Learning Attribute-driven Disentangled Representations for Interactive Fashion Retrieval – Supplementary Material

Yuxin Hou*
Aalto University
yuxin.hou@aalto.fi

Eleonora Vig
Amazon
eleonov@amazon.com

Michael Donoser
Amazon
donoserm@amazon.de

Loris Bazzani
Amazon
bazzanil@amazon.de

In this supplementary material, we add the following details which are missing from the main paper because of space limitation: 1) architecture, 2) implementation and runtime, and 3) some additional visualizations.

1. Architecture Details

We discuss here the architecture parameters and substructures for the proposed Attribute-Driven Disentangled Encoder (ADDE) and for the Outfit Complementary Retrieval Model which can be found in Table 1.

The choice of the backbone network in ADDE used in the different interactive retrieval tasks was influenced by the previous works that we compare to (Table 1, row 2). For the attribute manipulation task (ADDE-M), we use an AlexNet as also done by AMNet [3], whereas for conditional similarity retrieval (ADDE-C) and outfit complementary item retrieval (ADDE-O) we resort to a ResNet18 as used in CSN [2] and CSA-Net [1].

The attribute-specific learner ϕ_a (Table 1, row 3) in ADDE is a fully-connected two-layer network with a ReLU nonlinearity in-between, and ϕ_a maps the image representation coming from the above backbone into a $d = 340$ dimensional attribute-specific subspace. The training of these subspaces is supervised via an attribute prediction tasks, where the classification layer is made of a fully-connected layer followed by softmax (Table 1, row 4). For example, the size of the output for Shopping100k is 151.

In outfit complementary item retrieval, the output representation of ADDE (before softmax) is passed through a fully-connected layer ψ_a of dimension 340 (Table 1, row 6) to obtain attribute-specific representations specialized for outfits. To encode the input and target categories, we first map them as one-hot encoding; since there are 11 categories, the concatenated category representation is of size 22: input size of the model κ of row 7 in Table 1. We have in total 5 attributes out of 12 since we selected the ones that overlap with Shopping100k, therefore this is the output of the model κ in Table 1.

*Work done during an internship with Amazon

For mathematical definitions of the above parameters please see Sec. 3.1 and 3.4 in the main paper.

2. Implementation Details

ADDE Accuracy. We report here the accuracy of the attribute encoder for the attribute prediction task, which is 79.3% on Shopping100k and is on par with AMNet [3] which is 78.3%.

Attribute manipulation retrieval. The memory block $\mathcal{M}$ has size $A \cdot d \times J$, where A is the number of attribute types in the dataset, J is the total number of attribute values, and d is the dimension of each attribute specific embedding. For all datasets, we fix the dimension for attribute-specific representations to $d = 340$. In Shopping100k there are 12 attribute types and altogether 151 attribute values, hence the size of $\mathcal{M}$ is 4080×151. In DeepFashion experiments, we consider three attribute types and J is 202, hence $\mathcal{M} \in \mathbb{R}^{1020\times202}$.

The final loss function used to train our attribute manipulation network is defined as the weighted sum of the individual losses, which are defined in Sec. 3.2 of the main paper:

$$L = w_{cls}L_{cls}+w_cL_c+w_{ct}L_{ct}+w_{lt}L_{lt}+w_{mem}L_{mem} \quad (1)$$

We set the weight parameters to $w_{cls} = 0.2, w_c = 1.0, w_{ct} = 1.0, w_{lt} = 1.0, w_{mem} = 0.4$. For the two triplet losses L_{ct} and L_{lt}, we set the margin m to 0.5 at the beginning of the training and, every 10 epochs, we increase the margin by 0.1. We use a batch size of 128 and train the model for 30 epochs. The learning rate for the first 20 epochs is set to $1e^{-4}$, and for the remainig 10 epochs we decay the learning rate to $5e^{-5}$.

The Z constant in the formula of the NDCG metric (Eq. 8 of the main paper) is introduced to ensure that when the returned k results are all correct, the NDCG score is zero. Hence, Z is the ideal DCG score defined as: $\sum_{j=1}^{k} \frac{1}{log(j+1)}$.

Runtime of ADDE and ADDE-M. We test the runtime on an NVIDIA TESLA V100 GPU. Disentangled feature

Attribute-Driven Disentangled Encoder (ADDE)	
backbone	AlexNet (Conv1 to Conv5 + Linear 1) or ResNet18
attribute-specific learners ϕ_a	Linear (4096, 340) + ReLU + Linear (340, 340)
attribute predictor $softmax$	Linear (340, number of attribute values)
Outfit Complementary Retrieval Model	
attribute-specific outfit learners ψ_a	Linear (340, 340)
category-specific attention learner κ	Linear (22, 256) + ReLU + Linear (256, 5) + Softmax

Table 1: Model parameters for the Attribute-Driven Disentangled Encoder (ADDE) and for the Outfit Complementary Retrieval Model.

extraction for indexing is performed at 1.9 seconds per 1000 images. Attribute manipulation retrieval runs as 0.001 seconds per test query.

Conditional similarity retrieval. We follow the same training setup as CSN [2]: we set the margin to 0.2 for the triplet loss, use a learning rate of 5e-5, and train the network with a mini-batch size of 256.

