UIGR: Unified Interactive Garment Retrieval
(Supplementary Material)

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

A. Additional information on UIGR dataset

Our dataset is built upon Fashionpedia [8], which is a large-scale dataset for garment segmentation and fine-grained attribute localization. Fashionpedia provides an ontology built by fashion experts containing 27 garment categories and 19 garment parts. It provides not only fine-grained attributes but also implicit visual compatibility relationships for all garments in an outfit. All alternatives [5, 11] cannot meet all these conditions at the same time.

Image pre-processing. We want an IGR model to focus on the garment to be refined by the user feedback. The background and other garment items in a given image are thus distractions and should be removed. To this end, a series of pre-processing steps are introduced: (1) We use a salient object detection model [15, 21] to remove the background, which is an easy task given the typical clean background in fashion catalog images. (2) When there are multiple garments with the same category in one image (*e.g.*, shoes and gloves), if they do not overlap, we only keep the one with the largest pixel area; (3) We delete the masks of garment parts (*e.g.*, sleeves and pockets) but merge their attributes into the garments they belong to; (4) We delete the garments that have low-resolution or extreme aspect ratio; (5) If there are pixels of other garments in the bounding box, we mask these excess pixels with gray color. Finally, we cropped each garment with its attributes from the original image to construct a substantial image pool.

Prompt engineering. We list all used prompts for user feedback generation in Table 1. Our prompts simulate a variety of syntax structures: single phrases, compositional phrases, and propositional phrases.

More triplet examples. We present more triplet examples of UIGR in Figure 1 and Figure 2. As discussed in the main paper, the TGR triplets we collected successfully follow the assumption that there could not be too many visual changes between the reference garment and the target garment. Our TGR triplets thus are much higher quality than those in FashionIQ [20] with less ambiguity. Besides, our TGR subset contains 27 different garments, far more than FashionIQ, which only has three categories (top tee, shirt and dress).

TGR
search another item with a similar style
there are no changes between two images
change $\{A\}$ to $\{V\}$
has $\{V\}$ $\{A\}$
is $\{V\}$
change $\{A\}$ to $\{V_1\}$ and $\{V_2\}$
change $\{A_1\}$ to $\{V_1\}$ and change $\{A_2\}$ to $\{V_2\}$
has $\{V_1\}$ and $\{V_2\}$ $\{A_2\}$
has $\{V_1\}$ $\{A_1\}$ and $\{V_2\}$ $\{A_2\}$
is $\{V_1\}$ and with $\{V_2\}$ $\{A_2\}$
is $\{V_1\}$ and $\{V_2\}$
VCR
search a $\{TC\}$ that matches this $\{RC\}$ best
retrieve a $\{TC\}$ having a similar style with current $\{RC\}$
for this $\{RC\}$, find a visually compatible $\{TC\}$
replace this $\{RC\}$ with a $\{TC\}$ that has a consistent style
search a $\{TV\}$ $\{TC\}$ that matches this $\{RC\}$ best
retrieve a $\{TV\}$ $\{TC\}$ having a similar style with current $\{RC\}$
for this $\{RC\}$, find a visually compatible $\{TC\}$ with $\{TV\}$ $\{TA\}$
replace this $\{RC\}$ with a $\{TC\}$ that has $\{TV\}$ $\{TA\}$

Table 1. All prompts for user feedback generation of UIGR. $\{V\}$ and $\{A\}$ hold the blank for one attribute name and its value. $\{TV\}$, $\{TC\}$ and $\{RC\}$ stand for the target attribute value, target category and reference category, respectively. Which kind of TGR prompt to choose depends on how many related attributes (0, 1 or 2) need to be mentioned. Which kind of VCR prompt to choose depends on whether the target attributes need to be mentioned.

Thanks to the flexibility of text, our VCR subset includes more meaningful information compared to concatenated one-hot labels [11]. Now each user feedback sentence states category changes and intended attributes based on the statistics of attribute co-occurrence between compatible garment items, which is more in line with reality. With

Figure 1. More triplet examples in UIGR TGR subset.

Figure 2. More triplet examples in UIGR VCR subset.

the help of such kind of VCR triplets, the potential user can specific the search direction through mentioning some specific target attributes. Most importantly, now the VCR has the unified setting with TGR.

