Disentangling Features for Fashion Recommendation

LAVINIA DE DIVITIIS, FEDERICO BECATTINI, CLAUDIO BAECCHI, and ALBERTO DEL BIMBO, University of Florence, Italy

Online stores have become fundamental for the fashion industry, revolving around recommendation systems to suggest appropriate items to customers. Such recommendations often suffer from a lack of diversity and propose items that are similar to previous purchases of a user. Recently, a novel kind of approach based on Memory Augmented Neural Networks (MANNs) has been proposed, aimed at recommending a variety of garments to create an outfit by complementing a given fashion item. In this article we address the task of compatible garment recommendation developing a MANN architecture by taking into account the co-occurrence of clothing attributes, such as shape and color, to compose an outfit. To this end we obtain disentangled representations of fashion items and store them in external memory modules, used to guide recommendations at inference time. We show that our disentangled representations are able to achieve significantly better performance compared to the state of the art and also provide interpretable latent spaces, giving a qualitative explanation of the recommendations.

CCS Concepts: • **Networks** → **Network architectures**; • **Information systems** → **Information retrieval**; • **Computing methodologies** → **Machine learning**; • **Human-centered computing** → **Visualization;**

Additional Key Words and Phrases: Garment recommendation, memory augmented neural networks, recommendation systems

ACM Reference format:
Lavinia De Divitiis, Federico Becattini, Claudio Baecchi, and Alberto Del Bimbo. 2023. Disentangling Features for Fashion Recommendation. *ACM Trans. Multimedia Comput. Commun. Appl.* 19, 1s, Article 39 (January 2023), 21 pages.
https://doi.org/10.1145/3531017

1 INTRODUCTION

With a gross sale of over $3,000 billion, the fashion industry covers 2% of the world's **Gross Domestic Product (GDP)**.[1] These numbers are possible thanks to a thriving industry that sells and promotes fashion items all over the world, always renovating and rethinking itself. For this reason, captivating customers has become an essential part of the business process, as they are an essential asset. It is thus of paramount importance that they should not only be offered a satisfactory

[1] https://fashionunited.com/global-fashion-industry-statistics/.

This work was partially supported by the Italian MIUR within PRIN 2017, Project Grant 20172BH297: I-MALL - improving the customer experience in stores by intelligent computer vision.
Authors' address: L. De Divitiis, F. Becattini, C. Baecchi, and A. D. Bimbo, University of Florence, Florence, Italy; emails: {lavinia.dedivitiis, federico.becattini, claudio.baecchi, alberto.delbimbo}@unifi.it.

1551-6857/2023/01-ART39 $15.00
https://doi.org/10.1145/3531017

selling process but also accompanied in exploring and discovering new products that may be of their interest among the huge catalogs available both in stores and online.

For these reasons, the fashion industry constantly strives to engage customers into discovering new products. From mass advertising to personalized offers, multimedia systems are exploited to spark interest in the final user, with the intent of selling specific products. On the other hand, customers themselves often require assistance for discovering new outfits or identifying garments compatible with previously purchased items. This sort of aid may stem from employees in physical shops, but it must be replaced by automatic recommendation algorithms in online stores, which nowadays are the principal source of income for fashion companies.

An easy way to recommend garments is to follow trends or suggest items according to user preferences. This will likely yield recommendations capable of engaging lots of users in the short term, but will also keep suggesting similar items over and over. For effective long-term recommendations, suggested items must be variegated and cover different styles to meet temporary changes in customer tastes or be suitable for different social occasions. With this in mind, we formulate the problem of garment recommendation as the task of suggesting dressing modalities rather than exact fashion items out of a given collection. In fact, we want to guarantee diversity instead of proposing multiple similar outfits that only differ by small details.

In particular, in this article we address the problem of recommending compatible complementary clothing items that compose an outfit, e.g., identifying a set of bottoms that can be paired with a given top (Figure 1). To ensure diversity we rely on two strategies. First, we learn disentangled representations for shape and color using a self-supervised contrastive learning approach; then, we train two **Memory Augmented Neural Networks (MANNs)** [10, 20, 22, 24, 30, 35] to identify and store pairing modalities for shapes and colors separately. We exploit MANNs to model general garment compatibility and then, only after having identified different pairing modalities, we refine results including fashion trends to make recommendations.

The idea of using MANNs for fashion recommendation has been recently explored in [7], demonstrating promising capabilities thanks to the ability to match relevant pairs of garments that compose an outfit at training time and then making this information part of the recommendation process at inference time. Here, we extend this idea by proposing an improved memory module with an adaptive controller and separate memory banks to identify pairing modalities for different attributes, such as shape and color. This is made possible by our disentangled feature learning approach.

The task of compatible garment recommendation, which we address in this article, is closely related to the one of outfit compatibility estimation. This has been previously addressed in the literature by several works [26, 27, 29, 33]. The two settings differ since compatibility estimation establishes if two or more given items fit well together, whereas compatible garment recommendation proposes a ranked list of candidates that are compatible with a given item. This makes the recommendation task more challenging since a model must learn to select suitable garments among a large collection of fashion items. Nonetheless, we show how our model can be easily modified to perform compatibility estimation. In fact, our model naturally generates rankings for compatible garments, which can be used as a means to provide compatibility scores for outfits. Therefore, our model can be employed for a variety of tasks, i.e., to recommend garments, estimate compatibility between complementary garments, and retrieve compatible outfits among a set of candidates.

The main contributions of this article are the following:

- We exploit separate color and shape data augmentations while training our feature extractor as an autoencoder, in order to learn disentangled features relying only on self-supervision.

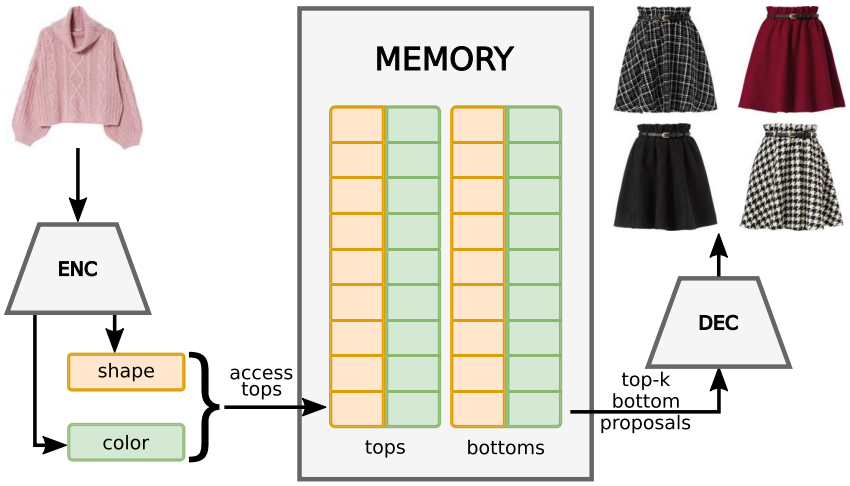

Fig. 1. Overview of our garment recommendation system. A memory network stores features extracted from the encoder part of a convolutional autoencoder (ENC). Top proposals are retrieved by decoding features using the decoder part of a convolutional autoencoder (DEC).

- We store non-redundant compatible pairing modalities in external memories, both for color and for shape. Our MANNs are equipped with a novel memory controller with an adaptive threshold, designed to write only a small fraction of representative samples.
- We demonstrate the effectiveness of combining disentangled features and external memories for tasks of compatible garment recommendation, outfit compatibility estimation, and complementary item retrieval, obtaining state-of-the-art results on two different datasets.

2 RELATED WORK

Given the great recent interest in customer recommendation, a lot of recent work among the scientific community has been focusing on estimating interest and performing recommendations in the fashion domain [3, 4, 14, 16, 23, 29]. In this work, we are interested in recommending fashion items that are compatible with a given complementary garment. For instance, given a top, we want to propose a ranked list of compatible bottoms that can be used to create an outfit.

