IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 30 November 2023, accepted 15 December 2023, date of publication 25 December 2023,
date of current version 3 January 2024.

Digital Object Identifier 10.1109/ACCESS.2023.3346933

RESEARCH ARTICLE

Text-Conditioned Outfit Recommendation With Hybrid Attention Layer

XIN WANG[1] AND YUEQI ZHONG[1,2]
[1]College of Textiles, Donghua University, Shanghai 201620, China
[2]Key Laboratory of Textile Science and Technology, Ministry of Education, Shanghai 201620, China

Corresponding author: Yueqi Zhong (zhyq@dhu.edu.cn)

This work was supported by the Shanghai Natural Science Foundation under Grant 21ZR1403000.

ABSTRACT Text-conditioned outfit recommendation aims to recommend a whole fashion outfit that satisfies the compatibility between the recommended items and given items and adheres to the text condition like "Paradise Tropical Vacation" or "60s Style". Using text description as a condition can provide users with a flexible and accurate way to retrieve and recommend fashion items but this problem is underexplored by existing studies. A challenge of text-conditioned outfit recommendation is how to encode and fuse the outfit text description and fashion item images and text. To solve this, this paper proposes a framework for this task which features a hybrid attention layer that constructs the relationship between outfit text description and fashion items for condition compliance, and the relationship between fashion items for internal compatibility. To encode fashion item features, our method uses pre-trained FashionCLIP as an extractor which significantly reduces the trainable parameters compared to previous methods training CNN from scratch. The whole outfits are generated by iteratively adding compatible items based on a given partial outfit. Compared with state-of-the-art methods on polyvore disjoint and non-disjoint datasets, our approach can achieve 3% relative improvement in compatibility prediction AUC, achieve 5% relative improvement in fill-in-the-blank accuracy; achieve 19% relative improvement on complementary item retrieval recall at different ranks in average. Besides, We demonstrate that our approach can recommend a whole outfit with inner compatibility and adhere to the text description.

INDEX TERMS Fashion recommendation, conditional recommendation, multimedia recommendation, visual fashion analysis, transformer.

I. INTRODUCTION

Recommending an outfit conditioned on text involves using existing fashion items in a database to create an outfit that satisfies the compatibility between items and complies with the text condition. This task leads to a more accurate and flexible fashion recommendation experience. For the accuracy aspect, depending on the outfit text description, the same partial outfit can be used to generate different compatible outfits. For example, as illustrated in Fig. 1, a floral blouse can be used to create an outfit with bikini bottoms for a "Paradise Tropical Vacation" or an outfit with retro-style shoes and shorts for a "60s style" theme. The outfit text can constrain the recommendation for users need. Without a text description, the number of possible outfits would be much larger. For the flexibility aspect, users can use natural language as the condition to compose an outfit instead of providing an image [1] or using limited style category [2], which is more convenient and intuitive. The benefit for society of building a better fashion recommendation system includes improving the shopping experience, improving business efficiency, empowering fashion economics, providing small brands more visibility, etc. Besides, using natural texts as a condition can adapt to more scenarios than using style categories condition. Therefore, it can reduce the environmental impact of the training model since it can reduce the model retraining times.

Recent fashion recommendation studies have explored complementary item retrieval (CIR) task [3] and style-conditioned outfit recommendation [2]. These tasks are similar to our task, but cannot achieve the goal of generating

The associate editor coordinating the review of this manuscript and approving it for publication was Mingbo Zhao.

FIGURE 1. Compatible outfit conditioned on different outfit text descriptions.

outfits from text with internal compatibility and condition compliance. The CIR task aims to retrieve the last item given a partial outfit, the situation when there are not sufficient items in the given outfit is not discussed. Moreover, most CIR approaches [3], [4], [5], [6] do not consider the text condition, this may have little effect when there are many items are given as constraints, however, when there are few given items the generated outfit without text condition can be less accurate to users' need. Style-conditioned outfit recommendation [2] is conditioned on style categories with limited numbers, which is less flexible than outfit text description. Li et al. [7] has the same goal as our work, but they implemented this by forcing each item representation, instead of an outfit representation, to be close to the outfit text description. To the best of the authors' knowledge, very few studies have been conducted to recommend outfits conditioned on text description.

One challenge of outfit recommendation conditioned on text is how to efficiently encode the outfit images and text, as well as the outfit text description. There are two steps to encode an outfit into an embedding,: 1) encode each item image and text in the outfit; 2) fuse the item embeddings into an outfit embedding. For encoding fashion items images and text, early works trained a convolutional neural network from scratch in an end-to-end manner, but this requires many parameters when the item number in the outfit is large. Recent work finds pretrained CLIP model with fashion domain data [8] named FashionCLIP can achieve similarly or outperform classification and retrieval tasks, therefore, we use FashionCLIP to encode fashion item images, as well as outfit text description. As a result, we find it performs similarly to a model trained from scratch in the compatibility tasks. For fusing outfit representation, we use transformer [4] to encode multiple item features. To construct the relationship between outfit text description and fashion items, we tried to use cross attention [9], which put text description as the key and value of the attention layer, and item features as the query, but we find this design does not have interaction between text description and item features, and between item themselves. Therefore, we propose a hybrid attention layer to implement these interactions.

We introduce a framework implementing the above ideas for recommending outfits conditioned on text. Firstly, we use FashionCLIP to extract fashion item features instead of a trainable convolutional neural network. Then a hybrid attention layer is used to interact with outfit text descriptions and fashion items and within items. The outfit is generated in an iterative way each step adding one compatible based on the given partial outfit and outfit text description. To compare with previous work, we evaluate the performance of compatibility prediction, fill-in-the-blank, and complementary item retrieval tasks on the Polyvore and Polyvore-D datasets [6]. The experiment shows that our method can achieve state-of-the-art performance on all tasks with outfit text conditions. To evaluate the outfit recommendation quality and condition compliance quantitatively, we propose two metrics: average cosine similarity and conditional compatibility prediction score. The experiment results show that our approach can recommend outfits with internal compatibility and condition compliance.

To summarize, our main technical contributions are:

- We introduce a framework for text-conditioned outfit recommendation. It features hybrid attention to interact between outfit text descriptions and item features.
- We show that using FashionCLIP features can achieve similar performance with a trainable CNN model in compatibility tasks, and it is parameter efficient.
- We conducted experiments to show that our approach can achieve state-of-the-art performance on compatibility prediction, fill-in-the-blank, and complementary item retrieval tasks on Polyvore and Polyvore-D datasets. We also propose two metrics to evaluate the outfit recommendation quality and condition compliance quantitatively.