For the triplet prediction task using Shopping100k, we generate the triplets in the following way. For each image in the training set and test sets (separately), we randomly pick a positive image and a negative image for each attribute type. In Shopping100k, there are 12 attributes, hence we will sample 12 triplets per image.

To perform conditional similarity retrieval, we first extract disentangled representations for all images in the database for indexing. Then, given the query image and the conditional attribute c, we extract the disentangled representation for the query image and return K-Nearest-Neighbors by computing the Euclidean distance between the attribute-specific embedding of the query image and the attribute-specific embeddings of gallery images for the attribute c.

Outfit complementary item retrieval: We use the same training parameters as in CSA-Net [1]: we set the margin to 0.3 for the ranking loss and the initial learning rate to 5e-5. During training, the learning rate linearly decreases to zero. We set the warm-up ratio to zero and train the network with a mini-batch size of 96.

3. Additional Qualitative Results

We provide additional qualitative results below. Figure 1 shows attribute manipulation retrieval results: succesful retieval on the left and some failure cases on the right. It is worth noticing that it is considered a failure when the results do not match all desired attributes. In fact, the change of attributes are performed correctly (*e.g.*, row 1: change color from gray to brown; row 2: change style from checked to floral), however some other aspects of the results are also changed.

Figures 2 and 3 show qualitative examples for conditional similarity retrieval. Finally, outfit complementary item retrieval results are given in Figures 4 and 5, both for successful retrieval and some failure cases. It is worth noticing that the exact match is not found in the top-10 items for the failure cases, however the recommended images still seems to show a high degree of compatibility with the query outfit in terms of color, style, and other attributes specific to a certain category (e.g., heel type, fit, ...).

References

[1] Yen-Liang Lin, Son Tran, and Larry S Davis. Fashion outfit complementary item retrieval. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3311–3319, 2020. 1, 2

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

[3] Bo Zhao, Jiashi Feng, Xiao Wu, and Shuicheng Yan. Memory-augmented attribute manipulation networks for interactive fashion search. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1520–1528, 2017. 1

Figure 1: Top-5 retrieval results for attribute manipulation retrieval. The green boxes denote images that match all desired attributes. On the right we show failure cases where retrieval results do not match all desired attributes of the queries.

Figure 2: Qualitative results for conditional similarity retrieval.

Figure 3: Qualitative results for conditional similarity retrieval.

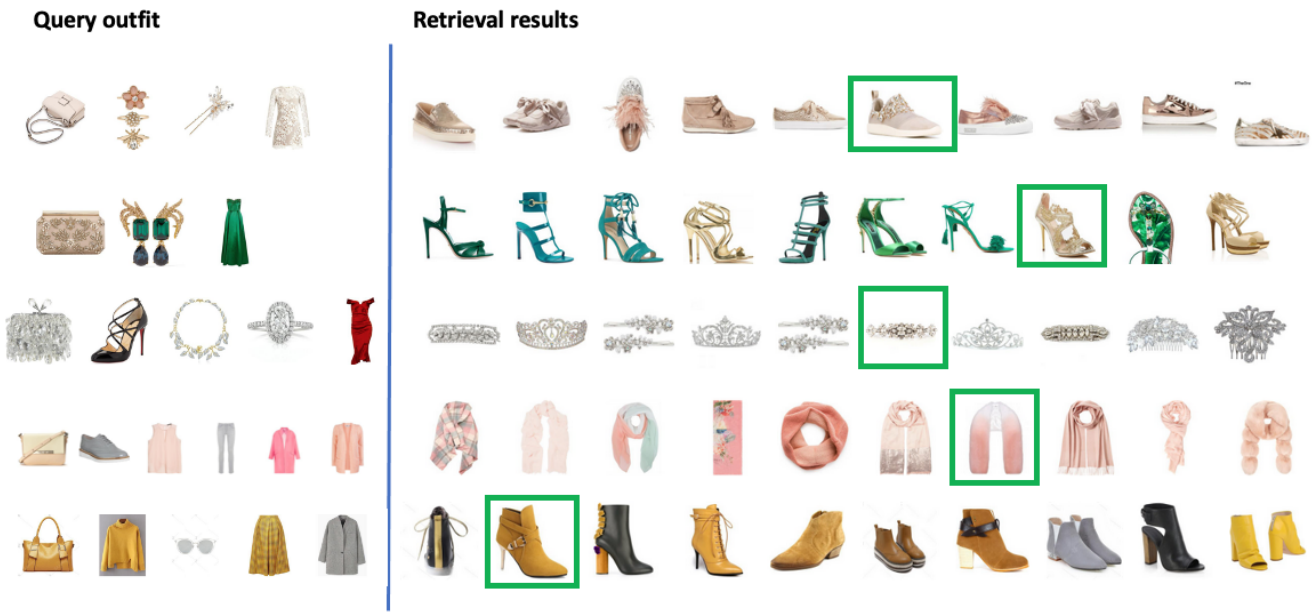

Figure 4: Top-10 retrieval results for outfit complementary retrieval. The green boxes denote the target complementary items.

Figure 5: Top-10 retrieval results for outfit complementary retrieval (failure cases). Retrieved results do not contain the target complementary items.