B. Additional information on the multi-task baseline model

Given a reference garment image g^r and an interaction signal (user feedback) s, the ultimate goal of interactive retrieval is to search the gallery for another garment image g^t that best matches the modification mentioned in s. Regardless of whether the user wants to modify the attributes or the category of the reference garment, the interaction signal is in the same textual format. TGR and VCR can thus be modeled in the same framework.

We will first briefly introduce how previous works study these two tasks separately and then describe our unified solution based on multi-task learning.

B.1. Preliminary method

In Figure 3, an existing pipeline for interactive retrieval typically consists of three components: image encoder $\mathcal{E}^I$, interaction signal encoder $\mathcal{E}^S$ and compositor $\mathcal{C}$.

Firstly, both reference image and target image are fed into the image encoder to obtain representations in the feature space: $\mathbf{g}^r = \mathcal{E}^I(g^r), \mathbf{g}^t = \mathcal{E}^I(g^t)$, where $\mathcal{E}^I$ is usually instantiated by a CNN pre-trained on ImageNet [3] and a linear projection layer [19, 17].

In the meantime, the interaction signal is processed by

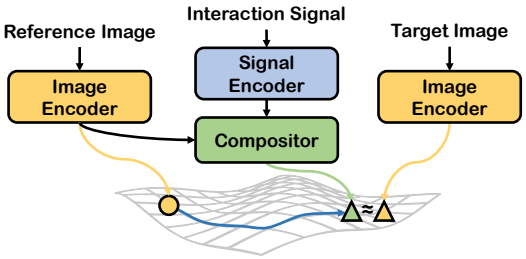


Figure 3. Previous architecture for TGR/VCR.

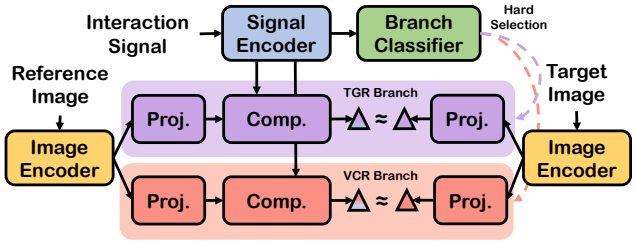


Figure 4. Proposed multi-task architecture for UIGR.

the signal encoder to get the signal representation: $\mathbf{s} = \mathcal{E}^S(s)$, where the interaction signal is represented by the concatenation of reference category c^r and target category c^t for VCR [11] or by user feedback t for TGR [19].

Finally, the most important step is to incorporate the interaction signal's feature into reference image's feature via a compositor: $\mathbf{x} = \mathcal{C}(\mathbf{g}^r, \mathbf{s})$. For VCR, this compositor is always instantiated by a conditional similarity module [11, 7] to learn different sub-spaces with different notions. For TGR, this compositor works globally [19, 17] or locally [1, 10] to modify the feature map of reference image.

The goal of this pipeline is to make the composed query $\mathbf{x}$ as close as possible to the target $\mathbf{g}^t$ in a shared feature space. A widely used objective function is the batch-based classification loss (BBC) [19], which assumes the same form as the InfoNCE loss [13]:

$$\mathcal{L}_{bbc} = \frac{1}{B}\sum_{i=1}^{B} -\log \frac{\exp\left[\kappa\left(\mathbf{x}_i, \mathbf{g}_i^t\right)/\tau\right]}{\sum_{j=1}^{B}\exp\left[\kappa\left(\mathbf{x}_i, \mathbf{g}_j^t\right)/\tau\right]}, \quad (1)$$

where $\kappa(\cdot,\cdot)$ and τ are cosine distance metric and tuneable temperature, respectively. In this loss, each example is contrasted with a set of other negatives. It thus achieves better discriminative learning and faster convergence.

During inference, the features of all gallery images will be calculated in advance by image encoder. For each composed query, its cosine similarity with all gallery features will be obtained. Finally, an identity list is sorted according to the cosine similarity as the retrieval result sequence.

B.2. Proposed multi-task framework

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 4, two branches are used for separately learning two composition processes with shared image and signal encoders.

