# UIGR: Unified Interactive Garment Retrieval

Xiao Han[1,2] Sen He[1,2] Li Zhang[3] Yi-Zhe Song[1,2] Tao Xiang[1,2]
[1]CVSSP, University of Surrey
[2]iFlyTek-Surrey Joint Research Centre on Artificial Intelligence
[3]School of Data Science, Fudan University
{xiao.han, sen.he, y.song, t.xiang}@surrey.ac.uk, lizhangfd@fudan.edu.cn

## Abstract

*Interactive garment retrieval (IGR) aims to retrieve a target garment image based on a reference garment image along with user feedback on what to change on the reference garment. Two IGR tasks have been studied extensively: text-guided garment retrieval (TGR) and visually compatible garment retrieval (VCR). The user feedback for the former indicates what semantic attributes to change with the garment category preserved, while the category is the only thing to be changed explicitly for the latter, with an implicit requirement on style preservation. Despite the similarity between these two tasks and the practical need for an efficient system tackling both, they have never been unified and modeled jointly. In this paper, we propose a* ***U****nified* ***I****nteractive* ***G****arment* ***R****etrieval (UIGR) framework to unify TGR and VCR. To this end, we first contribute a large-scale benchmark suited for both problems. We further propose a strong baseline architecture to integrate TGR and VCR in one model. Extensive experiments suggest that unifying two tasks in one framework is not only more efficient by requiring a single model only, it also leads to better performance. Code and datasets are available at GitHub.*

## 1. Introduction

In computer vision, there is a long line of research on understanding garment image content [26, 41, 37, 15, 2, 8]. Among them, **I**nteractive **G**arment **R**etrieval (IGR) [46, 39, 41] is most relevant to the garment search problem. IGR aims to retrieve a target garment image based on a reference garment image along with user feedback on what to change on the reference garment. It enables a shopper to find exactly what she/he wants because it allows for the fine-tuning of search results through user feedback.

Two IGR tasks, namely **T**ext-**G**uided garment **R**etrieval (TGR) [39, 27] and **V**isually **C**ompatible garment **R**etrieval (VCR) [25, 17] have been studied so far (see Figure 1). TGR (dialog 1 & 3 of Figure 1) retrieves garments in the same category as the reference garment. The feedback is in the form of either synthetic sentence [12, 39] or natural language [41], indicating the intended attribute changes from the reference to the target garment. In contrast, the feedback for VCR (dialog 2 of Figure 1) typically only indicates category change explicitly, in the form of an indicator rather than text [25, 17]. Nevertheless, as the retrieval is constrained to only visually compatible items, implicit feedback is to preserve the style so that the reference and target look lovely when worn together.

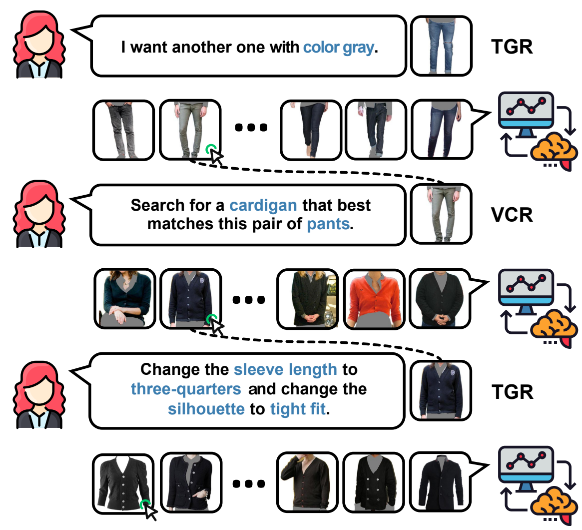


Figure 1. An example IGR scenario where both TGR and VCR can take place. Given a reference garment, users may search for a garment of the same category with some attribute changes (TGR), or visually compatible garments of different categories (VCR).

Despite being two instantiations of IGR, TGR and VCR have never been studied together in a unified framework. Indeed, they are evaluated on completely different sets of benchmarks. The developed methods also seem pretty different. TGR is usually done by first compositing the reference garment with the interaction signal together and then retrieving the garments similar to the composited query

[39, 7, 24, 35]. Since different garments are compatible along multiple dimensions, such as color, pattern, and material, previous works in VCR typically learn subspace embeddings to capture different notions of similarity and aim to learn a joint embedding space where compatible garments of different categories are close [25, 17].