II. RELATED WORK

A. FASHION COMPATIBILITY RECOMMENDATION

The studies of fashion compatibility recommendation so far can be grouped into two ways. One group focuses on pairwise compatibility. It trains a model to transform fashion item image or text information into embedding, and then calculate the distance between each pair of items as their compatibility. These methods can be extended to outfit compatibility prediction by using the average of each distance as the score. When conducting complementary item retrieval, the candidate item has to be calculated with each existing item in the outfit. The lack of outfit representation makes it have to do more calculations and achieve suboptimal performance. To get outfits containing multiple fashion item images, the strategy includes using clean product images or integrating

detection technique with whole-body images [10], [11]. The other group of methods focuses on outfit representation. It firstly extracts features from each fashion item image and text by a trainable CNN [12] or pretrained models [13], then encodes the whole outfit using a sequence model such as LSTM [12] or transformer [4]. The outfit representation can be used for outfit compatibility prediction and retrieval. Currently, the transformer method shows advantage than other methods in outfit compatibility tasks according to [4].

Context is important for fashion compatibility, even with the same partial outfit, different outfits can be recommended under different contexts. The context can be user perference [14], [15], [16], style category [2], body shape [17], scene image [1], or outfit text descripition. We propose to use an outfit text as context because it has advantages such as being easier to get than an image and more flexible to change than categorical style information. This was firstly proposed by Li et al. [7]. They implemented this idea by forcing each item embedding close to the global text embedding. Very few studies follows this work to the best of our knowledge. Our study aims to push the boundary of this task further.

B. VISION-LANGUAGE PRETRAINED MODELS IN FASHION DOMAIN

The Vision-Language Pre-Trained Models (VL-PTMs) can be used for many downstream tasks such as cross-modal matching (retrieval), cross-modal reasoning (VQA), content generation (image captioning) without retraining on a specific dataset. It not only takes fewer parameters but also achieves better performance according to recent research [18]. While many VL-PTMs are trained in the general domain, there are some studies pretrained models in the fashion domain by taking care of the characteristics of the fashion domain. The pretrained vision-language models in the fashion domain can achieve better performance on fashion tasks.

FashionBERT [19] used a single-stream architecture similar to VL-BERT [20]. It uses patch-based representation on the image side to make the model pay more attention to fine-grained information since the RoIs used in the general domain [18], [21] are not suitable here. KaledioBERT [22] improves previous work by adopting three strategies for the fashion domain: 1) use different scales to split the image into patches; 2) mask text-aligned area in images; 3) design different pretraining tasks for better image understanding. More recently, MVLT [23] proposes masked raw image reconstruction as the pre-trained task to enhance the model image side.

Apart from BERT structure, FashionCLIP [8] uses CLIP architecture [24] and contrastive learning to pretrain the model, it achieves better performance on zero-shot classification and retrieval on four fashion datasets than CLIP models. As observed in the general domain [25], the BERT structure pretrained model performs better on vision-language understanding tasks such as VQA, and dual encoder like CLIP performs better on retrieval tasks.

C. CONDITIONAL CONTENT GENERATION

Conditional content generation has achieved impressive success in image [9], [26], music [27], and video [28] fields. To enable the generative model to generate content conditioned on text, they use cross attention layer [29], [30] which uses both context and intermediate content representation for the output calculation. The context condition is usually preprocessed into a fixed-length vector, therefore the condition can be from different modalities such as text, sketch for images, whistle for audio, and so on.

However, we find that the cross attention layer is not enough for text-conditioned outfit recommendation, since it does not have an interaction between text and item features, and between item features. Therefore, we extend the cross attention layer to the hybrid attention layer to recommend outfits conditioned on text prompts, the text prompts are encoded by pretrained FashionCLIP model [8].

III. PROPOSED METHOD

To recommend outfit conditioned on outfit text description, our framework retreives next item iteratively based on outfit text and given items, which is similar to generative language model predicts next word based on previous words. Fig. 2 shows the single step of recommending next item. The partial fashion items and outfit text descripitions are encoded into feature vectors. To handle outfits with uncertain item numbers, the features will be padded with zeros into a fixed maximum item number. Since padded values are zero, the padding position will not affect other positions after attention layers. Then, the padded features will be fed into a transformer model with hybrid attention layer for interaction. The output target embedding will be used for retrieving the next fashion item.

In the following content, Section III-A will introduce how we encode the fashion item images and text. Section III-B will introduce how we use hybrid attention layer to generate the target embedding conditioned on outfit text and given items. Section III-C will introduce the outfit generation procedure.

A. REPRESENTATION OF FASHION ITEMS

We use pretrained model to extract image and text features of fashion items instead of a trainable CNN like previous works [4], [6], [12]. Specifically, we use FashionCLIP to encode fashion item images and outfit text description. Additionally, we use SentenceBERT [31] to encode fashion item text description follows [4] to provide a rich information. We do not use FashionCLIP to encode fashion item text since the image feature and text feature of the same item can be similar in a CLIP model. We chose a multilingual version (distiluse-base-multilingual-cased-v2) of SentenceBERT since there are different languages in the fashion item text.

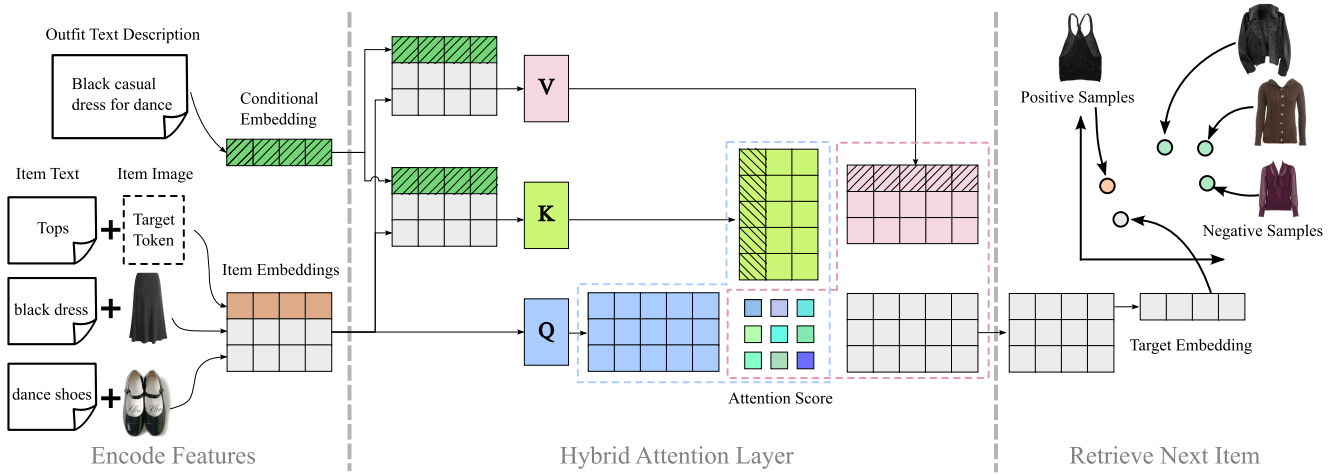

FIGURE 2. The illustration of the hybrid attention layer for generating the next fashion item based on the outfit text description and given fashion items.