A large crop of literature has studied how to model compatibility between fashion items when composing an outfit [5, 11, 26, 27, 29, 32, 33], although often declining the problem as compatibility estimation. Several of these works leverage a **Bayesian Personalized Ranking (BPR)** scheme to model compatibility between garments [17, 26–29]. The first to adopt such an approach was Song et al. [28] to overcome the limitations of matrix factorization due to excessive data sparsity. The approach was then extended by exploiting different strategies such as attentive knowledge distillation through a teacher-student network [27], personalized compatibility modeling including personal preferences [29], visual-textual multimodal learning [17], garment matching of labeled and unlabeled data with siamese networks [9], and attribute-wise interpretable compatibility scheme with personal preference modeling [26].

A different take on the problem has been provided by recent works trying to exploit contextual information from outfits including additional complementary garments and accessories. The collection of items composing an outfit has been either processed as a whole relying on graph-neural networks to learn context-conditioned item embeddings [5] or treated as a sequence using bidirectional LSTMs to iteratively predict the next compatible item based on previous ones. Additionally,

[31] proposes an outfit representation that is learned on both notions of similarity and inter-outfit compatibility, leveraging an image embedding that respects item types and jointly learns notions of item similarity and compatibility in an end-to-end model. In [32], instead, siamese networks are exploited to learn a visual notion of compatibility across categories and a feature transformation from images of items into a latent space that expresses compatibility. To further analyze the problem, an approach to not only predict but also diagnose outfit compatibility has been proposed in [33], where backpropagation gradients are used to identify the incompatible factors.

All the aforementioned approaches that address garment compatibility, however, focus on determining whether an outfit or a complementary garment is more compatible compared to another. Such models are thought to provide compatibility scores to rank outfits, rather than recommending a list of suitable items to complement a partial outfit. Recommending compatible items is indeed a more challenging task since the problem does not break down to comparing a few candidate outfits, but instead requires to identify suitable fashion items among a large collection of garments. This issue has recently been raised in [6, 7], where bottom garments are proposed to match an input top. The focus here shifts from modeling compatibility between garments to understanding possible dressing modalities to complement a top. In this article we follow such line of research. The problem has then been extended in [6] by leveraging emotive color information to give multiple recommendations that adhere to a desired style.

The most similar approach to ours is the one of De Divitiis et al. [7]. The authors propose to use a MANN as the central part of their garment recommendation system to pair compatible clothing items. The MANN is populated with a memory writing controller and trained to store a non-redundant subset of samples, which is then used to propose a ranked list of suitable bottoms to complement a given top. Similarly, we exploit a MANN but use two separate memory modules to store disentangled features for shape and color. This enables a more precise modeling of how fashion items can be worn together, breaking down the problem to how classes of garments can be paired and how colors can be combined. In addition, we also improve the memory controllers by adding a regularization term and exploiting an adaptive threshold to avoid trivial solutions that in the original formulation may occur when data is unbalanced. This leads to significant improvements compared to [7].

Aside from compatibility, research involving recommendation systems in the fashion domain has followed several promising directions. Several works have designed specific systems to model user preferences. For example, in [15], personalized outfit recommendation is achieved by suggesting sets of items through a functional tensor factorization method. This models the interactions between user and fashion items by using multi-modal features of fashion items and leveraging gradient-boosting-based methods to map the feature vectors into some low-dimensional latent space. He and McAuley [12] instead make use of visual features extracted from product images to build a scalable factorization model to incorporate visual signals into predictors of people's opinions. A consumer-oriented recommendation system by fuzzy techniques and **Analytic Hierarchy Process (AHP)** has been proposed in [37] to take into account consumers' perception on products. With efficiency in mind, [18] learns a binary code for efficient personalized fashion outfit recommendation using a set of type-dependent hashing modules to learn binary codes and a module that conducts pairwise matching. Focusing on the style, [13] creates a recommendation method that employs a pair of neural networks: a feedforward network generates article embeddings in "fashion space," which serves as input to a recurrent neural network that predicts a style vector in this space for each client, based on their past purchase sequence. Dynamic personalized recommendations have also been studied in [2], where user profiles are built with customers in the loop by analyzing facial reactions to recommended items.

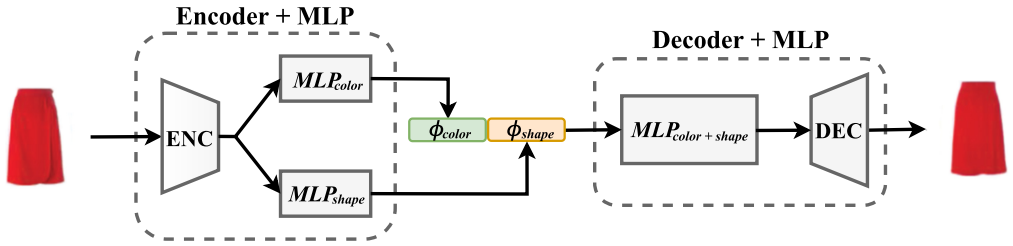

Fig. 2. The garment is encoded into two disentangled latent spaces, representing color and shape separately. The model is trained as an autoencoder to reconstruct the input image.

An interesting parallel line of research has focused on learning domain-specific features for fashion items. For instance, [36] focuses on aesthetic features, proposing a new tensor factorization model to incorporate such features in a personalized manner. To realize interpretable and customized fashion outfit compositions, [8] proposes a partitioned embedding network to learn interpretable representations from clothing items by leveraging an auto-encoder module, a supervised attributes module, and a multi-independent module to build an outfit composition graph and an attribute matching map. A pool or independent representation is learned by using attribute-specific classifiers in [14]. Such features are used to build attribute prototypes and perform attribute manipulation. Similarly to these works, we also propose to learn discriminative features to represent garments, but we rely on a self-supervised approach to disentangle color and shape latent spaces. We do not exploit manually annotated attributes; instead, we use specifically designed forms of data augmentation paired with a novel contrastive learning strategy.

3 COMPATIBLE GARMENT RECOMMENDATION

We formulate the task of compatible garment recommendation as the task of retrieving a dressing modality by suggesting suitable complementary fashion items to be paired with a given one.

More formally, let $o = (t, b)$ be an outfit composed of a top item t and a bottom item b. Each garment is labeled with a color and shape label, referred to as c and s, respectively. Given a top $\overline{t}$, we want to retrieve a set of bottoms $\{b_k\}_{k=1:K}$ that are compatible with $\overline{t}$. To consider a recommendation correct, at least one of the proposed bottoms b_k has to share the same color and shape labels with the ground-truth bottom $\overline{b}$; i.e., at least one $k \in \{1, \ldots, K\}$ must exist for which $c_k = \overline{c}$ and $s_k = \overline{s}$.

Our recommendation model is based on two external memory modules in which disentangled color/shape features are stored. Each module acts as an associative memory, relating top features with bottom features and describing different combination modalities for either color or shape. The retrieved modalities are then combined to recreate a final recommendation that can be re-ranked according to general preferences. In the following we present the building blocks of our architecture.

3.1 Disentangled Feature Representation

We process garment images with a convolutional encoder followed by a flattening operation. In this way we map garments into a latent representation ϕ. To obtain separate features for shape and color, we use two different **Multi-Layer Perceptron (MLP)** models, MLP_{shape} and MLP_{color}, that yield descriptors ϕ_{shape} and ϕ_{color}, which are intended to capture different traits of the garment. The two representations are then concatenated, blended together with an additional MLP, and finally decoded with a deconvolutional decoder reconstructing the input image. The model, shown