In this paper, for the first time, we propose **U**nified **I**nteractive **G**arment **R**etrieval (UIGR) to unify the two tasks in a single framework. We argue that there are two benefits for doing so: (1) As shown in Figure 1, it is common to have both tasks incurred in the same shopping session. It is thus more efficient to build one rather than two separate models to tackle both tasks. (2) Due to the similarity in format (*i.e.*, both are IGR tasks), having a single multi-task framework makes it possible for both tasks to benefit from each other when trained jointly end-to-end. However, unifying the two tasks is challenging, with two main obstacles to overcome: the lack of benchmarks and the discrepancy in the two types of user feedback.

To this end, we try to solve these problems with two main contributions: (1) We establish a novel benchmark for the study of this unified problem by re-purposing Fashionpedia [20], where prompt engineering is adopted to generate user feedback from fine-grained attributes. (2) We introduce a multi-task model jointly learning two tasks, which unifies TGR and VCR in a single framework and serves as a strong baseline for UIGR. Experiments demonstrate that unifying the two tasks in a single model is not only possible but also yields better overall performance, compared with modeling them separately using two models.

## 2. Related work

**Text-guided garment retrieval.** TGR is a special type of image retrieval problem with multimodal compositional queries [39, 27, 4]. In general, the user feedback used to guide the searching process can be attributes [46, 13, 1], synthetic sentences [12, 39], and natural language (free text) [41, 43]. Different TGR models proposed so far differ primarily in the design of their compositors. A compositor plays a fundamental role to integrate the textual information with the imagery modality. TGR compositors have been proposed based on various techniques, such as gating mechanism [39], hierarchical attention [7, 19, 10, 16], graph neural network [44, 35], joint learning [6, 23, 35, 42, 45], ensemble learning [40], style-content modification [24, 5] and vision & language pre-training [27].

**Visually compatible garment retrieval.** Predicting fashion compatibility is to determine whether two garments of different categories match well aesthetically. On this basis, the recommendation can be done either as fill-in-the-blank [14] at item level or as personalized outfit recommendation [31, 30] at outfit level. In addition to being a set, an outfit can also be represented as a sequence [14], or a graph [9].

Instead of computing the compatibility in a single space, most approaches [38, 37, 36, 25, 17, 22] explore learning subspace embeddings to capture different notions of compatibility. [38, 37] learn many conditional subspaces, each for a pair of categories. [36] learns several subspaces conditioned on the features from both the reference garment and the target garment. However, this kind of method is not suitable for large-scale retrieval where exhaustive comparison is prohibitive. [25, 17] concatenate one-hot labels of the reference and target category to represent the interaction signal to meet the setting of large-scale retrieval.

**Fashion datasets.** Over the past few years, many fashion datasets have been proposed for multiple applications [8], such as detection [26, 28, 11], retrieval [26, 11, 33], attribute recognition [26, 13], popularity learning [29, 32, 2] and synthesis [15, 21]. The most related datasets to our work are [12, 13, 41] for TGR and [37, 25, 34] for VCR. Besides not being suitable for the unified setting, previous TGR and VCR benchmarks have some other problems, which will be explained in next Section.

## 3. New benchmark for IGR

Next, we describe the data collection process and provide an in-depth analysis of UIGR. The overall data collection procedure is illustrated in Figure 2. The basic statistics is summarized in Table 1 and 2 [1].

### 3.1. Image and attribute collection

We collect UIGR garment images based on the original images, garment bounding boxes, garment segmentation masks, and fine-grained attributes from Fashionpedia [20] with a series of pre-processing [2].

### 3.2. Image pair selection

**TGR subset.** Previous benchmarks [13, 41] select image pairs by comparing the similarity of text information, *e.g.*, image titles or attribute labels. As shown in Figure 3, this selection strategy often leads to weakly related image pairs with drastically different visual appearances. The user feedback thus cannot accurately describe all the changes necessary to align the image pairs because there are too many changes needed. We thus take a different strategy: using image similarity instead of text similarity for pair selection. Specifically, we use a DenseNet [18, 23] pre-trained on DeepFashion [26] to get image feature vectors. Next, for each image, we calculate the cosine similarity between it and all images of the same category in the image pool and only consider the top three most similar matches.