More specifically, we use a quintuplet $\{g^r, s_v, s_t, g_v^t, g_t^t\}$ containing reference garment, VCR signal, TGR signal, VCR target garment, and TGR target garment as the input for training. These three garment images will be fed into a shared image encoder $\mathcal{E}^I$ to get respective features $\mathbf{g}^r$, $\mathbf{g}_v^t$ and $\mathbf{g}_t^t$. Similarly, two signals will get their features $\mathbf{s}_v$ and $\mathbf{s}_t$ via a shared signal encoder $\mathcal{E}^S$.

Considering that the features needed to be modified for the two branches are not the same, we use two projection modules $\mathcal{P}_t$ and $\mathcal{P}_v$ to project image features to two latent spaces ahead of the composition process. Exactly how the projection module is realized depends on what compositor is employed here. Specifically, for the compositor who directly modifies the feature map [1, 10], we implement the projection module with a lightweight CNN; for the compositor working globally [19, 17], we use a linear projection layer following the global average pooling instead.

After choosing a compositor architecture from a existing method (*e.g.*, [19, 17, 1, 10]), we need two compositors $\mathcal{C}_t$ and $\mathcal{C}_v$ of the same architecture but without shared weights, to separately learn two composition processes for the two tasks. For each branch, the compositor serves for incorporating signal feature into the projected image feature of reference garment:

$$\mathbf{x}_v = \mathcal{C}_v\left(\mathcal{P}_v\left(\mathbf{g}^r\right), \mathbf{s}_v\right), \quad \mathbf{x}_t = \mathcal{C}_t\left(\mathcal{P}_t\left(\mathbf{g}^r\right), \mathbf{s}_t\right). \quad (2)$$

For both branches, two BBC losses $\mathcal{L}_{bbc}^v$ and $\mathcal{L}_{bbc}^t$ will be calculated independently according to Equation 1.

We also jointly learn a classifier to distinguish different user feedback. Specifically, we simply choose the branch with a higher score predicted by the classifier, *i.e.*, hard selection, which is empirically found to be the most effective design. We instantiate this branch classifier with an MLP $\mathcal{M}$ and optimize it via cross-entropy loss (CE):

$$\begin{aligned}\mathcal{L}_{ce} = &\frac{1}{B}\sum_{i=1}^{B} -\log \frac{\exp\left[\mathcal{M}_0\left(\mathbf{s}_{vi}\right)\right]}{\exp\left[\mathcal{M}_0\left(\mathbf{s}_{vi}\right)\right] + \exp\left[\mathcal{M}_1\left(\mathbf{s}_{vi}\right)\right]} \\ &+\frac{1}{B}\sum_{i=1}^{B} -\log \frac{\exp\left[\mathcal{M}_1\left(\mathbf{s}_{ti}\right)\right]}{\exp\left[\mathcal{M}_0\left(\mathbf{s}_{ti}\right)\right] + \exp\left[\mathcal{M}_1\left(\mathbf{s}_{ti}\right)\right]}.\end{aligned} \quad (3)$$

Our model is end-to-end optimized by the overall objective function, which is the direct summation of two BBC losses and one CE loss:

$$\mathcal{L} = \mathcal{L}_{bbc}^v + \mathcal{L}_{bbc}^t + \mathcal{L}_{ce}. \quad (4)$$

Arch.	TGR Results			VCR Results			Mean	
	R@10	R@50	mAP	R@10	R@50	mAP	R@K	mAP
I	46.27	77.57	19.78	69.30	85.88	46.15	69.76	32.97
U+SC	43.97	76.22	17.67	71.18	87.89	46.89	69.82	32.28
U+SP	42.74	75.04	17.84	69.67	87.51	45.60	68.74	31.72
U+SC+SP	43.94	75.76	18.30	68.83	87.26	44.42	68.95	31.36
U	45.10	76.84	18.94	72.15	88.61	48.49	**70.68**	**33.72**

Table 2. Ablation study on the proposed multi-task model. SC: sharing compositor across two branches; SP: sharing projection module across two branches.