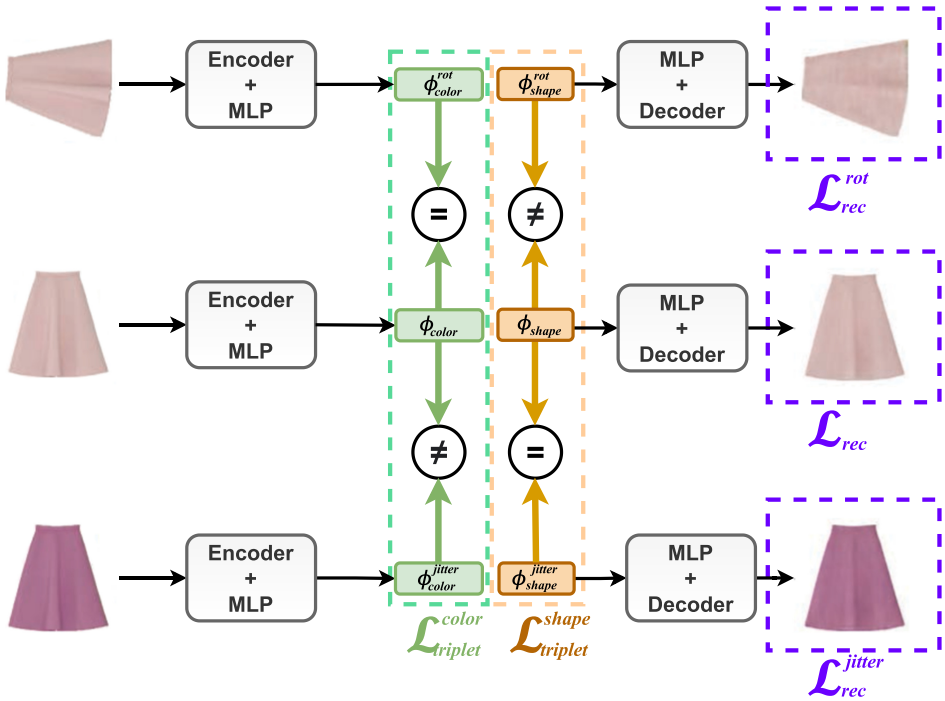

Fig. 3. We exploit rotated and color jittered augmentations to learn disentangled features for shape and color with triple losses. Each branch processes an image with a different augmentation (top: rotated image; center: original image; bottom: color-jittered image).

in Figure 2, acts as an autoencoder with two intermediate latent states, trained by optimizing a reconstruction MSE loss $\mathcal{L}_{rec}$ over pixels.

To disentangle the hidden representations of such states and capture either shape or color, we adopt a contrastive learning approach using a siamese network with three branches (Figure 3). At training time, we feed to the model three images in parallel. The main branch processes the original unaltered image, while the other two branches receive as input color-jittered and rotated versions of the same image. Thanks to these augmentations, the three images share attributes pair-wise: the main branch and the color branch receive images of garments with the same shape, while the main branch and the shape branch observe images with the same color. The rotated and color-jittered images instead do not share any color/shape attribute. In order to disentangle the latent states of the autoencoder, we optimize two triplet margin losses [1] across the three branches.

The rationale is to make features of shared attributes close in the latent space, while pushing away the feature of the altered image. For instance, we want the shape features to be similar for the original and color-jittered images while being dissimilar to the ones of the rotated image.

Let the triplet margin loss be defined as

$$\mathcal{L}_{triplet}(\phi, \phi^+, \phi^-) = \max\{d(\phi, \phi^+) - d(\phi, \phi^-) + M, 0\}, \tag{1}$$

where $d(\cdot, \cdot)$ is a distance function, ϕ is a reference anchor feature, ϕ^+ a positive feature that we want to be close to ϕ, and, vice versa, ϕ^- is a negative feature that we want to separate from the others by a margin M. In our experiments we use the cosine distance and $M = 0.5$.

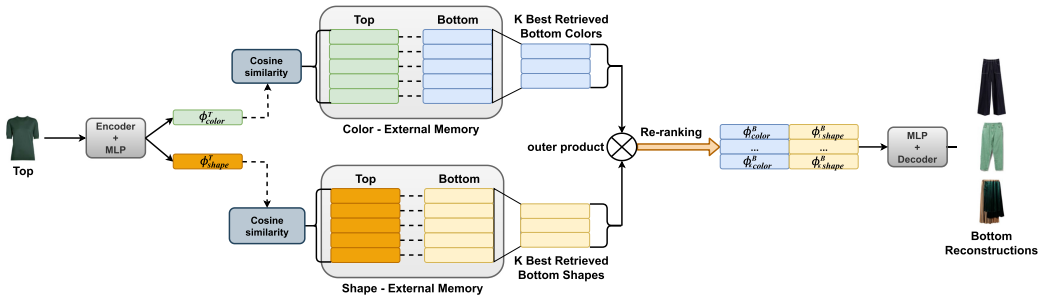

Fig. 4. Architecture overview. The top garment is encoded into disentangled color and shape features, which are used as access keys for the two memory modules. The best K bottom features are read from the memories and are combined via outer product. After a re-ranking based on frequency co-occurrence of top-bottom attributes, we are able to decode the best K bottoms.

To ensure ϕ_{color} and ϕ_{shape} respectively capture color and shape characteristics, we optimize the following losses:

$$\mathcal{L}^{color}_{triplet} = \mathcal{L}_{triplet}\left(\phi_{color}, \phi^{rot}_{color}, \phi^{jitter}_{color}\right), \tag{2}$$

$$\mathcal{L}^{shape}_{triplet} = \mathcal{L}_{triplet}\left(\phi_{shape}, \phi^{jitter}_{shape}, \phi^{rot}_{shape}\right). \tag{3}$$

Here, the *rot* and *jitter* superscripts indicate that the feature is extracted from the rotated image or the color-jittered image, respectively. Overall, to train the siamese autoencoder, we jointly optimize the reconstruction losses for the three branches, which share all the parameters, and the two triplet margin losses for shape and color:

$$\mathcal{L} = \mathcal{L}_{rec} + \mathcal{L}^{rot}_{rec} + \mathcal{L}^{jitter}_{rec} + \lambda\left(\mathcal{L}^{color}_{triplet} + \mathcal{L}^{shape}_{triplet}\right), \tag{4}$$

where the triplet losses are weighed by a coefficient λ, which we set to 0.01.

We train the autoencoder using both top and bottom images, thus obtaining generic encoder/decoder functions that can be used for any garment image. For the sake of simplicity, in the following we refer to ϕ^T for features extracted from top garments and ϕ^B for bottom garments.

3.2 Model

To perform recommendations, we adopt a MANN, with two external memories, M_{color} and M_{shape}, as depicted in Figure 4. The idea is derived from [7], where top and bottom garments are paired in a permanent memory, according to user-defined outfits. Here, instead, we learn to store pairs of top-bottom features concerning either shape or color and we perform a late fusion in the decoding phase. The advantage of our approach is that we can identify non-redundant modalities to pair colors and shapes separately, thus avoiding combinatorial growth in memory size and obtaining more diverse recommendations.

Both memories contain pairs of top-bottom features belonging to known outfits. The memories reflect the feature disentanglement provided by the encoders of the autoencoder. That is, in M_{color} we only store pairs of color features $(\phi^T_{color}, \phi^B_{color})$ and in M_{shape} pairs of shape features $(\phi^T_{shape}, \phi^B_{shape})$.

At inference time, a top image is fed as input to our recommendation system and encoded into two latent vectors ϕ^T_{shape} and ϕ^T_{color}, using the shape and color encoders. The two features are

compared via cosine similarity against the respective memories, acting as read keys to find the most relevant locations.

We retrieve the best K elements from both memories, retaining only the bottom items. We then combine such bottom features from both memories together with an outer product, creating all K^2 possible combinations of shapes and colors. These combined features represent different ways of matching shapes and colors, which can be decoded into actual garment images. However, to perform recommendations we need to suggest existing garments, so we simply retrieve the training sample with the highest cosine similarity according to the concatenation of shape and color features.

Among the K^2 generated pairs, we establish a re-ranking based on the frequency of co-occurrence of color and shape labels (c, s) between top and bottom pairs that belong to a same outfit in the training set. The idea behind this re-ranking strategy is that the MANN is useful to extract good modalities from a pure content-based point of view. By re-ranking using co-occurrence frequencies, we are also taking into account how common these outfits are according to fashion trends.

Memory Controller Training. In order to fill up the two memories, we train two separate memory controllers, C_{shape} and C_{color}. The training process for both controllers is identical, so we will refer to a generic memory M and generic features (ϕ^T, ϕ^B) without any attribute subscript.