[1] "Triplet" in this article refers to one piece of data, *i.e.*, two images and one sentence, rather than anchor, positive and negative sample pair.

[2] More details about pre-processing steps are listed in Supp. Mat.

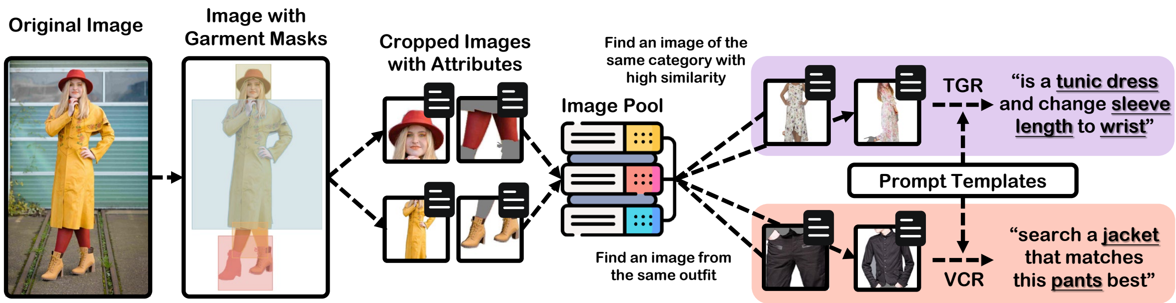


Figure 2. Overview of the dataset collection process. The whole pipeline is based on the image and corresponding high-quality annotations from Fashionpedia [20]. (1) We firstly construct an image pool by cropping each garment using its ground truth mask. (2) To construct TGR triplets, we select a pair of images with the same category and high similarity. Then the user feedback is generated by filling relative attributes in the blank of prompt templates. (3) For VCR triplets construction, the image pair is selected according to whether both images are from the same outfit. We generate this kind of user feedback by mentioning the categories of both reference and target images.


Figure 3. Typical bad triplets in FashionIQ [41].

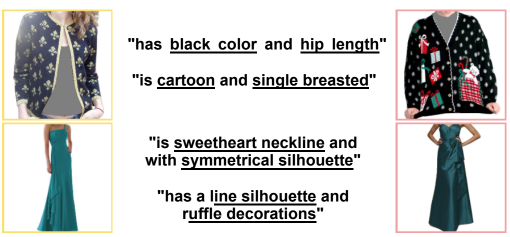


Figure 4. Triplet examples in UIGR TGR subset.

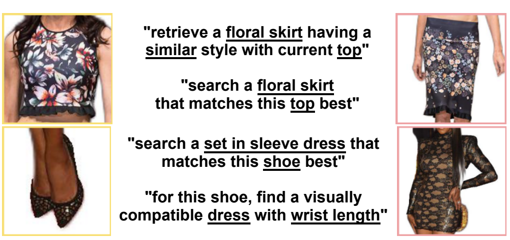


Figure 5. Triplet examples in UIGR VCR subset.

| Split | # Images | # Outfits | # Triplets | |
|---|---|---|---|---|
| | | | TGR | VCR |
| Train | 76,685 | 29,321 | 210,189 | 190,150 |
| Validation | 25,181 | 9,688 | 68,847 | 61,776 |
| Test | 25,434 | 9,814 | 69,639 | 62,418 |

Table 1. Dataset statistics of UIGR.

| Dataset | # Triplets | # Categories | Caption length |
|---|---|---|---|
| Shoes [3, 12] | 10k | 1 | 5.22 words |
| Fashion200K [13, 39] | 172k | 5 | 4.00 words |
| FashionIQ [41] | 18k | 3 | 5.36 words |
| Our TGR | 381k | 27 | 6.33 words |
| **Dataset** | **# Outfits** | **# Categories** | **Interaction signal** |
| Polyvore retrieval [25] | 17k | 16 | One-hot labels |
| Our VCR | 49k | 27 | Text |

Table 2. Comparisons with other related datasets.