User Feedback	Attribute Augmented	R@10	R@50	mAP
One-hot		69.30	85.88	46.15
One-hot	✓	70.98	87.16	47.80
Text		70.77	86.88	47.51
Text	✓	**72.65**	**88.64**	**49.06**

Table 3. Experiment results of attribute argumented models (with one-hot labels or text as the user feedback) on VCR subset.

C. Additional information on experiments

Implementation details. We realize the image encoder and signal encoder by utilizing ResNet50 [6] and Bi-GRU [2]. The ResNet50 is pre-trained on ImageNet [3] and the word embeddings of Bi-GRU are initialized by CLIP text encoder [16, 4]. To demonstrate the universality of our multi-task architecture, we instantiate the compositor with recent representative methods [11, 19, 1, 10, 17]. For the projection module, we adopt two different architectures (convolution layer with 512 output channels or linear layer with 512 output dimensions) according to whether the compositor is used to modify the feature map or the global feature.

Hyper-parameters setting. We use random horizontally flip and random crop as image data augmentation methods. All images are resized to 224×224. The batch size and temperature in the $\mathcal{L}_{bbc}$ are 64 and 0.0625, respectively. Our model is trained with Adam optimizer [9] for 40 epochs with an initial learning rate 2×10^{-4}, which is decayed by a factor 0.1 at the 15^{th} and 25^{th} epoch, respectively. We also linearly increase the learning rate from 2×10^{-5} to 2×10^{-4} at the first 5 epochs. All experiments are conducted on one Tesla V100 GPU (32GB memory) with Pytorch [14].

Evaluation metrics. We adopt the standard evaluation metric for retrieval, *i.e.*, Recall@K, denoted as R@K for short. To circumvent the problem of false negatives [12], we follow FashionIQ [20] to set K as larger values (10 and 50). In addition, we also report the mean Average Precision (mAP) [1] for a comprehensive evaluation.

Evaluation protocols. Since we are integrating VCR into TGR, we want the model has the ability to distinguish different categories. Consequently, we lead a more difficult evaluation protocol than FashionIQ. Unlike FashionIQ, which evaluates three categories separately, category labels are not available for our evaluation protocol. That is, all images in the gallery will calculate a similarity with the composed query.

[1] For each query, mAP is calculated with top 50 results.

D. More quantitative results

D.1. Ablation study

We examine the design of each component in our proposed model. The critical problem we are going to explore is whether compositor and projection module can be shared between TGR and VCR. In all experiments, we remove the branch classifier and use TIRG [19] as the compositor.

As shown in Table 2, sharing both projection module and compositor leads to a performance drop. In addition, a shared projection module alone leads to a more considerable performance drop than a shared compositor. This result demonstrates that projecting the features of reference garments into different latent spaces is vital for this multi-task framework. To unify VCR and TGR in a single model, the projection module and compositor thus cannot be shared because different tasks need different embedding features.

D.2. Attribute augmented VCR model

In addition to helping to unify VCR and TGR, we believe that mentioning target attributes is a more general way for VCR, even for models that use one-hot labels as user feedback. To demonstrate that, we conduct a small experiment by concatenating the one-hot label of the target attribute behind that of the reference category and target category.

As shown in Table 3, we can conclude that one-hot labels also benefit from mentioning target attributes, but text modality can integrate this kind of attribute information into user feedback better.

D.3. Cross-domain evaluation

To demonstrate the universality of our generated user feedback, we conduct cross-domain evaluation. Precisely, we compare the results of the same model with 3 different strategies: (1) trained on UIGR-TGR, tested on FashionIQ (zero-shot); trained and tested on FahsionIQ (fully supervised); (3) trained on UIGR-TGR and FashionIQ, and then tested on FashionIQ (transfer learning). As shown in Table 4, we can draw several conclusions: (1) Even under the zero-shot setting, every method achieves reasonable performance; (2) With the transferred knowledge from UIGR-TGR, every model has a substantial performance gain (2.93 R@K and 1.85 mAP increase on average). In general, although our user feedback is generated, its generalization ability is sufficient to help the model achieve good performance on the manually annotated dataset.