Given a top attribute representation ϕ^T, the memory outputs K bottom features ϕ^B_k for $k = 1...K$. Since the memory should be able to propose an attribute (either shape or color, depending on the memory), we compare the features of all proposed items against the corresponding feature of the ground-truth bottom $\bar{\phi}^B$ using a cosine distance:

$$d_k = 1 - \frac{\phi^B_k \cdot \bar{\phi}^B}{\|\phi^B_k\| \cdot \|\bar{\phi}^B\|}, k = 1, \ldots, K. \tag{5}$$

As in [7, 21], we take the minimum error and we feed it to the memory controller, which is trained to write samples in a non-redundant way, storing only relevant information necessary to obtain a satisfactory recommendation. The advantage of considering only the best recommendation is that the network is not penalized for recommending a variety of different outputs, while instead it is enforced to recommend at least an item similar to the ground truth.

A memory controller is a simple linear layer with sigmoidal activation that emits a writing probability p_w, taking as input the minimum distance $d^* = \min\{d_k\}$. A sample gets written in memory when such probability exceeds a given threshold th_w. Previous works in the literature [7, 21] have trained similar memory controllers to maximize the writing probability when the error is high, i.e., when the sample should be added in memory to obtain a better prediction, and minimizing the writing probability when the output is already satisfactory, thus avoiding redundancy. Such behavior is obtained minimizing the following controller loss:

$$\mathcal{L}_{controller} = d^* \cdot (1 - p_w) + (1 - d^*) \cdot p_w. \tag{6}$$

The controller loss in this form, however, suffers from two issues: (1) dependence on the distribution of d^*, which reflects on the number of samples getting written in memory, and (2) collapsing to trivial solutions where p_w is always 0 or 1. Both issues arise when d^* does not follow a normal or uniform distribution, i.e., when there is a strong unbalance toward either low or high distances.

To avoid these issues, we extend this loss in the following way. First, we scale the cosine distances in $[0, 1]$ and we apply a normalization dividing each d^* by an estimate of the maximum distance d^*_{max}, accumulated during training. This has the effect of stretching all errors to cover

the whole $[0, 1]$ interval, making the second term in $\mathcal{L}_{controller}$ tend to zero when d^* is sufficiently high. Second, we add a penalty term to avoid collapsing to trivial solutions. To do so, we accumulate the Nth percentile of d^*, which we denote with $perc_N$, averaging across batches. We assume that samples with errors higher than such value should be written in memory, and therefore we penalize the model when their writing probability is lower than th_w. Vice versa, we still add the penalty when a sample is written but the corresponding d^* is lower than the Nth percentile. The penalties can be formalized as margin losses with margin m:

$$\mathcal{L}_{penalty} = \begin{cases} th_w - p_w + m \quad if \quad p_w < th_w \quad \& \quad d^* > perc_N \\ \\ p_w - th_w + m \quad if \quad p_w > th_w \quad \& \quad d^* < perc_N, \end{cases} \tag{7}$$

where we make the threshold th_w adaptive by setting it equal to the estimate of the Nth percentile of d^*, normalized by d^*_{max}:

$$th_w = \frac{perc_N}{d^*_{max}}. \tag{8}$$

The final controller loss that we adopt is therefore

$$\mathcal{L}^*_{controller} = \frac{d^*}{d^*_{max}} \cdot (1 - p_w) + \left(1 - \frac{d^*}{d^*_{max}}\right) \cdot p_w + \alpha \cdot \mathcal{L}_{penalty}. \tag{9}$$

In our experiments we use $m = 0.3$, $\alpha = 10$ and set the distance threshold to the 99.5 percentile of the distance distribution.

3.3 Training Details

We train our model in two separate steps. First, we train the autoencoder to learn disentangled features for shape and color, and then we train the memory controllers to store nonredundant samples. Separating training into two phases is necessary since we do not want the representations of stored samples to change during training. During the training phase or the memory controllers, to avoid storing incorrect samples at the first iterations, we reset the memory after each epoch by emptying it and re-initializing it with K random samples, i.e., the number of samples that we want to suggest. When the controller is fully trained, we fill the memory from scratch by iterating over the training samples for an additional epoch. We observed that, once convergence is achieved, different initializations do not lead to substantial differences in the final results.

We train our model on two different datasets, IQON3000 [29] and FashionVC [28], as outlined in Section 5. Our final memory modules, trained on the IQON3000 dataset, are filled with 9,282 pairs for color and 2,157 pairs for shape, whereas the memories trained on FashionVC are filled up with 399 pairs for color and 262 pairs for shape. The different number of pairs in the memories filled with the two datasets is given by the different sizes of the datasets: IQON3000 has 308,747 outfits; on the contrary, FashionVC has just 20,726 outfits.

As for the components of our model, the autoencoder is composed as follows. The encoder has 4 convolutional layers with kernel size 3×3, padding 1, and number of channels equal to 8, 16, 32, and 64. Each layer has a ReLU activation and is followed by a max-pooling operation. The resulting feature is a $9 \times 9 \times 64$ feature map, which is flattened into a $5,184$-dimensional vector and fed to the two MLP encoders, both with a hidden dimension of 1,024 and an output of 256. Again, all outputs are followed by ReLU activations. The MLP decoder and the convolutional decoder follow an inverse structure, replacing convolutions with transposed convolutions with stride 2.

For training both models we use the Adam optimizer with a learning rate of 0.005.

4 OUTFIT COMPATIBILITY ESTIMATION

To provide a more comprehensive evaluation with reference to the state of the art, we show that our model can be easily adapted to address a task of outfit compatibility estimation. Even if closely related, this task is slightly different from the task of garment recommendation, since instead of proposing a compatible bottom for a given top, we need to establish the compatibility of a given outfit. Since our model is capable of providing a ranked list of bottoms after accessing memory via top similarity, we exploit such similarity to obtain a compatibility score.

In detail, for either the color or shape modality, we want to obtain a compatibility score c for an outfit $o = (t, b)$ composed of a top t and a bottom b. We first access memory via top similarity using t:

$$s_i^T = \frac{\phi^T \cdot \phi_i^T}{\|\phi^T\| \cdot \|\phi_i^T\|}, \; i = 1, \ldots, |M|, \tag{10}$$

where ϕ^T is the feature corresponding to the top garment and ϕ_i^T is the $i - th$ memory key. This gives us a way of ranking each memory entry, according to top similarities s_i^T. We then compare b against each bottom in memory to find sufficiently similar items. We consider only bottoms with a bottom similarity s^B higher than a chosen threshold th^B by building a positive bottom set $\mathcal{P}_{\mathcal{B}} = \{i \in 1, \ldots, |M| \quad if \quad s_i^B > th^B\}$.

Finally, in order to obtain the compatibility score c, we simply take the top similarity s^T of the highest-ranked memory entry with bottom belonging to $\mathcal{P}_{\mathcal{B}}$. We perform this operation for both color and shape memories and simply add the two scores together to get a final compatibility.

5 EXPERIMENTS

In this section, we report experiments to demonstrate the compatible garment recommendation capabilities of our system. In addition, to provide a more comprehensive comparison with the state of the art, we also adapt our model to perform outfit compatibility estimation and complementary item retrieval. In the following we provide a brief overview of these experimental settings and the metrics we used, explaining how our model can be modified to address such tasks, along with a description of the datasets we use to carry out experiments. Finally, a detailed quantitative and qualitative analysis is reported and a series of ablation studies underline the contribution of several modules in our recommendation network.

5.1 Datasets

The evaluation of the proposed method is extensively performed on the real-world datasets IQON3000 [29] and FashionVC [28].

IQON3000. This dataset is composed of garment images and metadata. Garments are grouped by outfit and are associated to different users. These data were collected from IQON, a Japanese fashion community website, in which members could mix fashion items in order to create new outfits. Once created, an outfit could be shared with other users, and they could express their preferences and create new ones in turn. Each outfit is paired with a csv file that contains the metadata of the fashion apparel: a "setID," which is the unique identifier of the outfit; "setUrl," the URL of the outfit; "likeCount," the number of likes that the outfit received; "user," a unique identifier that identifies the user; and "items," the list of all the garments that compose the outfit, and for each of them, the authors provide the URL of the image, price, category, and color; an item id; an item name; and a description. The dataset includes 308,747 outfits created by 3,568 users with 672,335 fashion items. Each fashion item was also labeled with a category and a color (among 16 categories and 12 colors, respectively). The authors also provide train, validation, and test splits specifying

user id, top id, bottom id, and negative bottom id, which corresponds to a different bottom chosen randomly. We adopted these settings in our experiments.