Formally, in each single item recommendation step, our model will use an outfit text description T_o, partial outfit with image and text description $\{(I_1, T_1), (I_2, T_2), \ldots, (I_L, T_L)\}$, and target item category T_{tgt} as input. We use FashionCLIP to encode the outfit text description, which can be annotated as $\varphi(T_o) \in \mathbb{R}^{d_1}$. For each item in the outfit, we use FashionCLIP to encode the image and SentenceBERT to encode the text description, which can be annotated as $\varphi(I_i) \in \mathbb{R}^{d_1}$ and $\tau(T_i) \in \mathbb{R}^{d_2}$, where d_1 and d_2 are the dimensions of the feature vector encoded by FashionCLIP and SentenceBERT respectively. Before feeding into the following hybrid attention layer, image features and text features are transformed into the same dimension by another linear layer. The transformed features can be annotated as $\varphi'(I_i) \in \mathbb{R}^{d_3}$ and $\tau'(T_i) \in \mathbb{R}^{d_3}$. The feature of each item can be written as $F_i = (\varphi'(I_i) \| \tau'(T_i))$, where $\|$ is the concatenation operation. Multiple item features can be represented as a matrix $F \in \mathbb{R}^{L \times (d_3+d_3)}$. We add a special target token combined with target item category text at the first place of the input. The output at this place is considered as the target embedding. The target token and encoded category text can be represented as $(x_{\text{tgt}} \| \tau'(T_{tgt})) \in \mathbb{R}^{(d3+d3)}$. The outfit text description embedding is also transformed by a linear layer as $\varphi'(T_o) \in \mathbb{R}^{(d_3+d_3)}$ so it can be concatenated with F for the following hybrid attention layer.

B. HYBRID ATTENTION LAYER

The hybrid attention layer is used to to create a mixed understanding of partial outfit and outfit text description. The hybrid attention layer is defined similar to other attention layer as $\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V$, but the difference is the setting of query, key and value as is shown in Fig. 2.

The query contains not only the fashion item features but also a speical target token x_{tgt}. The target token position output a global representation of outfit similar to ViT model [32]. The target token combined with target item text can be used to retrieval next item with item text descrption condition, but in our experiment, we only use category such as "Tops" as the item text description.

For the key and the value, the self-attention layer uses the same input as the query which is item features in our problem, but the cross attention layer uses the condition representation that is outfit text description. Using the self-attention layer does not consider the condition but the interaction within items. Using the cross attention layer will get output with condition representation but lose the interaction within the items. Our idea is to leverage the advantage of both sides, the operation is to concatenate both item features and condition representation into key and value therefore the output will have the conditioning content and interaction with item features. The query, key, and value can be written as:

$$\begin{aligned} Q &= W_Q \cdot [(x_{\text{tgt}} \| \tau'(T_{\text{tgt}})) \| F], \\ K &= W_K \cdot [\varphi'(T_o) \| F], \\ V &= W_V \cdot [\varphi'(T_o) \| F]. \end{aligned} \tag{1}$$

The hybrid attention layer is used to build a transformer decoder model. We use the output embedding c at the target token position for retrieving the next item. Another MLP layer is added to it for model capacity $t = \epsilon_{mlp}(c)$. During the training, the learnable parameters including transformer decoder parameters ϵ_{trans}, MLP layers ϵ_{mlp}, target image token x_{tgt}, and linear encoder of image and text features.

The model parameters are optimized with a margin ranking loss. It forces the target item embedding moves closer to the positive embedding and farther apart from the negative embeddings. The target item embedding is compared with ground-truth embedding and multiple negative samples. We sample relatively easy negatives at the early stage of training and sample relatively hard samples by randomly choosing from the same fine-grained categories candidates at the later stage. We use cosine similarity as the distance function beucase it has several advantages such as: insensitivity to magnitude, suitable for high-dimensional data, and

robustness to outliers. The loss function can be written as:

$$
\begin{aligned}
L(\mathrm{t}, p, N) &= L(\mathrm{t}, p, N)_{All} + L(\mathrm{t}, p, N)_{Hard} \\
L(\mathrm{t}, p, N)_{All} &= \frac{1}{|N|}\sum_{j=1}^{|N|}\left[d\left(\mathrm{t}, f^{p}\right) - d\left(\mathrm{t}, f_{j}^{N}\right) + m\right]_{+} \\
L(\mathrm{t}, p, N)_{\mathrm{Hard}} &= \left[d\left(\mathrm{t}, f^{p}\right) - \min_{j=1\ldots|N|} d\left(\mathrm{t}, f_{j}^{N}\right) + m\right]_{+},
\end{aligned}
\tag{2}
$$

where p is the positive sample, and N is the multiple negative samples.

C. OUTFIT GENERATION PROCEDURE

Our approach recommend outfits by iteratively retrieving next item. In each step, the transformer will generate target item embedding for retrieval based on given information. If there is no item in the partial outfit, only the target item token and outfit text will be input into the model.

To sample the next item, the item candidates in the dataset will be ranked based on the cosine similarity with the target item embedding. While evaluating metrics, the top-1 item will be selected as the generated item for reproducibility. To create diverse outfits, the candidates can be sampled with weights such as the cosine similarity score.

IV. EXPERIMENT

A. DATASET

To conduct experiment, it requires the dataset contains each item image and text in the outfits and outfit text description. Therefore, the polyvore non-disjoint dataset (PO) and polyvore disjoint dataset (PO-D) [6] are chosen for the experiment. Both PO and PO-D dataset are collected from polyvore.com which is a fashion website that people share their outfits. The difference is the rule to split the training, validation, and test set. In PO-D dataset, no garment item appears in more than one split which is achieved by a graph segmentation algorithm, but in PO dataset one garment item could be in multiple splits. The PO dataset has 53,306 outfits in training split, 5000 outfits in validation split, 10,000 in test split, totally 365,054 unique items; the PO-D dataset has 16,995 outfits in training split, 3000 outfits in validation split, 15,154 outfits in test split, totally 175,485 unique items.