Comp.	Training Dataset	Dress			Shirt			Top Tee			Mean	
		R@10	R@50	mAP	R@10	R@50	mAP	R@10	R@50	mAP	R@K	mAP
TIRG[19]	UIGR	7.59	19.98	3.25	7.90	18.99	3.25	8.77	23.56	3.91	14.47	3.47
	FashionIQ	23.65	49.93	11.89	21.98	**46.61**	9.31	27.84	55.07	12.53	37.51	11.24
	UIGR + FashionIQ	**26.97**	**53.64**	**12.65**	**22.87**	46.07	**10.29**	**29.58**	**57.73**	**13.80**	**39.48**	**12.25**
VAL[1]	UIGR	6.05	18.20	2.78	7.31	17.76	2.85	7.50	20.04	3.05	12.81	2.89
	FashionIQ	19.09	44.57	9.02	16.68	37.93	7.21	20.45	46.76	8.88	30.91	8.37
	UIGR + FashionIQ	**26.43**	**52.66**	**13.02**	**20.36**	**43.52**	**9.54**	**25.85**	**53.14**	**12.21**	**36.99**	**11.59**
CoSMo[10]	UIGR	7.14	18.80	3.23	6.04	17.52	2.73	7.45	20.96	3.22	12.99	3.06
	FashionIQ	20.87	46.80	9.35	**18.30**	40.92	8.00	22.95	50.33	10.36	33.36	9.24
	UIGR + FashionIQ	**23.50**	**49.48**	**10.42**	17.96	**41.76**	**8.22**	**25.14**	**52.58**	**11.68**	**35.07**	**10.11**
RTIC[17]	UIGR	8.13	21.32	3.53	7.85	20.31	3.32	9.43	23.56	4.12	15.10	3.66
	FashionIQ	25.93	51.76	12.00	22.37	46.57	9.91	27.84	56.65	13.10	38.52	11.00
	UIGR + FashionIQ	**28.01**	**53.74**	**13.58**	**24.04**	**47.64**	**11.36**	**31.67**	**57.78**	**14.92**	**40.48**	**13.29**

Table 4. The cross-domain (UIGR-TGR → FashionIQ [20]) evaluation results. All results are reported on the three subsets of FashionIQ.

E. More qualitative results

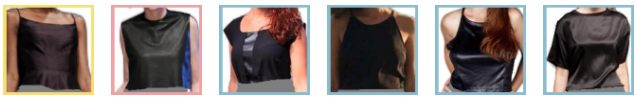
E.1. Visualizations of retrieval results

To better understand the retrieval process of our unified interactive garment retrieval, we visualize some retrieval results in Figure Figure 5 and Figure 6. It shows that given a sentence, our model captures both concrete and abstract semantics, including fine-grained attributes and various garment categories. Besides, many failure cases are also provided in Figure 7 and Figure 8 to better understand our model's performance. Even for the failure cases, our model also provides very reasonable predictions.

E.2. Visualizations of learned latent spaces

To gain insights into the latent spaces learned by our model, we provide t-SNE [18] visualizations for features processed by projection modules in two branches. Figure 9(a) and 9(b) illustrate the latent space learned in TGR and VCR branch, respectively. Both of them demonstrate that our model can learn meaningful latent spaces, where the clusters contain garments with similar appearances. Specifically, the latent space of the TGR branch mainly focuses on the semantic visual similarity among garments, demonstrating that our TGR branch is superior in learning visual attributes. Nevertheless, the latent space of the VCR branch does not have a clear boundary as those in the TGR branch. It seems to pay more attention to the common features of different categories, demonstrating its ability to measure visual compatibility across categories.

(a) is scoop neck and change color to mustard



(b) is a crop top and has round neck neckline type

(c) is plunging neckline and with zip up opening type

(d) has flap type pocket and change color to maroon

(e) is applique and with loose fit silhouette

Figure 5. Retrieval results of our multi-task model on TGR subset. Yellow: reference garment; Red: target garment (ground truth); Blue: other retrieved garments.

(a) search a brown shoe that matches this jacket best

(b) retrieve a hat having a similar style with current skirt

(c) retrieve a yellow shoe having a similar style with current skirt

(d) search an above the hip length jacket that matches this shoe best

(e) retrieve a zip up skirt having a similar style with current top

Figure 6. Retrieval results of our model on VCR subset.