FashionVC. In our experiments we also involved the FashionVC [28] dataset. The dataset is composed of images and contextual metadata. Specifically, the images are grouped by tops and bottoms, whereas the metadata are composed of id, title, and category. The id is a unique identifier that identifies a garment; the title is a brief description of the garment, for instance, "*Balenciaga Stretch-leather skinny pants*"; the category expresses in which category the garment belongs, for instance, "*Women's Fashion>Clothing>Pants>Balenciaga pants.*" The authors also provide top and bottom pairs that compose an outfit. These data were collected from Polyvore. Polyvore is a website that allows users to mix and match different garments in order to create new outfits. These outfits can be shared with other members, who can express their preferences about them. The dataset is composed of 20,726 outfits, which include 14,871 tops and 13,663 bottoms. Differently from IQON3000, FashionVC does not include any metadata regarding color. Regarding train, validation, and test splits, we adopted the same split proposed by the authors in our experiments: 80% for training, 10% for validation, and 10% for testing.

On Dataset Biases. Since IQON3000 [29] was crawled from the Japanese website IQON, data will be biased toward the Japanese culture and outfits will reflect Japanese taste. FashionVC [28], instead, is based on data extracted from Polyvore, which is an American fashion portal and has a wider user base. Different dataset biases, however, will likely reflect a bias in the model. The memory controller in fact will store pairs of outfits based on their occurrence in the dataset. Nonetheless, two important remarks have to be considered. First, the memory controller is trained to store a sample, i.e., an outfit, when the previously stored ones are not sufficient to perform well. This makes the model able to deal with outliers, intended as dressing modalities that do not follow the bias in the training data. Our model thus is able to model the overall preferences of the users, both the frequent ones and the less common ones. Second, in a real-world context, making the suggestions adhere to specific tastes (or biases) would correspond to be coherent with the collection of a given store and the tastes of its customers.

5.2 Tasks and Metrics

In the following sections, we will perform the evaluation of our method using multiple metrics depending on the task. Here we briefly introduce those metrics, referring to the relative task.

Compatible Garment Recommendation. For the task of compatible garment recommendation, we follow the experimental settings of [7]. We compute *Accuracy@K* for category and color classification while varying the number K of recommended garments. We compute the fraction of recommendations that have at least one sample among the first K suggested items with the same category and/or color of the ground truth.

As in [7], we also use ***mean Average Precision (mAP)*** to establish the ranking quality of the recommendations suggested by our model. We consider as correct each bottom for which the category and/or color matches the one in the ground truth, varying the number of recommended items K. For both metrics, category and color labels are derived from the IQON3000 annotations.

Outfit Compatibility Estimation. As detailed in Section 4, our model can be adapted to perform outfit compatibility estimation. To evaluate this task, we use the ***Area Under the Curve (AUC)*** metric. Here we follow [28, 29, 34] and compute the fraction of times that an outfit, considered as *positive* by a user, is preferred over a random negative one. In other words, we keep track of how often the system prefers items that are appreciated by a user over the ones that he/she does not like.

Table 1. Accuracy Results for Category and Color Classification on the IQON3000 Dataset

	Num Items	5	10	20	30	40	50	60
Category × Color	Ours	**46.76**	**67.00**	**80.57**	**86.15**	**89.08**	**91.12**	**92.61**
	GR-MANN [7]	30.00	45.00	59.00	67.00	71.00	75.00	78.00
Category	Ours	78.99	87.01	91.84	94.10	95.32	**96.11**	96.67
	GR-MANN [7]	**81.00**	**89.00**	**93.00**	**95.00**	**96.00**	96.00	**97.00**
Color	Ours	**58.70**	**76.80**	**87.62**	**91.52**	93.42	94.78	95.76
	GR-MANN [7]	58.00	73.00	85.00	91.00	**94.00**	**96.00**	**97.00**

Table 2. Accuracy Results for Category and Color Classification on the IQON3000 Dataset Using Bottom Garments as Queries and Proposing Tops

Num Items	5	10	20	30	40	50	60
Category × Color	36.96	59.91	75.11	81.31	86.68	89.86	90.58
Category	59.95	75.73	83.70	88.15	93.06	95.93	96.35
Color	60.97	78.58	89.73	92.36	93.24	93.70	94.03

Complementary Fashion Item Retrieval. We extend the evaluation for outfit compatibility estimation by also evaluating our model for complementary fashion item retrieval, as in [28, 29]. We compute the ***Mean Reciprocal Rank (MRR)*** metric. For each top, we randomly select K bottoms as ranking candidates, among which only one is labeled as correct in the ground truth. Since we need to rank K given outfits instead of proposing a ranked list of bottoms as in the mAP evaluation, we assign a compatibility score to each outfit similarly to the AUC evaluation. We use such scores to sort the candidates. To compute the MRR, we average for each sample the inverse of the ranked position (i.e., the reciprocal rank), where the correct ranked position is compared against the ground truth.

5.3 Experimental Results

In the following we present the results for the tasks of garment recommendation, compatibility estimation, and complementary fashion item retrieval. Our focus is on the capability of our model to recommend bottom garments to complement the given top, but we show that our model performs well also for compatibility tasks, obtaining state-of-the-art results.

Compatible Garment Recommendation Results. We present a quantitative analysis of our method for the task of compatible garment recommendation in terms of accuracy for category and/or color. We use the IQON3000 dataset for garment recommendation, as in [7], since it provides both labels for category and color. Our model, which disentangles color and shape features, well adapts to this kind of evaluation, since our two modalities loosely correspond to the provided labels. In fact, when modeling shapes, we are learning (without direct supervision) to represent information that is closely tied to the category annotations in the dataset.

To first evaluate the method, we compute accuracy over both color and category domains separately. Then we perform a cross-domain evaluation as described in [7]. Results are shown in Table 1. Here, we can see that our method performs particularly well when using the combination of the two disentangled features, showing that our model has learned to extrapolate meaningful representations from the two domains. Note that this sub-task requires to correctly predict both category and shape at the same time, making it considerably harder than predicting one single modality. Nonetheless, our method outperforms GR-MANN [7] by a considerable margin. Using

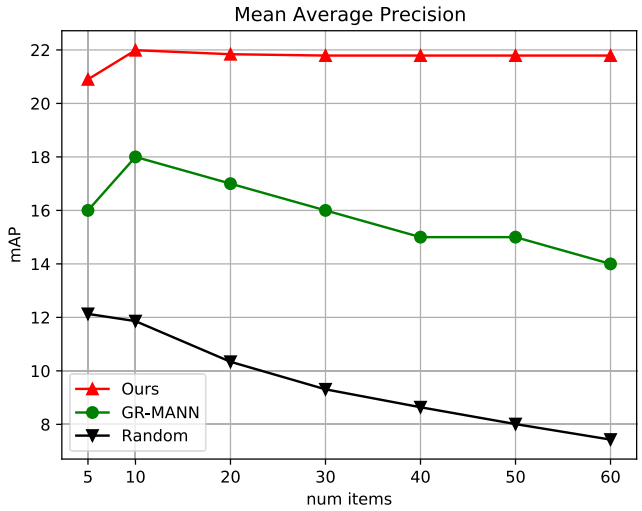

Fig. 5. Mean Average Precision for garment retrieval using both category and color on the IQON3000 dataset.

only one modality instead, it can be seen that our method performs comparably with the baseline. Also note that the original authors do not report decimal values for their results. These results show that our method was able to learn to correctly disentangle the two pieces of information while preserving single domain discriminatory power.

As a sanity check, we retrained our model to perform the same task but inverted the queries in order to suggest tops given a bottom. We retrain the memory controller inverting the data and we then populate a new memory mapping bottoms to tops. For training and testing we adopt the usual IQON split. Results are shown in Table 2. It can be observed that the accuracy, varying the number of suggested items, follows a similar trend to the one in Table 1. Interestingly, though, category accuracy is lower than color accuracy consistently for the whole experiment, indicating that modeling top categories is harder than bottom categories.

