

[4] L. Bossard, M. Dantone, C. Leistner, C. Wengert, T. Quack, and L. J. V. Gool. Apparel classification with style. In *ACCV*, 2012.

[5] L. Chen, H. Zhang, J. Xiao, L. Nie, J. Shao, W. Liu, and T.-S. Chua. SCA-CNN: spatial and channel-wise attention in convolutional networks for image captioning. In *CVPR*, 2017.

[6] Z.-Q. Cheng, X. Wu, Y. Liu, and X.-S. Hua. Video2shop: Exact matching clothes in videos to online shopping images. In *CVPR*, 2017.

[7] S. Chopra, R. Hadsell, and Y. LeCun. Learning a similarity metric discriminatively, with application to face verification. In *CVPR*, 2005.

[8] W. Di, C. Wah, A. Bhardwaj, R. Piramuthu, and N. Sundaresan. Style finder: Fine-grained clothing style detection and retrieval. In *CVPRW*, 2013.

[9] R. Hadsell, S. Chopra, and Y. LeCun. Dimensionality reduction by learning an invariant mapping. In *CVPR*, 2006.

[10] X. Han, Z. Wu, P. X. Huang, X. Zhang, M. Zhu, Y. Li, Y. Zhao, and L. S. Davis. Automatic spatially-aware fashion concept discovery. In *CVPR*, 2017.

[11] X. Han, Z. Wu, Y.-G. Jiang, and L. S. Davis. Learning fashion compatibility with bidirectional lstms. In *MM*, 2017.

[12] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[13] W.-L. Hsiao and K. Grauman. Creating capsule wardrobes from fashion images. In *CVPR*, 2018.

[14] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, 2015.

[15] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu. Spatial transformer networks. In *NIPS*, 2015.

[16] Y. Jing, D. C. Liu, D. Kislyuk, A. Zhai, J. Xu, J. Donahue, and S. Tavel. Visual search at pinterest. In *SIGKDD*, 2015.

[17] W.-C. Kang, C. Fang, Z. Wang, and J. McAuley. Visually-aware fashion recommendation and design with generative image models. In *ICDM*, 2017.

[18] M. H. Kiapour, X. Han, S. Lazebnik, A. C. Berg, and T. L. Berg. Where to buy it: Matching street clothing photos in online shops. In *ICCV*, 2015.

[19] M. H. Kiapour, K. Yamaguchi, A. C. Berg, and T. L. Berg. Hipster wars: Discovering elements of fashion styles. In *ECCV*, 2014.

[20] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

[21] S. Kornblith, J. Shlens, and Q. V. Le. Do better imagenet models transfer better? *arXiv preprint arXiv:1805.08974*, 2018.

[22] X. Li, L. Zhao, L. Wei, M.-H. Yang, F. Wu, Y. Zhuang, H. Ling, and J. Wang. Deepsaliency: Multi-task deep neural network model for salient object detection. *IEEE TIP*, 2016.

[23] Y. Li, L. Cao, J. Zhu, and J. Luo. Mining fashion outfit composition using an end-to-end deep learning approach on set data. *IEEE TMM*, 2017.

[24] X. Liang, L. Lin, W. Yang, P. Luo, J. Huang, and S. Yan. Clothes co-parsing via joint image segmentation and labeling with application to clothing retrieval. *IEEE TMM*, 2016.

[25] S. Liu, J. Feng, Z. Song, T. Zhang, H. Lu, C. Xu, and S. Yan. Hi, magic closet, tell me what to wear! In *MM*, 2012.

[26] S. Liu, Z. Song, G. Liu, C. Xu, H. Lu, and S. Yan. Street-to-shop: Cross-scenario clothing retrieval via parts alignment and auxiliary set. In *CVPR*, 2012.

[27] Z. Liu, P. Luo, S. Qiu, X. Wang, and X. Tang. Deepfashion: Powering robust clothes recognition and retrieval with rich annotations. In *CVPR*, 2016.

[28] J. J. McAuley, R. Pandey, and J. Leskovec. Inferring networks of substitutable and complementary products. In *SIGKDD*, 2015.

[29] J. J. McAuley, C. Targett, Q. Shi, and A. van den Hengel. Image-based recommendations on styles and substitutes. In *SIGIR*, 2015.

[30] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson. CNN features off-the-shelf: An astounding baseline for recognition. In *CVPRW*, 2014.