B. IMPLEMENTATION DETAILS

Hyperparameters The text of the fashion item is encoded by a pretrained multilingual version (distiluse-base-multilingual-cased-v2) of SentenceBERT [31] since the descriptions in the dataset use different languages. The fashion item image and text embeddings are linearly mapped to 64 dimensions, the global text embedding is mapped to 128 dimensions. The dimension numbers are the same as the previous study [4], therefore we can conclude that the performance improvement is from hybrid attention instead of other components. The maximum item number for padding features is 16 for PO-D dataset and 19 for PO dataset. We use a transformer decoder with 3 layers and 16 heads and found that the model capacity is enough for our task. The margin of the ranking loss is set as 0.3. For each sample, we sample 10 negative samples per positive sample. We trained the model for 100 epochs with batch size 50. The model is optimized with ADAM optimizer, the initial learning rate is 5e-5, and the learning rate is decayed by 0.5 every 10 epochs. We implemented this framework with PyTorch. A single NVIDIA GeForce RTX 3090 GPU with 24GB memory was used to accelerate the computation. We preprocessed the images and text into embeddings before training the model. After that, it takes about 2 hours to finish the training on PO-D dataset and 5 hours on PO dataset thanks to removing trainable CNN model.

Hard Negative Sampling The hard negative samples are defined as the fashion item with the same fine-grained category with the target. Since polyvore dataset has both high-level category and fine-grained category for each item, we use easy negative samples that has the same high-level cateogry with targe fashion item in the first 40 epochs, then use hard negative samples for the rest of the training.

1) PRETRAINING BY COMPATIBILITY PREDICTION TASK

We find initializing model parameters with compatibility prediction task can improve the performance. Different from previous work [4], our initialization only influences the transformer parameters which model the relationship between outfit text description and fashion item features.

TABLE 1. Compatibility Prediction Performance of different methods.

Method	Features	PO-D	PO
BiLSTM + VSE [12]	Img + Text	0.62	0.65
GCN (k=0) [33]	Img	0.67	0.68
SiameseNet [6]	Img	0.81	0.81
Type-Aware [6]	Img + Text	0.84	0.86
SCE-Net [5]	Img + Text	-	0.91
CSA-Net [3]	Img	0.87	0.91
OutfitTransformer [4]	Img + Text	0.88	0.93
OutfitCoherence [7]	Img + Text	0.901	0.928
Ours (CIR w/o cond)	Img + Text	0.899	0.933
Ours (CIR w cond)	Img + Text	0.917	0.956
Ours (CP w/o cond)	Img + Text	0.923	**0.956**
Ours (CP w cond)	Img + Text	**0.930**	0.954

C. COMPARE WITH STATE-OF-THE-ART

We compare our approach with the state-of-the-art baselines including: Bi-LSTM [12], CSN [6], GCN [33], SCE-Net [5], CSA-Net [3], OutfitTransformer [4], OutfitCoherence [7]. The following common outfit compatibility tasks are used for the comparison:

- **Compatibility Prediction (CP)** is to use a binary compatibility classification model to predict whether the generated outfit is compatible which was firstly proposed by [12]. It uses Area under Curve (AUC)

of ROC curve of the predicted score as the evaluation metric. The higher the AUC, the better the performance.

- **Fill in the Blank (FITB)** is to test whether the model can correctly choose the target compatible item to complete a partial outfit while there are 3 other distractive candidates, which was proposed by [12]. It uses the accuracy of correctly answering the questions as the evaluation metric. Higher accuracy means better performance.
- **Complementary Item Retrieval (CIR)** evaluate model ability to retrieve target complementary item from the database to complete a partial outfit, which was proposed by [3]. It uses recall at k (R@k) as the evaluation metric, which is also often used by image retrieval tasks [11], [34], [35], [36], [37]. The definition of recall at k is the times the ground-truth item appears in the top-k retrieved items divided by the number of test samples, which can be written as:

$$\mathrm{R@k} = \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}(r_i \leq \mathrm{k}), \tag{3}$$

where N is the number of test samples, r_i is the rank of the ground-truth item in the retrieved items, and $\mathbb{1}(r_i \leq \mathrm{k})$ equals to 1 if r_i is less than k otherwise 0. Higher R@k means better performance. We chose R@10, R@30, and R@50, so the metric values can be larger and we can conclude that the change is caused by method difference instead of randomness.

We trained two models with our approach: one for CP task and one for CIR task respectively. When the model is trained on CP task, there is a binary classifier head append after the target embedding, we use a binary cross entropy loss instead of ranking loss during CP task. We train our model on the CP task for two purposes: one is for initializing model parameters when trained on the CIR task, and another is for evaluating whether the geneted outfit is compatible. The model trained on CIR task can be used for CP, FITB, CIR tasks. We trained our model with and without outfit text description.

CP Performance with Model Trained by CP Task The results of CP task of our model and compared methods are shown in Table 1. It can be seen that when there is no condition, the performance is still superior to the baselines. This indicated that the FashionCLIP can extract effective features for items. When there is a condition, the performance is further improved, this says the outfit compatibility is influenced by outfit text condition. This improvement also says our framework can effectively use both the fashion item information and outfit text description.

CP Performance with Model Trained by CIR Task Even though when our model is trained with CIR task, the trained model can be directly used for the CP task. We achieve this by iteratively inputting the partial outfit items to get the target item embedding, then using the average distance between the target item embedding and the next ground-truth item as the score of compatibility. The result of our methods is shown in Table 1. It can be observed that when the model is trained without conditioning, the performance on both PO-D and PO set is similar to [4] and [7]. When using the hybrid attention layer with conditioning, the performance is superior to other compared methods. This result says that the model trained with CIR task can generate target embedding similar to the ground-truth item embedding. For the CP task, the model trained on CP task performs better than the model trained on the CIR task, this is because there is an additional binary classifier head which can better fit the CP task.

FITB and CIR Performance The performance on the FITB and CIR tasks are both evaluated with a model trained on the CIR task. They are both achieved by retrieving with target item embedding from candidates. On CIR task, we use the same setting as CSA-Net [3]. In detail, we set 3000 candidates with the same fine-grained category for each retrieval. When there are not enough candidates in the test split, we used the items from the training split to fulfill. We filtered out the fine-grained category with not enough items. After that, there are 20/152 fine-grained categories are kept in the PO-D dataset, and 33/143 categories are kept in the PO dataset.

The result of the FITB and CIR task is shown in Table 2. When there is no conditioning, we found the model can also perform better than OutfitTransformer on PO-D which trained ResNet50 from scratch, but our method has lower in R@10, which could be due to ResNet50 can extract better fine-grained features from pixels. Our method also performs lower than OutfitTransformer on PO dataset, the reason is PO dataset has more data therefore the ResNet50 can be better trained than on PO-D dataset. When there is condition, our model can use the outfit text description to make a better prediction, it performs better than other compared methods on all metrics. Our method performs better than OutfitCoherence [7], which also uses outfit text as condition. This indicates that the transformer architecture with hybrid attention layer can better use the outfit text description and fashion item features.