We also perform an analysis in terms of ranking of the proposed bottoms. Following [7], we report the mAP of the ranking proposed by our model on IQON3000, varying the number of retrieved garments. That is, given a top, we want to retrieve the first K bottoms for which both category and color should be correctly predicted. Results are shown in Figure 5. It can be seen that our proposed method significantly performs better than [7] for each number of retrieved items, gaining from 4 to 8 points. Moreover, our method is also more robust when predicting a high number of proposals as the mAP does not decrease as fast as for GR-MANN [7]. The explanation for this behavior can be found in the variety of the bottoms proposed by the model. In fact, since the memory is trained to store non-redundant outfits, there will be a limited amount of bottoms that share the same category and color among the proposed ones. As a consequence, this will limit the number of garments to be retrieved, thus reducing the decreasing effect of the mAP that happens when relevant garments are proposed with a bad ranking. In addition, this confirms that on average, the model is capable of proposing the correct bottom with just a small amount of recommendations.

We also added in Figure 5 a Random baseline, obtained by randomly shuffling the K^2 proposals of our architecture and taking the first K. This helps to better define a lower bound for the task and thus to allow a better comparison.

Outfit Compatibility Estimation Results. We apply the strategy outlined in Section 4 to assign compatibility scores to outfits. This allows us to assess the capabilities of our model also for the task of outfit compatibility estimation. We compare compatibility scores for outfits in both

Table 3. Performance Comparison among Different Approaches in Terms of Area under the Curve (AUC) for IQON3000 (left) and FashionVC (Right)

	Method	AUC
Baselines	POP-T [29]	60.42
	POP-U [29]	59.51
	RAND [29]	50.14
	Bi-LSTM [11]	66.11
	BPR-DAE [28]	69.12
	BPR-MF [25]	78.67
	VBPR [12]	80.88
	TBPR [29]	81.02
	VTBPR [29]	81.94
	GP-BPR [29]	83.21
	PAI-BPR-V [26]	84.13
	PAI-BPR-T [26]	84.32
	PAI-BPR [26]	85.02
Proposed	Shape	80.77
	Color	81.61
	Combined	**88.08**

	Method	AUC
Baselines	POP [28]	42.06
	RAND [28]	50.94
	RAW [28]	54.94
	IBR [23]	60.75
	ExIBR [23]	70.33
	BPR-DAE [28]	76.16
Proposed	Shape	81.37
	Color	79.48
	Combined	**88.13**

the IQON3000 and FashionVC datasets. In Table 3 (left) we compare the AUC obtained by our model against several competing methods from the state of the art. For our model we propose three different approaches, i.e., using scores derived from only color or shape or by combining them together, summing the two compatibility scores. Interestingly, using a single modality does not suffice to obtain a higher AUC than several baselines. On the other hand, when combining the two scores together, we observe a 10% improvement, yielding state-of-the-art results. The same kind of behavior can be seen for FashionVC in Table 3 (right). Here, even using color or shape alone, our method is able to obtain a higher AUC compared to the best competing method.

Complementary Item Retrieval.

Here we study how our model performs for complementary item retrieval.

Following [29] and [28], we compute the MRR of the positive bottom among K candidates. Since a candidate can be any bottom in the training set and may be missing from memory, we compute the ranking of the candidates using the compatibility score described in Section 5.3. In Figure 6 we report the MRR for different K values for both the IQON3000 and FashionVC datasets. As in the outfit compatibility evaluation, we propose the three variants with only color, with only shape, or combining both. In both datasets, our combined and color-based methods outperform the state of the art. The shape-based model instead exhibits different behaviors for the two datasets. In IQON3000 the MRR is much lower than the other variants and even lower than some BPR baselines. On FashionVC, instead, the shape-based model is able to slightly outperform the combined version of our model with a large number of items as candidates. We attribute this difference to the higher complexity of the IQON dataset, which has a much larger variability in shapes compared to FashionVC.

5.4 Ablation Studies

We carry out a series of ablation studies to demonstrate the effectiveness of the architectural choices. We trained three variants of our model: *Ours-NoPenalty,* a memory network without the

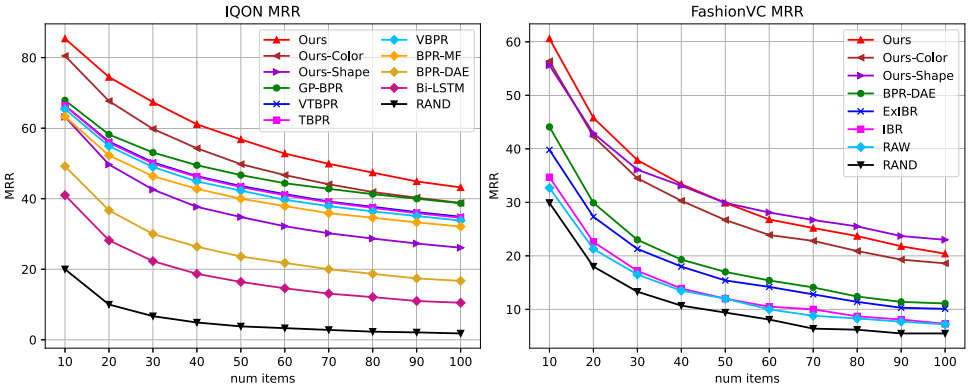

Fig. 6. Mean Reciprocal Rank (MRR) for garment retrieval for the IQON3000 (left) and FashionVC (right) datasets.

penalty term $\mathcal{L}_{penalty}$ in the controller loss of Equation (9); *Ours-ConcatFeat,* a memory network with a single memory obtained by concatenating color and shape features; and *Ours-SingleFeat,* a model with no shape and color distinction. For the first two variants, *Ours-NoPenalty* and *Ours-ConcatFeat*, we used the same autoencoder as in our standard model, while for *Ours-SingleFeat* the whole model is re-trained from scratch. All the ablations are performed on IQON3000 and evaluated for the task of compatible garment recommendation varying the number of suggested items, as in Table 1.

First, we observed that *Ours-NoPenalty* did not manage to converge to meaningful solutions. The controller without the penalty term is not capable of handling samples due to distances d^* not being normally distributed, as discussed in Section 3.2. The training phase always yields a controller that stores in memory either all samples or none. This confirms the lack of flexibility of the controller loss as introduced in previous works such as [7, 20] and the need for a more complex formulation.

In Table 4 we report the results for the two other ablations. We compare them to our standard approach, which we dub here as *Ours-Full*. Both variants achieve lower results compared to the standard approach, with the notable exclusion of *Ours-ConcatFeat* for a large number of suggested items, where the difference with *Ours-Full* is minimal. The memory in the *Ours-ConcatFeat* model is trained by letting a single controller decide whether to store pairs of shape and color features in memory. The two features, however, are kept separate and are concatenated when performing memory access. Thanks to this ablation, we show that having two separate memories adds expressiveness to the model, allowing it to retrieve meaningful samples from memory for both modalities.

A lower drop in accuracy is reported when shape and color disentanglement is completely removed from the model. In fact, when training *Ours-SingleFeat*, we kept only a single MLP encoder and halved the input size of the MLP decoder. Since we now have only a single feature, we trained the autoencoder by removing entirely the triplet losses and retaining only the reconstruction loss $\mathcal{L}_{rec}$. The memory network instead is trained as usual, but using a single memory and a single controller to store samples, similarly to *Ours-ConcatFeat*. From Table 4 it can be seen how this deeply affects the accuracy of the model, confirming the benefits of learning disentangled features. The accuracy drop for this model is due to a lack of diversity in the recommendations and confirms that using disentangled features to perform separate color and shape recommendations yields a more diverse set of proposed bottoms.