(a) is a sheath dress and has printed textile techniques

(b) is napoleon lapel and no special manufacturing technique

(c) is a leggings and has curved fit silhouette

(d) is a tank top and is short length

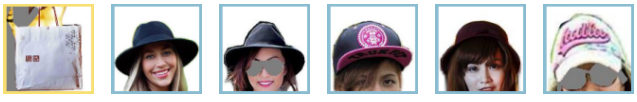
(e) change non-textile material to plastic and change decorations to ruffle

Figure 7. Failure cases of our multi-task model on TGR subset.

(a) replace this shoe with a jacket that has lining textile techniques

(b) retrieve a blazer jacket having a similar style with current belt

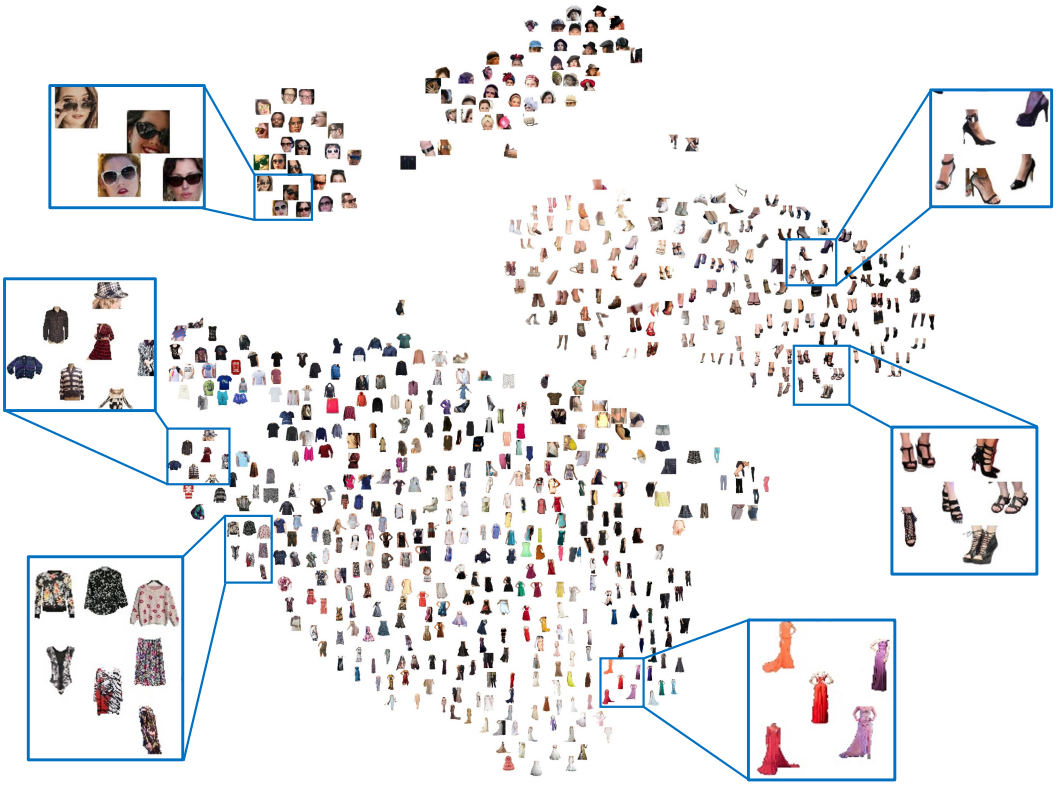
(c) for this shoe, find a visually compatible tights with black color