D. ABLATION STUDY

We evaluate the contribution of different components in our method including: 1) compare hybrid attention with cross attention, 2) the effect of hard negative sampling, 3) the effect of pretrained model on CP task, 4) compare CLIP and FashionCLIP features.

1) COMPARE WITH CROSS ATTENTION

To show the effect of hybrid attention layer, we compare our model with cross attention layer. The setting of cross attention layer in our experiment is make the key and value in the attention layer be only the outfit text description embeddings, it is identical to make the K and V tensors in Fig. 2 only have the green part. The result is shown in Table 3. It can be observed from the third and fourth rows that the hybrid attention layer can improve the performance on both FITB and CIR tasks with text-conditioning and

TABLE 2. FITB and CIR performance of different methods.

Method	Polyvore Outfits-D				Polyvore Outfits			
	FITB	R@10	R@30	R@50	FITB	R@10	R@30	R@50
Type-Aware [6]	55.65	3.66	8.26	11.98	57.83	3.50	8.56	12.66
SCE-Net Average [5]	53.67	4.41	9.85	13.87	59.07	5.10	11.20	15.93
CSA-Net [3]	59.26	5.93	12.31	17.85	63.73	8.27	15.67	20.91
OutfitTransformer [4]	59.48	6.53	12.12	16.64	67.10	9.58	17.96	21.98
OutfitCoherence [7]	62.80	-	-	-	66.10	-	-	-
Ours (w/o cond)	63.04	6.10	13.24	18.81	65.32	6.81	14.46	20.38
Ours (w cond)	**65.78**	**7.35**	**16.22**	**22.19**	**70.33**	**10.12**	**19.49**	**26.17**

TABLE 3. Ablation Study. The Cond, Hard, CA, HA, Pre stands for text-conditioning, hard negative sampling, cross attention, hybrid attention and CP pretraining respectively.

Line Number	Method					Polyvore Outfits-D				Polyvore Outfits			
	Cond	Hard	CA	HA	Pre	FITB	R@10	R@30	R@50	FITB	R@10	R@30	R@50
1						61.93	5.68	12.49	17.67	65.32	6.81	14.46	20.38
2		✓				63.04	6.10	13.24	18.81	68.37	8.50	17.66	23.92
3	✓	✓	✓			63.96	6.41	14.31	19.99	68.59	9.35	18.54	24.77
4	✓	✓		✓		65.42	6.87	14.97	21.28	69.77	9.75	19.19	25.48
5	✓	✓		✓	✓	**65.78**	**7.35**	**16.22**	**22.19**	**70.33**	**10.21**	**19.49**	**26.17**

TABLE 4. Ablation Study of CLIP and FashionCLIP features.

Method		Polyvore Outfits-D				Polyvore Outfits			
Feature	Setting	FITB	R@10	R@30	R@50	FITB	R@10	R@30	R@50
CLIP	Cond + Hard + HA	60.87	5.76	13.26	18.38	65.48	7.91	16.19	21.90
FashionCLIP	Cond + Hard + HA	**65.42**	**6.87**	**14.97**	**21.28**	**69.77**	**9.75**	**19.19**	**25.48**

TABLE 5. Average Recall@k at different position of different methods.

Length	No Condition			Cross Attention			Hybrid Attention			Counts
	R@10	R@30	R@50	R@10	R@30	R@50	R@10	R@30	R@50	
0	0.38	1.03	1.65	0.93	2.47	3.87	1.77	3.90	5.88	11463
1	4.24	9.71	13.98	4.62	10.44	15.18	5.43	12.10	16.53	9684
2	5.61	12.13	16.79	5.46	12.37	17.38	6.52	13.94	19.43	9022
3	6.51	13.58	19.11	6.28	13.89	19.24	7.14	15.38	21.16	6787
4	6.48	13.65	18.99	6.74	13.74	19.00	7.30	15.86	21.67	4614
5	6.77	14.69	19.97	6.31	14.31	20.08	7.15	15.30	21.69	2614
6	5.31	11.42	16.89	5.39	12.47	17.30	6.28	13.92	19.87	1243
7+	5.30	12.44	18.09	4.95	11.98	17.28	6.11	13.82	21.20	868
Avg	5.08	11.08	15.68	5.09	11.46	16.17	**5.96**	**13.03**	**18.43**	5787

hard negative smapling settings. It can also be seen from second and third rows that the cross attention layer performs better than self-attention layer. This indicates that outfit text descripition conditioning can improve fashion compatibility learning.

2) HARD NEGATIVE SAMPLING

As is shown in the first and second rows in Table 3, training with hard negative samples in the later epochs can improve the performance on both FITB and CIR tasks. This says the model can learn more from the hard negative samples.

3) THE EFFECT OF PRETRAINING

Table 3 shows that initializing parameters from the pretrained model on the CP task can improve the performance on both FITB and CIR tasks. The model trained by the CP task learns to extract outfit representation which is helpful to the FITB and CIR tasks.

4) COMPARE CLIP AND FASHOINCLIP FEATURES

We compare the model trained with CLIP and FashionCLIP features. The results are shown in Table 4. The model trained with FashionCLIP features can achieve better performance on

FIGURE 3. Attention visualization of different layers in the transformer decoder.

both FITB and CIR tasks. This indicates that the FashionCLIP can extract more effective features for fashion items.

E. ATTENTION VISUALIZATION

To understand how the outfit text embedding affect the output of the model, we visualize the attention map of the model. The result is shown in Fig. 3.

There are three characteristics we can see from the attention map. First, in the first row of attention, the top left position has the highest value no matter which layer. Since the first row is the target token position, it says that the features of the target token position are mostly influenced by the text condition embedding. Second, the first column has a higher value than other columns, it means output embedding at each postion are mostly influenced by the condition embedding. Third, Apart from the area with high value, other areas still have some value, which means the model still uses the information from the item embedding, this enables the model to construct a relationship between items and fuse the condition embedding and item embedding.

F. EVALUATING GENERATED OUTFITS

Few works attempted to evaluate the model's ability to generate a whole compatible outfit. OutfitCoherence [7] used cluster size and query-outfit coherence to evaluate generated outfit quality and condition compliance. But we argue that there is some inappropriateness. First is cluster size of conditional generated outfits may not have a smaller cluster size compared to unconditionally generated outfits, it could also be cluster translation; Second is their query-outfit coherence were calculated in a embedding space measuring the distance between the single item and query text instead of a fashion embedding space for general purpose, therefore their score is not strictly indicates the outfit coherence.