Table 4. Ablation Study

	Num Items	5	10	20	30	40	50	60
Category × Color	Ours-Full	46.76	67.00	80.57	86.15	89.08	91.12	92.61
	Ours-ConcatFeat	42.37	62.49	78.19	85.13	88.66	91.21	92.78
	Ours-SingleFeat	39.52	54.59	70.27	78.35	82.68	85.12	86.56
Category	Ours-Full	78.99	87.01	91.84	94.10	95.32	96.11	96.67
	Ours-ConcatFeat	76.70	86.50	91.53	93.80	94.99	95.91	96.51
	Ours-SingleFeat	73.58	80.73	87.06	86.94	88.15	88.88	89.34
Color	Ours-Full	58.70	76.80	87.62	91.52	93.42	94.78	95.76
	Ours-ConcatFeat	54.03	71.46	85.04	90.54	93.13	94.94	96.00
	Ours-SingleFeat	48.92	61.74	75.78	82.84	86.27	88.11	89.13

We report the accuracy of our compatible garment recommendation system varying the architecture. *Ours-Full* denotes the standard architecture. *Ours-ConcatFeat* refers to the architecture with a single memory populated with the concatenation of color and shape features. *Ours-SingleFeat* is an architecture with a single memory and no color-shape separation.

(a) (b)

Fig. 7. T-SNE embeddings for (a) category and (b) color features of the bottoms in the test set.

5.5 Qualitative Results

Here we present a qualitative study of the features generated by our system in terms of both category and color. As our research aims to give bottom recommendations for a given top, we focused the qualitative analysis on these two questions: (1) how are the features of all the bottoms in the test set of the IQON3000 dataset distributed w.r.t. their category and color? and (2) given a bottom, what are the categories and the colors of the closest bottoms in each feature space? The first question allows us to study the feature space distributions and thus to understand how the two feature spaces are organized and assess the quality of the two learned embeddings. The second question, on the other hand, serves the purpose of understanding how descriptive our features are and how well they may perform for tasks such as image retrieval where similarity plays an important role.

Feature Distribution. To understand the feature distribution w.r.t. category and color of the test bottoms, we performed a T-SNE analysis on the features [19]. Figure 7 shows the T-SNE of category (Figure 7(a)) and color (Figure 7(b)), respectively. In Figure 7(a) we can see how bottoms of similar categories are put close to each other in the embedding space. Jeans and trousers are put close to each other, whereas skirts and shorts are more distant. Focusing on skirts, we can

Fig. 8. Reconstruction of shape and color feature combinations applied to jittered and rotated images of tops and bottoms.

see how skirts of similar shapes are indeed closer to each other w.r.t. other skirts with different shapes (e.g., long skirts and short ones). Similarly, Figure 7(b) shows us how bottoms are placed in the color feature space. It is clearly visible that bottoms of similar colors are placed in the same region of the space, with region colors going from dark tones to brighter ones. Note that, despite the dataset only containing 12 colors, the system is trained on the actual colors of the bottoms and thus the underlying space is able to reflect the diversity of a greater number of colors. As a result, we were able to confirm that the system is able to learn feature embeddings that correctly model both category and color characteristics of the garments.

To further understand the disentanglement degree, we also performed a qualitative analysis on the features generated by each modality encoder on the variation of shape and color while training the network. As described in Section 3.1, our system performs several augmentations to learn disentangled representations. In Figure 8, we show the reconstruction of top and bottom features, where both rotation and color jitter is applied. Features are extracted from the respective autoencoders, merged, and decoded by our decoder to produce a visual result of the reconstructed garment. The figure serves two purposes: to show how our augmentations work and to evaluate how disentangled the two modalities are. In the right part of the figure we can see how the reconstructions are able to preserve both shape and color, while also preserving jitter and rotation. The reconstruction is essentially a new garment that preserves the properties that have been captured by the features, confirming once again that the two embeddings are correctly disentangling the two modalities.

Feature Similarity. For tasks such as image retrieval, leveraging an embedding that ensures a good similarity among features of similar items usually reflects in good retrieval performances. This is due to the fact that, to perform retrieval given an input image, results are given by looking for images for which features are close to the input one for a certain distance metric, such as the cosine distance. For this reason, in Figure 9 we present the qualitative results of image retrieval given a test bottom. Note that we are not using the memory in this experiment, just bottom features. Figure 9(a) shows the retrieval results of 10 bottoms in the category feature space ordered by

(a) (b)

Fig. 9. Retrieval results using (a) category and (b) color for 10 bottoms (first column).

Fig. 10. Retrieval of bottom garments given a top using IQON3000. *From left to right*: The first column represents the query top; the second one is the ground-truth bottom; the third shows the bottoms retrieved by our network. Results are kept in the order given by our memory network. In this example the model has to retrieve 10 bottoms.

similarity, while Figure 9(b) shows the result for the same number of bottoms in the color feature space with the same ordering. Starting from the category, we can see how the input image in the first column produces bottoms that have a similar shape. Skirts for the first and last examples produce other skirts that are almost identical in the first results except for the color, which is a desired property that tells us that the two modalities have been well disentangled by the system. Another good example is given by the second and third bottoms, where, ignoring the color, similar results preserve the peculiar properties of the shape given by the fabric.

Similarly, in Figure 9(b) we can see how bottoms of similar colors are retrieved. With some exceptions where the brightness of the image is dominant over the color, such as for the last bottom, usually all retrieved items possess a color similar to the queried one. Similarly to the previous case, the color embedding does not consider the shape, as we can see in the fourth example, where trousers, jeans, skirts, and shorts are all correctly considered similar because of their color, confirming once again that the two modalities have been correctly disentangled to a good extent.

We also performed a qualitative analysis of garment retrieval using the two datasets IQON3000 and FashionVC. In this analysis we used tops, from the test set, as inputs. Figure 10 shows the

Fig. 11. Retrieval of bottom garments given a top using FashionVC. *From left to right*: The first column represents the query top; the second one is the ground-truth bottom; the third shows the bottoms retrieved by our network. Results are kept in the order given by our memory network. In this example the model has to retrieve 10 bottoms.

results given by our system using the IQON3000 dataset. The aim of our memory network is to suggest bottoms that preserve the correct shape and color of the ground truth, but in addition we want variation, in terms of shapes and colors, in the proposed results. We can see that the system both proposes garments that preserve the characteristics of the ground-truth bottom and bottoms that differ for shape/color from the ground truth. Looking at the first row of Figure 10, the top input is paired with a gray and black skirt. We can see that our system mostly proposes skirts of similar shape and suitable color variations. The same happens for the jeans in the second row and for the trousers in the third row. In these cases, the recommendations are close to the ground truth according to color but still offer different styles to complement the top. In the last row we can observe how the system suggests a variety of bottoms, e.g., proposing both skirts and trousers while mostly retaining similar color tonalities to the ground truth. The same evaluation was performed on the FashionVC dataset as shown in Figure 11. The system preserves the same behavior of the results shown above; i.e., the retrieved bottoms preserve the correct shape and color of the ground truth but introduce variation in the proposed results.

Quantitative results can also be discussed in light of the qualitative properties of the features. In fact, experiments have shown a good improvement when both color and shape features are used together. As Figures 7 and 9 suggest, both embeddings exhibit good descriptive power in their respective domain. Thus, it is not unexpected that, when used together, they are able to produce better proposals that are correct in both category and color.

6 CONCLUSIONS

In this article we have presented an approach based on the combination of color/shape feature disentanglement and the usage of external memory modules to store pairing modalities between top and bottom fashion items. We have extended the common controller loss to train such memory modules by addressing issues arising from uneven data distributions, obtaining compact and representative memories. The usage of external memories with disentangled representations has led to significant improvements over the state of the art for compatible garment recommendation.

REFERENCES

[1] Vassileios Balntas, Edgar Riba, Daniel Ponsa, and Krystian Mikolajczyk. 2016. Learning local feature descriptors with triplets and shallow convolutional neural networks. In *Proceedings of the British Machine Vision Conference 2016 (BMVC'16)* Vol. 1, 3.

[2] Federico Becattini, Xuemeng Song, Claudio Baecchi, Shi-Ting Fang, Claudio Ferrari, Liqiang Nie, and Alberto Del Bimbo. 2021. PLM-IPE: A pixel-landmark mutual enhanced framework for implicit preference estimation. In *ACM Multimedia Asia (MMAsia'21)*. Association for Computing Machinery, New York, NY, Article 42, 5 pages. https://doi.org/10.1145/3469877.3490621

[3] Wolmer Bigi, Claudio Baecchi, and Alberto Del Bimbo. 2020. Automatic interest recognition from posture and behaviour. In *Proceedings of the 28th ACM International Conference on Multimedia (ACMMM'20)*. 2472–2480.