**VCR subset.** We select all garments coming from the same outfit in a bidirectional way to construct image pairs, which is a standard procedure adopted in previous VCR benchmarks [37, 25].

### 3.3. User feedback generation

Because the scale of UIGR is more than twenty times that of FashionIQ, manually annotating each image pair with fine-grained user feedback is laborious and costly. To this end, we adopt prompt engineering to automatically generate the user feedback based on the relative attributes between two garments. Following the setting of FashionIQ, we generate two sentences for each image pair.

**TGR subset.** We manually summarize tens of cloze prompt templates from FashionIQ captions. These templates include several single phrases, such as *"has* $\{V\}$ $\{A\}$*"* and *"change* $\{A\}$ *to* $\{V\}$*"*, where {V} and {A} hold the blank for one attribute name and its value. The templates of multiple phrases are based on the combination of single phrases. Finally, the relative attributes between two images are filled in the blanks of the randomly selected prompt template.

**VCR subset.** To unify the VCR task with TGR, they need to have the same user feedback format, *i.e.*, sentences describing the intended changes to the reference garment. One obvious choice is to use the prompt engineering technique to generate sentences describing only the category changes for VCR. However, this fails to capture the implicit user feedback when it comes to VCR. That is, the style of the target garment needs to be consistent with that of the reference.

To this end, we first calculate the correlation matrix of all attributes between any two kinds of garments. When constructing VCR triplets, we will predict the most likely

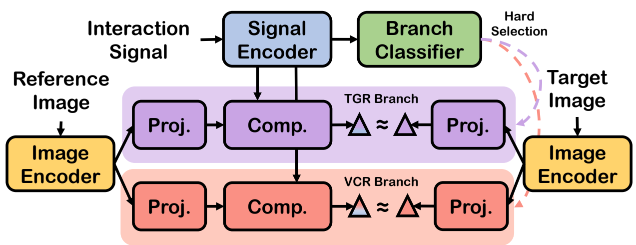


Figure 6. Proposed multi-task architecture for UIGR.

target attributes based on the existing attributes of the reference image. Next, we will randomly mention one attribute in the predicted attributes using the attribute correlation matrix when generating user feedback.

Different from the TGR subset, we manually design several prompt templates for VCR, such as *"search a {TV} {TC} that matches this {RC} best"* and *"for this {RC}, find a visually compatible {TV} {TC}"*, where {TV}, {TC} and {RC} stand for the target attribute value, target category and reference category, respectively.

### 3.4. Dataset analysis

The examples of our collected TGR triplets are depicted in Figure 4. Compared with those from FashionIQ in Figure 3, our triplets seems more reasonable. In particular, although all relative captions in FashionIQ are annotated via a crowdsourcing platform, many captions are too ambiguous to describe the exact search direction. Since we select image pairs based on the image similarity to avoid significant visual changes, the subsequently generated user feedback is more accurate and fine-grained.

Figure 5 shows the examples of VCR subset Compared with one-hot labels for user feedback, sentences are more flexible and scalable to integrate more fine-grained information from users. Further, the VCR task now has the same setting as the TGR, making unification possible.

## 4. Experiments

Although there are different implementations for the compositors of VCR and TGR, they share the same goal: preserving unmentioned visual appearance aspects of the reference and changing only those mentioned in the interaction signal/feedback. Our multi-task model unifies the two tasks based on the same goal. However, to accommodate the major difference in the change directions of the two tasks, namely whether the category is preserved or changed, we use different compositors. As shown in Figure 6, two branches are used for separately learning two composition processes with shared image and signal encoders. Considering that the features needed to be modified for the two branches are not the same, we use two projection modules to project image features to two latent spaces ahead of the composition process. We also jointly learn a classifier to