To evaluate recommended outfits qualitatively, we propose the following metrics:

- **Average Recall at K (Avg R@k) at Different Positions**: It is a metric that calculates the recall of ground-truth items in each step of outfit generation;
- **Average cosine similarity (ACS)**: It calculates the similarity between generated outfits and original outfits in FashionCLIP embedding space;
- **Conditional Compatibility Prediction Score (CCPS)**: It uses the score of compatibility prediction model to tell whether the generated outfits are compatible;
- **Query-Outfit Coherence**: It evaluates whether generated outfits are correlated with outfit text description which was proposed by [7].

The following contents will provide details of each evaluation technique during the experiment. Besides, qualitative evaluation was also conducted to show the results visually.

Average R@k at Different Positions Previous CIR task only evaluates performance to fulfill the last item in the outfit, but recommend a whole outfit need multiples steps therefore R@k for the last item is not enough to evaluate the model ability to generate a whole outfit. Avg R@k evaluates the ratio that target item appears in the top k retrieved items in every step. The formula of Avg R@k is:

$$\text{Avg R@k} = \frac{1}{P}\sum_{i=1}^{P}\frac{1}{N_i}\sum_{j=1}^{N_i}\mathbb{1}(r_{i,j} \leq \text{k}), \tag{4}$$

where P is the step position number ranging from 1 to 16 in PO-D dataset, and from 1 to 19 in PO dataset. N_i is the total evaluation times at postion i, $\mathbb{1}(r_{i,j} \leq \text{k})$ is equal to 1 when the rank of the ground-truth item $r_{i,j}$ is less than k otherwise 0.

We experimented on the test set of the PO-D dataset. For the first step, only the outfit text description was used as input to get target embedding. For other steps, one more ground-truth item is added into the partial outfit, the model uses a partial outfit combined with an outfit text description as input. The compared methods including the recommendation without condition, conditioned outfit recommendation using

cross-attention, and conditioned outfit recommendation using hybrid attention. The outfits has items without enough candicates were filtered, after that, 11463 outfits were kept in experiment.

The result is shown in Table 5. It can be seen that: 1) When parital outfit length is small such as 0 and 1, The cross attention layer and hybrid attention layer can achieve better top-k recall than self attention without condition. This is because the self attention setting uses existing items in the partial outfit to retrieve the next item, it cannot take advantage of the outfit text description. When there is no item provided, no information is provided for computing the output embedding for retrieval therefore the performance is low, the cross attention can use outfit text description to retrieve the next item, the performance is clearly better than the No Condition setting. 2) When the length is large, self attention is better than cross attention. This is because the cross attention layer has only text description in key and value, according to the attention formula $\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$,the output is a weighted sum of outfit text embeddings, it does not fuse information from fashion item features. When the length is larger than 0, the No Condition setting starts to use existing items as the information for recommendation but the cross attention cannot use this part of the information. 3) the hybrid attention can perform better than cross attention and self attention. This is because it makes the key and value a concatenation of outfit text description embedding and item embeddings, take advantage of both self attention and cross attention. When the length is 0, it can leverage the outfit text description even better than the cross attention setting, when the length is larger than 0, it performs better than other methods because using both item embeddings and outfit text description embeddings to compute the output embeddings for retrieval.

Average Cosine Similarity When there is enough given item in the partial outfit, the model should generate outfits visually similar to the original outfit, we can use the similarity between generated outfit and the original outfit to evaluate the model ability to generate compatible outfits with adherence. Therefore, we propose to use average cosine similarity (ACS) as metric which measure similarity between each item in the generated outfit and original outfit, then use the average as the outfit similarity. It can be written as:

$$\text{ACS} = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{P_i}\sum_{j=1}^{P_i}\frac{f_{i,j}\cdot f^*_{i,j}}{\|f_{i,j}\|\,\|f^*_{i,j}\|}, \tag{5}$$

where $f_{i,j}$ is the FashionCLIP embedding of the j-th item in the i-th generated outfit, $f^*_{i,j}$ is the FashionCLIP embedding of the j-th item in the i-th original outfit, N is the number of outfits, P_i is the number of items in the i-th outfit. We ignore the outfit without enough items during calculation.

The result of ACS is shown in Table 6. It can be seen that there is a trend that the more items are kept, the higher ACS will be achieved. This is reasonable since the more items are kept, the more constraint the model will have.

TABLE 6. The results of average cosine similarity (ACS) and conditional compatibility prediction score (CCPS).

Method	ACS			CCPS		
	Keep0	Keep1	Keep2	Keep0	Keep1	Keep2
No Condition	0.442	0.552	0.569	0.965	0.964	0.942
Cross Attn	0.500	0.565	0.580	0.978	0.970	0.946
Hybrid Attn	**0.502**	**0.566**	**0.582**	**0.988**	**0.979**	**0.962**

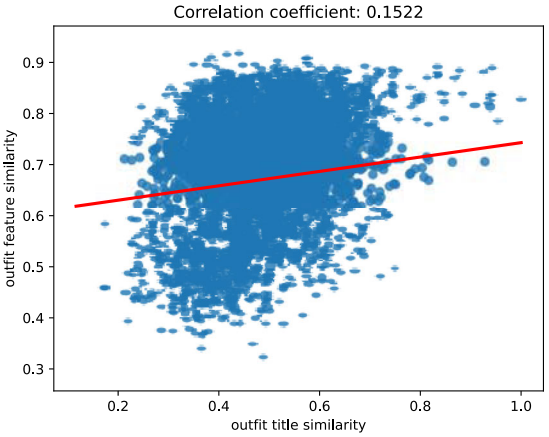

FIGURE 4. Illustration of correlation between outfit text description and outfit features.

The model with hybrid attention can achieve higher ACS than the model with cross attention, this is because the hybrid attention can build a relationship between items and a relationship between outfit text and items. The method with outfit text description can achieve higher ACS than the method without outfit text description. The ACS of the method with no condition significantly dropped because there first item is only generated with a category constraint therefore the generated outfit is highly possible to be different from the original outfit.

Conditional Compatibility Prediction Score Conditioned compatibility prediction model can predict whether an outfit is compatible with the outfit text description. We can use the prediction score as the metric to evaluate the model ability to generate compatible outfits and adhere to the text description. The result is shown in Table 6. It can be seen that more kept items lead to a relatively lower score, this may be because the compatibility prediction model learns a slightly different compatibility pattern from the dataset, less kept items can give the model more flexibility to generate the compatible outfit the model learned. In general, the table shows that the model with hybrid attention can achieve a higher score than the model with cross attention, and the method with condition can achieve a higher score than the method without condition by comparing different methods.