[31] S. Ren, K. He, R. B. Girshick, and J. Sun. Faster R-CNN: towards real-time object detection with region proposal networks. In *NIPS*, 2015.

[32] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme. BPR: bayesian personalized ranking from implicit feedback. In *UAI*, 2009.

[33] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. In *CVPR*, 2015.

[34] K. J. Shih, S. Singh, and D. Hoiem. Where to look: Focus regions for visual question answering. In *CVPR*, 2016.

[35] Y.-S. Shih, K.-Y. Chang, H.-T. Lin, and M. Sun. Compatibility family learning for item recommendation and generation. In *AAAI*, 2018.

[36] E. Simo-Serra and H. Ishikawa. Fashion style in 128 floats: Joint ranking and classification using weak data for feature extraction. In *CVPR*, 2016.

[37] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.

[38] X. Song, F. Feng, J. Liu, Z. Li, L. Nie, and J. Ma. Neurostylist: Neural compatibility modeling for clothing matching. In *MM*, 2017.

[39] Z. Song, M. Wang, X.-s. Hua, and S. Yan. Predicting occupation via human clothing and contexts. In *ICCV*, 2011.

[40] P. Tangseng, K. Yamaguchi, and T. Okatani. Recommending outfits from personal closet. In *WACV*, 2018.

[41] M. I. Vasileva, B. A. Plummer, K. Dusad, S. Rajpal, R. Kumar, and D. A. Forsyth. Learning type-aware embeddings for fashion compatibility. In *ECCV*, 2018.

[42] A. Veit, B. Kovacs, S. Bell, J. McAuley, K. Bala, and S. J. Belongie. Learning visual clothing style with heterogeneous dyadic co-occurrences. In *ICCV*, 2015.

[43] F. Wang, M. Jiang, C. Qian, S. Yang, C. Li, H. Zhang, X. Wang, and X. Tang. Residual attention network for image classification. In *CVPR*, 2017.

[44] J. Wang, Y. Song, T. Leung, C. Rosenberg, J. Wang, J. Philbin, B. Chen, and Y. Wu. Learning fine-grained image similarity with deep ranking. In *CVPR*, 2014.

[45] W. Wang, Y. Xu, J. Shen, and S.-C. Zhu. Attentive fashion grammar network for fashion landmark detection and clothing category classification. In *CVPR*, 2018.

[46] Z. Wang, Z. Jiang, Z. Ren, J. Tang, and D. Yin. A path-constrained framework for discriminating substitutable and complementary products in e-commerce. In *WSDM*, 2018.

[47] C.-Y. Wu, R. Manmatha, A. J. Smola, and P. Krähenbühl. Sampling matters in deep embedding learning. In *ICCV*, 2017.

[48] H. Xu and K. Saenko. Ask, attend and answer: Exploring question-guided spatial attention for visual question answering. In *ECCV*, 2016.

[49] K. Xu, J. Ba, R. Kiros, K. Cho, A. C. Courville, R. Salakhutdinov, R. S. Zemel, and Y. Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *ICML*, 2015.

[50] T. Xu, P. Zhang, Q. Huang, H. Zhang, Z. Gan, X. Huang, and X. He. Attngan: Fine-grained text to image generation with attentional generative adversarial networks. In *CVPR*, 2018.

[51] K. Yamaguchi, M. H. Kiapour, L. E. Ortiz, and T. L. Berg. Parsing clothing in fashion photographs. In *CVPR*, 2012.

[52] K. Yamaguchi, M. H. Kiapour, L. E. Ortiz, and T. L. Berg. Retrieving similar styles to parse clothing. *IEEE TPAMI*, 2015.

[53] F. Yang, A. Kale, Y. Bubnov, L. Stein, Q. Wang, M. H. Kiapour, and R. Piramuthu. Visual search at ebay. In *SIGKDD*, 2017.

[54] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena. Self-attention generative adversarial networks. *arXiv preprint arXiv:1805.08318*, 2018.

[55] Y. Zhang, H. Lu, W. Niu, and J. Caverlee. Quality-aware neural complementary item recommendation. In *RecSys*, 2018.

[56] Y. Zhang, P. Pan, Y. Zheng, K. Zhao, Y. Zhang, X. Ren, and R. Jin. Visual search at alibaba. In *SIGKDD*, 2018.