| Comp. \ Metrics | | TGR Results | | | VCR Results | | | Mean | |
|---|---|---|---|---|---|---|---|---|---|
| | | R@10 | R@50 | mAP | R@10 | R@50 | mAP | R@K | mAP |
| CSA [25] | I | 38.98 | 72.08 | 14.29 | 71.03 | 86.83 | 46.82 | **67.23** | **30.56** |
| | U | 36.90 | 70.57 | 13.37 | 70.46 | 86.88 | 46.47 | 66.20 | 29.92 |
| TIRG [39] | I | 46.27 | 77.57 | 19.78 | 69.30 | 85.88 | 46.15 | 69.76 | 32.97 |
| | U | 45.06 | 76.75 | 18.91 | 72.11 | 88.42 | 48.54 | **70.59** | **33.73** |
| VAL [7] | I | 43.27 | 75.10 | 18.06 | 62.99 | 81.97 | 40.40 | 65.83 | 29.23 |
| | U | 40.19 | 71.78 | 17.28 | 67.95 | 86.24 | 44.72 | **66.54** | **31.00** |
| CoSMo [24] | I | 40.24 | 72.15 | 17.31 | 64.10 | 83.18 | 41.36 | 64.92 | 29.34 |
| | U | 40.96 | 72.40 | 17.62 | 68.64 | 86.41 | 45.05 | **67.10** | **31.36** |
| RTIC [35] | I | 48.23 | 78.79 | 19.98 | 69.28 | 86.26 | 46.10 | 70.64 | 33.04 |
| | U | 46.75 | 77.80 | 19.26 | 74.18 | 89.56 | 50.78 | **72.07** | **35.02** |

Table 3. The evaluation results for the proposed unified (U) model with five different compositors on UIGR test split. For each compositor, the compared model (I) is the combination of two models independently trained on TGR and VCR.

distinguish different user feedback. With it, our model can automatically determine which branch should be selected to do composition during inference, thus allowing the real-world application scenario depicted in Figure 1 to be supported by one model.

We compare our multi-task model with previous methods where TGR and VCR are studied independently [3]. The main experiment results are reported in Table 3. We can draw the following conclusions from the results: (1) Overall, our proposed multi-task model achieves comparable and even better performance (1.18 mAP increase on average) compared with the combination of two separately trained models. The best result (the last row) is achieved by our multi-task model with RTIC [35] as the compositor. (2) In most cases (4 out of 5), our model achieves significantly better performance than an independently trained model on the VCR task. It suggests that text is more suitable than one-hot labels as the user feedback for VCR. With the user feedback in the same modality of TGR, VCR can learn useful information from TGR in our unified model. (3) Although our model has a slight performance drop on the TGR task, its performance is still competitive against an independently trained model on the TGR subset (*e.g.*, only 0.58 mAP drop for TGR but 2.95 mAP gain for VCR on average).

In summary, the experiment results demonstrate that VCR and TGR can be unified and implemented in a single model through our proposed framework. It is more efficient by having one model only and more effective with improved overall performance over the two tasks.

## 5. Conclusion

We have proposed a unified setting for TGR and VCR with a new large-scale benchmark and a baseline multi-task architecture, in which we use text as the unified user feedback format for both TGR and VCR. We conducted experiments to show that the proposed baseline model has competitive or even better performance than previous methods, and it is also more efficient to use one model instead of two.

[3]We put implementation details, hyperparameter settings, evaluation protocols, ablation study and qualitative results in Supp. Mat.

## References


[1] Kenan E Ak, Ashraf A Kassim, Joo Hwee Lim, and Jo Yew Tham. Learning attribute representations with localization for flexible fashion search. In *CVPR*, 2018. 2

[2] Ziad Al-Halah, Rainer Stiefelhagen, and Kristen Grauman. Fashion forward: Forecasting visual style in fashion. In *ICCV*, 2017. 1, 2

[3] Tamara L Berg, Alexander C Berg, and Jonathan Shih. Automatic attribute discovery and characterization from noisy web data. In *ECCV*, 2010. 3

[4] Soravit Changpinyo, Jordi Pont-Tuset, Vittorio Ferrari, and Radu Soricut. Telling the what while pointing to the where: Multimodal queries for image retrieval. In *ICCV*, 2021. 2

[5] Pranit Chawla, Surgan Jandial, Pinkesh Badjatiya, Ayush Chopra, Mausoom Sarkar, and Balaji Krishnamurthy. Leveraging style and content features for text conditioned image retrieval. In *CVPR workshops*, 2021. 2

[6] Yanbei Chen and Loris Bazzani. Learning joint visual semantic matching embeddings for language-guided retrieval. In *ECCV*, 2020. 2

[7] Yanbei Chen, Shaogang Gong, and Loris Bazzani. Image search with text feedback by visiolinguistic attention learning. In *CVPR*, 2020. 2, 4