Query-Outfit Coherence To verify the recommended outfits adhere to text description, we choose 500 text descripition from the PO-D test set for recommending outfits. There are 124750 different pairs of different outfits. We calculate the correlation coefficient between the distance of outfit text description representation and the distance of outfit

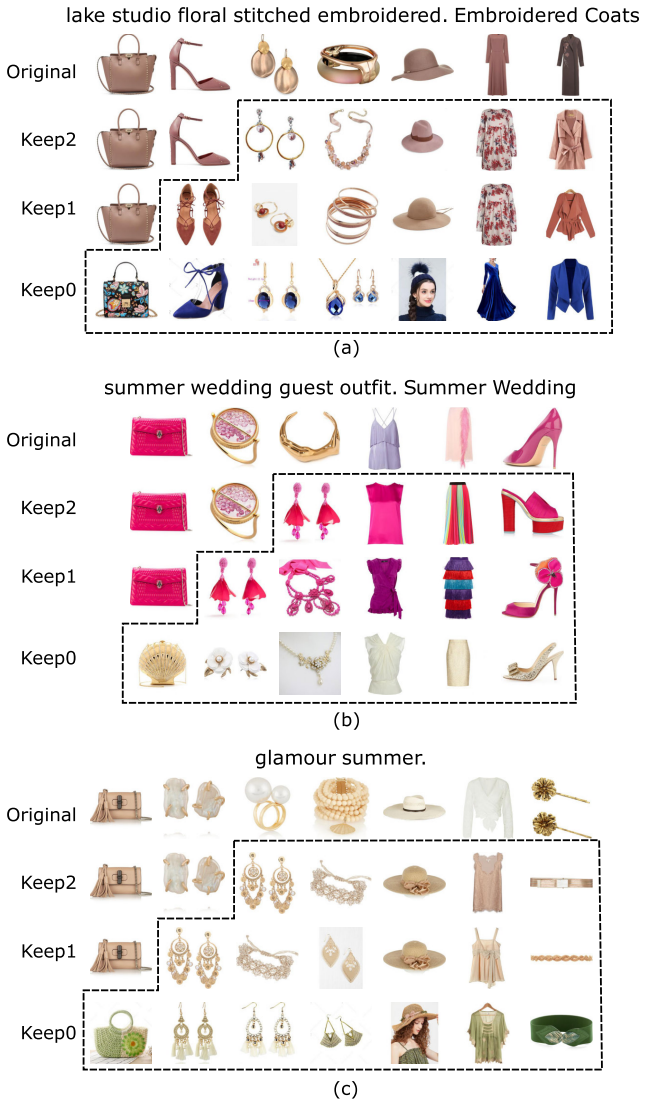

FIGURE 5. Compare the generated outfit using hyrid attention method and the original outfit in the dataset. Each outfit text description generates 3 outfits by keeping 0, 1, 2 items in the original outfit.

representation. Different from [7], our outfit representation is the mean vector of FashionCLIP features. The result is shown in Fig. 4, this indicates that the model can generate outfits with similar representation when the outfit text descriptions are similar.

Qualitative Evaluation of Generated Outfits We provide a qualitative evaluation by showing the generated outfits visually. The qualitative evaluation is conducted as follows: for each original outfit, we generate 3 different Outfits by keeping 0, 1, and 2 items in the original outfit. It is like a real scenario the user may provide only outfit text or provide both outfit text and partial items.

The results are shown in Fig. 5. It can be observed that when there is no kept item, the generated outfits still keep internal compatibility, but they can be different from the original outfit, especially in the color and pattern aspect. The difference is mainly due to different the first item since the following items should be compatible with the first item.

FIGURE 6. Visualization of generated outfit with no condition, with cross attention and with hybrid attention.

When the first item is kept, the generated outfit is under enough constraints to look similar to the original outfit. When there are 2 items kept, the difference between keeping 1 item is not very clear.

Compare Conditional Generation with Unconditional Generation We visualize generated outfit with no condition, with cross attention and hybrid attention in Fig. 6. It can be seen that the outfit generated with no condition can be inconsistent with the outfit text description, for example, in the first case with the query "mermaid for life", the unconditional generated outfit only tries to be compatible with the first black shoes, but conditional outfit generation will also try to adhere to the keyword "mermaid".

Visualizing Recommended Outfit Conditioned on Different Outfit Text To visually verify the recommended outfits can be conditioned on different text description, we conducted specific case study and the results are in Fig. 7. It can be observed that the model can recommend different outfits from different text descriptions. For example, the outfit based on the text "Sports" includes sports shoes while keep color similar to the bag. The outfit based on "Paradise Tropical Vacation" includes bikini style item. This

FIGURE 7. Visualization of recommended outfits with different outfit text description with test set of PO-D dataset.

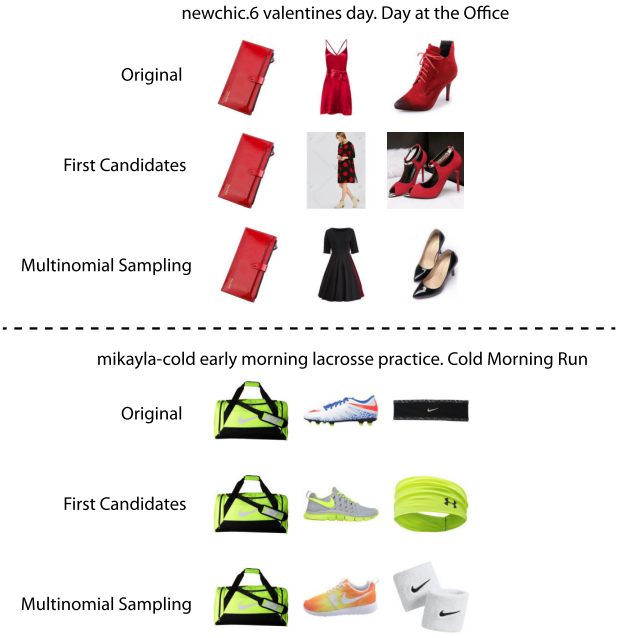

FIGURE 8. Visualization of recommended outfits with multinomial sampling.

result indicates that our model can generate different outfits from different text descriptions and corresponding to the text content.