[57] S. Zheng, F. Yang, M. H. Kiapour, and R. Piramuthu. Modanet: A large-scale street fashion dataset with polygon annotations. In *MM*, 2018.

[58] B. Zhou, A. Khosla, À. Lapedriza, A. Oliva, and A. Torralba. Learning deep features for discriminative localization. In *CVPR*, 2016.

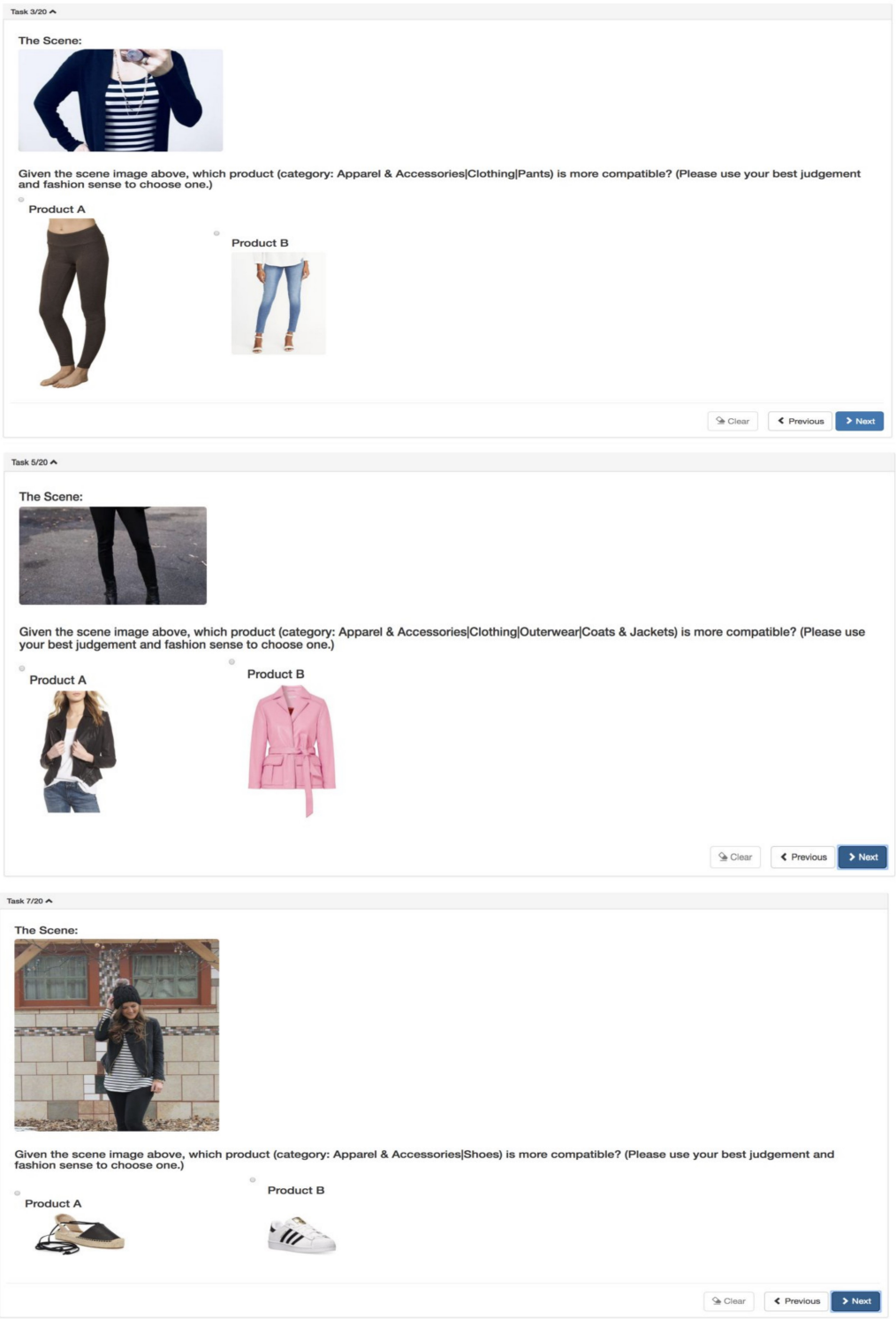


Figure 9: Screenshots of the user study interface.