[8] Wen-Huang Cheng, Sijie Song, Chieh-Yun Chen, Shintami Chusnul Hidayati, and Jiaying Liu. Fashion meets computer vision: A survey. *ACM CSUR*, 2021. 1, 2

[9] Guillem Cucurull, Perouz Taslakian, and David Vazquez. Context-aware visual compatibility prediction. In *CVPR*, 2019. 2

[10] Eric Dodds, Jack Culpepper, Simao Herdade, Yang Zhang, and Kofi Boakye. Modality-agnostic attention fusion for visual search with text feedback. *arXiv preprint arXiv:2007.00145*, 2020. 2

[11] Yuying Ge, Ruimao Zhang, Xiaogang Wang, Xiaoou Tang, and Ping Luo. Deepfashion2: A versatile benchmark for detection, pose estimation, segmentation and re-identification of clothing images. In *CVPR*, 2019. 2

[12] Xiaoxiao Guo, Hui Wu, Yu Cheng, Steven Rennie, Gerald Tesauro, and Rogério Schmidt Feris. Dialog-based interactive image retrieval. In *NeurIPS*, 2018. 1, 2, 3

[13] Xintong Han, Zuxuan Wu, Phoenix X Huang, Xiao Zhang, Menglong Zhu, Yuan Li, Yang Zhao, and Larry S Davis. Automatic spatially-aware fashion concept discovery. In *ICCV*, 2017. 2, 3

[14] Xintong Han, Zuxuan Wu, Yu-Gang Jiang, and Larry S Davis. Learning fashion compatibility with bidirectional lstms. In *ACM MM*, 2017. 2

[15] Xintong Han, Zuxuan Wu, Zhe Wu, Ruichi Yu, and Larry S Davis. Viton: An image-based virtual try-on network. In *CVPR*, 2018. 1, 2

[16] Mehrdad Hosseinzadeh and Yang Wang. Composed query image retrieval using locally bounded features. In *CVPR*, 2020. 2

[17] Yuxin Hou, Eleonora Vig, Michael Donoser, and Loris Bazzani. Learning attribute-driven disentangled representations for interactive fashion retrieval. In *ICCV*, 2021. 1, 2

[18] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *CVPR*, 2017. 2

[19] Surgan Jandial, Ayush Chopra, Pinkesh Badjatiya, Pranit Chawla, Mausoom Sarkar, and Balaji Krishnamurthy. Trace: Transform aggregate and compose visiolinguistic representations for image search with text feedback. *arXiv preprint arXiv:2009.01485*, 2020. 2

[20] Menglin Jia, Mengyun Shi, Mikhail Sirotenko, Yin Cui, Claire Cardie, Bharath Hariharan, Hartwig Adam, and Serge Belongie. Fashionpedia: Ontology, segmentation, and an attribute localization dataset. In *ECCV*, 2020. 2, 3

[21] Wentao Jiang, Si Liu, Chen Gao, Jie Cao, Ran He, Jiashi Feng, and Shuicheng Yan. Psgan: Pose and expression robust spatial-aware gan for customizable makeup transfer. In *CVPR*, 2020. 2

[22] Donghyun Kim, Kuniaki Saito, Samarth Mishra, Stan Sclaroff, Kate Saenko, and Bryan A Plummer. Self-supervised visual attribute learning for fashion compatibility. In *ICCV workshops*, 2021. 2

[23] Jongseok Kim, Youngjae Yu, Hoeseong Kim, and Gunhee Kim. Dual compositional learning in interactive image retrieval. In *AAAI*, 2021. 2

[24] Seungmin Lee, Dongwan Kim, and Bohyung Han. Cosmo: Content-style modulation for image retrieval with text feedback. In *CVPR*, 2021. 2, 4

[25] Yen-Liang Lin, Son Tran, and Larry S Davis. Fashion outfit complementary item retrieval. In *CVPR*, 2020. 1, 2, 3, 4

[26] Ziwei Liu, Ping Luo, Shi Qiu, Xiaogang Wang, and Xiaoou Tang. Deepfashion: Powering robust clothes recognition and retrieval with rich annotations. In *CVPR*, 2016. 1, 2