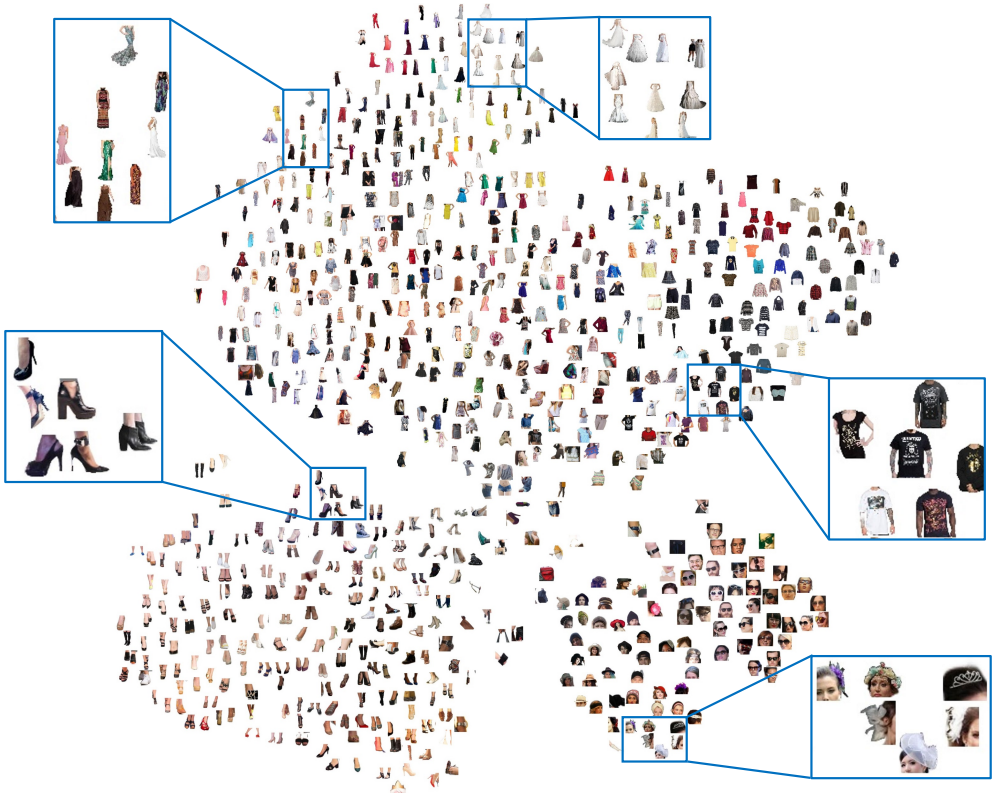
(d) retrieve a hat having a similar style with current bag

(e) retrieve a gem dress having a similar style with current shoe

Figure 8. Failure cases of our multi-task model on VCR subset.



(a) Visualized latent space of TGR branch.



(b) Visualized latent space of VCR branch.

Figure 9. t-SNE visualizations for two latent spaces learned via our multi-task model. Best zoom in and view in color.

References

[1] Yanbei Chen, Shaogang Gong, and Loris Bazzani. Image search with text feedback by visiolinguistic attention learning. In *CVPR*, 2020. 3, 4, 5

[2] Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *EMNLP*, 2014. 4

[3] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Fei-Fei Li. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 2, 4

[4] Xiao Han, Sen He, Li Zhang, and Tao Xiang. Text-based person search with limited data. In *BMVC*, 2021. 4

[5] 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. 1

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 4

[7] Yuxin Hou, Eleonora Vig, Michael Donoser, and Loris Bazzani. Learning attribute-driven disentangled representations for interactive fashion retrieval. In *ICCV*, 2021. 3

[8] 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. 1

[9] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 4

[10] Seungmin Lee, Dongwan Kim, and Bohyung Han. Cosmo: Content-style modulation for image retrieval with text feedback. In *CVPR*, 2021. 3, 4, 5

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

[12] Zheyuan Liu, Cristian Rodriguez-Opazo, Damien Teney, and Stephen Gould. Image retrieval on real-life images with pre-trained vision-and-language models. In *ICCV*, 2021. 4

[13] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. 3

[14] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*, 2019. 4

[15] Xuebin Qin, Zichen Zhang, Chenyang Huang, Chao Gao, Masood Dehghan, and Martin Jagersand. Basnet: Boundary-aware salient object detection. In *CVPR*, 2019. 1

[16] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021. 4

[17] 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, 3, 4, 5

[18] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *JMLR*, 2008. 5

[19] 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. 2, 3, 4, 5

[20] 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, 4, 5

[21] Mingchen Zhuge, Dehong Gao, Deng-Ping Fan, Linbo Jin, Ben Chen, Haoming Zhou, Minghui Qiu, and Ling Shao. Kaleido-bert: Vision-language pre-training on fashion domain. In *CVPR*, 2021. 1