[4] Wen-Huang Cheng, Sijie Song, Chieh-Yun Chen, Shintami Chusnul Hidayati, and Jiaying Liu. 2020. Fashion meets computer vision: A survey. *arXiv preprint arXiv:2003.13988* (2020).

[5] Guillem Cucurull, Perouz Taslakian, and David Vazquez. 2019. Context-aware visual compatibility prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR'19)*. 12617–12626.

[6] Lavinia De Divitiis, Federico Becattini, Claudio Baecchi, and Alberto Del Bimbo. 2021. Style-based outfit recommendation. In *2021 International Conference on Content-Based Multimedia Indexing (CBMI'21)*. https://doi.org/10.1109/CBMI50038.2021.9461912

[7] Lavinia De Divitiis, Federico Becattini, Claudio Baecchi, and Alberto Del Bimbo. 2020. Garment recommendation with memory augmented neural networks. In *Pattern Recognition. ICPR International Workshops and Challenges - Virtual Event, Proceedings, Part II (Lecture Notes in Computer Science)*, Vol. 12662. Springer, 282–295. https://doi.org/10.1007/978-3-030-68790-8_23

[8] Zunlei Feng, Zhenyun Yu, Yezhou Yang, Yongcheng Jing, Junxiao Jiang, and Mingli Song. 2018. Interpretable partitioned embedding for customized fashion outfit composition. *arXiv preprint arXiv:1806.04845* (2018).

[9] Guangyu Gao, Liling Liu, Li Wang, and Yihang Zhang. 2019. Fashion clothes matching scheme based on siamese network and autoencoder. In *Proceedings of Multimedia Systems (MMSys'19)*.

[10] Alex Graves, Greg Wayne, and Ivo Danihelka. 2014. Neural Turing machines. *arXiv preprint arXiv:1410.5401* (2014).

[11] Xintong Han, Zuxuan Wu, Yu-Gang Jiang, and Larry S. Davis. 2017. Learning fashion compatibility with bidirectional LSTMs. In *Proceedings of the 25th ACM international conference on Multimedia (ACMMM'17)*. 1078–1086.

[12] Ruining He and Julian McAuley. 2016. VBPR: Visual Bayesian personalized ranking from implicit feedback. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI'16)*, Vol. 30.

[13] Sebastian Heinz, Christian Bracher, and Roland Vollgraf. 2017. An LSTM-based dynamic customer model for fashion recommendation. *arXiv preprint arXiv:1708.07347* (2017).

[14] Yuxin Hou, Eleonora Vig, Michael Donoser, and Loris Bazzani. 2021. Learning attribute-driven disentangled representations for interactive fashion retrieval. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV'21)*. 12147–12157.

[15] Yang Hu, Xi Yi, and Larry S. Davis. 2015. Collaborative fashion recommendation: A functional tensor factorization approach. In *Proceedings of the 23rd ACM International Conference on Multimedia (ACMMM'15)*. 129–138.

[16] Vignesh Jagadeesh, Robinson Piramuthu, Anurag Bhardwaj, Wei Di, and Neel Sundaresan. 2014. Large scale visual recommendations from street fashion images. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (SIGKDD'14)*. 1925–1934.

[17] Jinhuan Liu, Xuemeng Song, Zhumin Chen, and Jun Ma. 2019. Neural fashion experts: I know how to make the complementary clothing matching. *Neurocomputing* 359 (2019), 249–263.

[18] Zhi Lu, Yang Hu, Yunchao Jiang, Yan Chen, and Bing Zeng. 2019. Learning binary code for personalized fashion recommendation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR'19)*. 10562–10570.

[19] Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of Machine Learning Research* 9, (November 2008), 2579–2605.

[20] Francesco Marchetti, Federico Becattini, Lorenzo Seidenari, and Alberto Del Bimbo. 2020. MANTRA: Memory augmented networks for multiple trajectory prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'20)*.

[21] Francesco Marchetti, Federico Becattini, Lorenzo Seidenari, and Alberto Del Bimbo. 2020. Multiple trajectory prediction of moving agents with memory augmented networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2020).

[22] Francesco Marchetti, Federico Becattini, Lorenzo Seidenari, and Alberto Del Bimbo. 2022. SMEMO: Social memory for trajectory forecasting. *arXiv preprint arXiv:2203.12446* (2022).

[23] Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton Van Den Hengel. 2015. Image-based recommendations on styles and substitutes. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'15)*. 43–52.

[24] Federico Pernici, Matteo Bruni, and Alberto Del Bimbo. 2020. Self-supervised on-line cumulative learning from video streams. *Computer Vision and Image Understanding* (2020), 197–198.

[25] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2012. BPR: Bayesian personalized ranking from implicit feedback. *arXiv preprint arXiv:1205.2618* (2012).

[26] Dikshant Sagar, Jatin Garg, Prarthana Kansal, Sejal Bhalla, Rajiv Ratn Shah, and Yi Yu. 2020. PAI-BPR: Personalized outfit recommendation scheme with attribute-wise interpretability. In *2020 IEEE 6th International Conference on Multimedia Big Data (BigMM'20)*. IEEE, 221–230.

[27] Xuemeng Song, Fuli Feng, Xianjing Han, Xin Yang, Wei Liu, and Liqiang Nie. 2018. Neural compatibility modeling with attentive knowledge distillation. In *Proceedings of the 41st International ACM SIGIR Conference on Research & Development in Information Retrieval (SIGIR'18)*. 5–14.

[28] Xuemeng Song, Fuli Feng, Jinhuan Liu, Zekun Li, Liqiang Nie, and Jun Ma. 2017. Neurostylist: Neural compatibility modeling for clothing matching. In *Proceedings of the 25th ACM international conference on Multimedia (ACMMM'17)*. 753–761.

[29] Xuemeng Song, Xianjing Han, Yunkai Li, Jingyuan Chen, Xin-Shun Xu, and Liqiang Nie. 2019. GP-BPR: Personalized compatibility modeling for clothing matching. In *Proceedings of the 27th ACM International Conference on Multimedia (ACM MM'19)*. 320–328.

[30] Sainbayar Sukhbaatar, Jason Weston, Rob Fergus, et al. 2015. End-to-end memory networks. In *Proceedings of Advances in neural information processing systems (NIPS'15)*. 2440–2448.

[31] Mariya I. Vasileva, Bryan A. Plummer, Krishna Dusad, Shreya Rajpal, Ranjitha Kumar, and David Forsyth. 2018. Learning type-aware embeddings for fashion compatibility. In *Proceedings of the European Conference on Computer Vision (ECCV'18)*. 390–405.

[32] Andreas Veit, Balazs Kovacs, Sean Bell, Julian McAuley, Kavita Bala, and Serge Belongie. 2015. Learning visual clothing style with heterogeneous dyadic co-occurrences. In *Proceedings of the IEEE International Conference on Computer Vision (CVPR'15)*. 4642–4650.

[33] Xin Wang, Bo Wu, and Yueqi Zhong. 2019. Outfit compatibility prediction and diagnosis with multi-layered comparison network. In *Proceedings of the 27th ACM International Conference on Multimedia (ACMMM'19)*. 329–337.

[34] Chen Fang Wang-Cheng Kang, Zhaowen Wang, and Julian J. McAuley. 2017. Visually-aware fashion recommendation and design with generative image models. *CoRR* abs/1711.02231 (2017). arXiv:1711.02231 http://arxiv.org/abs/1711.02231.

[35] Jason Weston, Sumit Chopra, and Antoine Bordes. 2014. Memory networks. *arXiv preprint arXiv:1410.3916* (2014).

[36] Wenhui Yu, Huidi Zhang, Xiangnan He, Xu Chen, Li Xiong, and Zheng Qin. 2018. Aesthetic-based clothing recommendation. In *Proceedings of the 2018 World Wide Web Conference (WebConf'18)*. 649–658.

[37] Junjie Zhang, Kaixuan Liu, Min Dong, Hua Yuan, Chun Zhu, and Xianyi Zeng. 2020. An intelligent garment recommendation system based on fuzzy techniques. *Journal of the Textile Institute* 111, 9 (2020), 1324–1330.

Received 30 January 2022; revised 22 March 2022; accepted 10 April 2022