[27] Zheyuan Liu, Cristian Rodriguez-Opazo, Damien Teney, and Stephen Gould. Image retrieval on real-life images with pretrained vision-and-language models. In *ICCV*, 2021. 1, 2

[28] Ziwei Liu, Sijie Yan, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Fashion landmark detection in the wild. In *ECCV*, 2016. 2

[29] Ling Lo, Chia-Lin Liu, Rong-An Lin, Bo Wu, Hong-Han Shuai, and Wen-Huang Cheng. Dressing for attention: Outfit based fashion popularity prediction. In *ICIP*, 2019. 2

[30] Zhi Lu, Yang Hu, Yan Chen, and Bing Zeng. Personalized outfit recommendation with learnable anchors. In *CVPR*, 2021. 2

[31] Zhi Lu, Yang Hu, Yunchao Jiang, Yan Chen, and Bing Zeng. Learning binary code for personalized fashion recommendation. In *CVPR*, 2019. 2

[32] Yunshan Ma, Xun Yang, Lizi Liao, Yixin Cao, and Tat-Seng Chua. Who, where, and what to wear? extracting fashion knowledge from social media. In *ACM MM*, 2019. 2

[33] Zhe Ma, Jianfeng Dong, Zhongzi Long, Yao Zhang, Yuan He, Hui Xue, and Shouling Ji. Fine-grained fashion similarity learning by attribute-specific embedding network. In *AAAI*, 2020. 2

[34] Ambareesh Revanur, Vijay Kumar, and Deepthi Sharma. Semi-supervised visual representation learning for fashion compatibility. In *ACM RecSys*, 2021. 2

[35] Minchul Shin, Yoonjae Cho, Byungsoo Ko, and Geonmo Gu. Rtic: Residual learning for text and image composition using graph convolutional network. *arXiv preprint arXiv:2104.03015*, 2021. 2, 4

[36] Reuben Tan, Mariya I Vasileva, Kate Saenko, and Bryan A. Plummer. Learning similarity conditions without explicit supervision. In *ICCV*, 2019. 2

[37] Mariya I Vasileva, Bryan A Plummer, Krishna Dusad, Shreya Rajpal, Ranjitha Kumar, and David Forsyth. Learning type-aware embeddings for fashion compatibility. In *ECCV*, 2018. 1, 2, 3

[38] Andreas Veit, Serge Belongie, and Theofanis Karaletsos. Conditional similarity networks. In *CVPR*, 2017. 2

[39] Nam Vo, Lu Jiang, Chen Sun, Kevin Murphy, Li-Jia Li, Li Fei-Fei, and James Hays. Composing Text and Image for Image Retrieval - an Empirical Odyssey. In *CVPR*, 2019. 1, 2, 3, 4

[40] Haokun Wen, Xuemeng Song, Xin Yang, Yibing Zhan, and Liqiang Nie. Comprehensive linguistic-visual composition network for image retrieval. In *SIGIR*, 2021. 2

[41] Hui Wu, Yupeng Gao, Xiaoxiao Guo, Ziad Al-Halah, Steven Rennie, Kristen Grauman, and Rogerio Feris. Fashion iq: A new dataset towards retrieving images by natural language feedback. In *CVPR*, 2021. 1, 2, 3

[42] Yuchen Yang, Min Wang, Wengang Zhou, and Houqiang Li. Cross-modal joint prediction and alignment for composed query image retrieval. In *ACM MM*, 2021. 2

[43] Yifei Yuan and Wai Lam. Conversational fashion image retrieval via multiturn natural language feedback. In *SIGIR*, 2021. 2

[44] Feifei Zhang, Mingliang Xu, Qirong Mao, and Changsheng Xu. Joint attribute manipulation and modality alignment learning for composing text and image to image retrieval. In *ACM MM*, 2020. 2

[45] Gangjian Zhang, Shikui Wei, Huaxin Pang, and Yao Zhao. Heterogeneous feature fusion and cross-modal alignment for composed image retrieval. In *ACM MM*, 2021. 2

[46] Bo Zhao, Jiashi Feng, Xiao Wu, and Shuicheng Yan. Memory-augmented attribute manipulation networks for interactive fashion search. In *CVPR*, 2017. 1, 2