Multinomial Sampling Strategy The recommendation results in Fig. 5 and Fig. 6 shows that an compatbile outfits recommended by the model tend to be similar color. One reason is that many compatible outfits in the dataset are in similar color. Another reason is our method uses a iterative

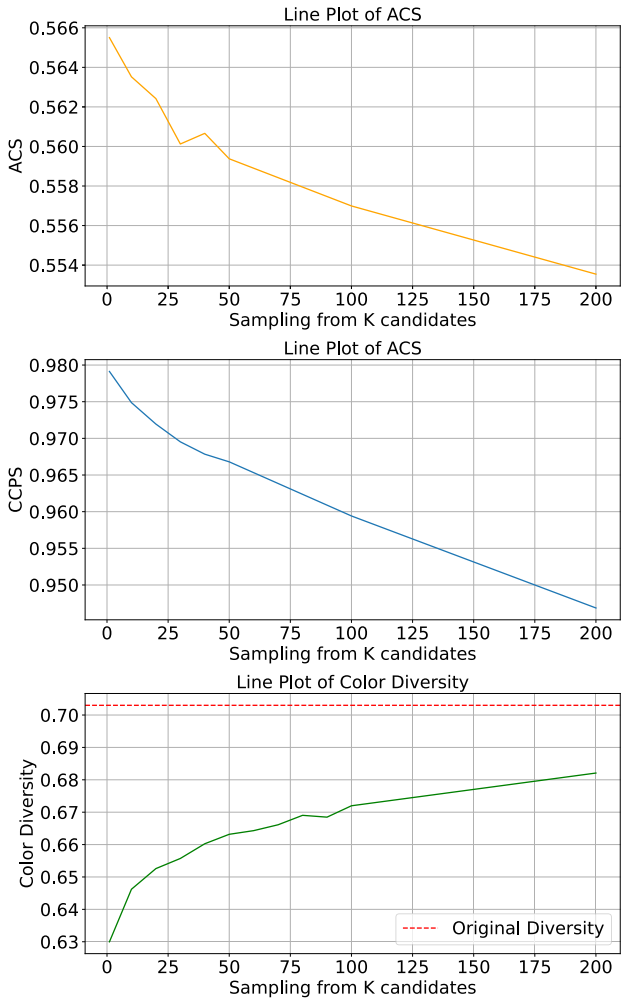

FIGURE 9. The effect of K value in multinomial sampling to CCPS, ACS, and color diversity.

process and tries to select the most compatible item in each step. For example given a red handbag, our retrieval algorithm gives a red dress the highest score in the compatible candidates as is shown in the Fig. 10. If the first and second item are in the same red color, the third item will be more possible in the same color. However in reality it still needs to consider the diversity of the outfit.

One solution is to use multinomial sampling during the recommendation. As shown in Fig. 10, the top-k retrieved compatible items by our method can be in different colors. Multinomial sampling can give the not first candicates chance to be chosen then the model can generate more diverse outfits. During experiment, we used the similarity value between target item embeddings and candicate embeddings as the possiblity for multinomial sampling. We visualized recommended outfit with only first candidates and with multinomial sampling in Fig. 8. It can be seen that the outfits with multinomial sampling are more diverse in color. Furthermore, we analyzed the effect choice of K candicates to the compatibility and color diversity quantitatively. We used ACS and CCPS to express the compatibility. To calculate the

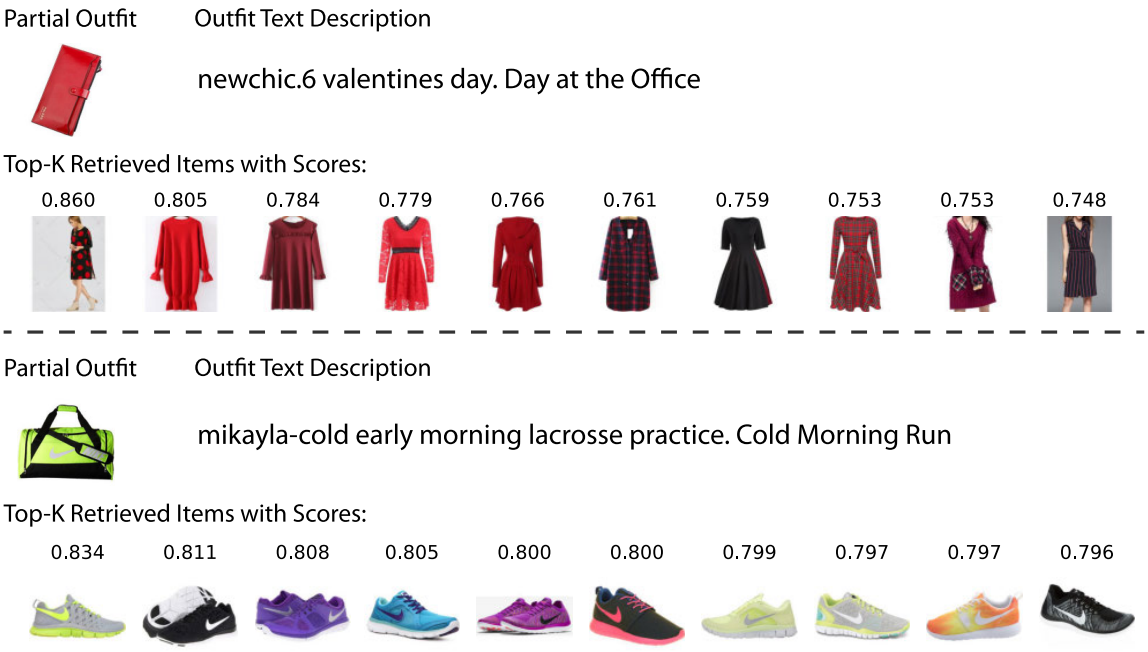

FIGURE 10. Visualization of top-k retrieved compatible candidates.

color diversity of an outfit, we first remove the background in item image with U^2-Net [38], then convert the image to LAB color space, and calculate the bhattacharyya distance of color histogram of every pair of items in the outfit. The color diversity of an outfit is the average of the distance of every pair of items. The result is shown in Fig. 9. It can be seen that with the K value increases from 1 to 200, the ACS and CCPS drops linearly, but their values still hold a hight level. This indicates that the recommended outfits are still compatible. The color diversity increases with the K value increases and converge to the color diversity of original outfits average, this indicates that the recommended outfits are more diverse in color.

G. LIMITATIONS

In the experiment, we found that the current method may not capture specific items in the outfit description, for example, if the outfit description is 'street style with a black t-shirt' while partial outfit items are all in red, then the generated outfit may still choose a red t-shirt. The reason is the condition conflict between outfit text and partial items. If the other items are all red, the model may choose a red t-shirt for compatibility consideration and ignore the outfit text.

V. CONCLUSION AND FUTURE WORK

This is study introduce an approach to recommend fashion outfit from text description. Our approach use hybrid attention layer to interact between outfit text description and outfit item, and within outfit items. We also use FashionCLIP to encode fashion item images and text efficiently and effectively. We evaluate our model on CP, FITB, and CIR tasks and show that it can achieve the state-of-the-art performance. We also illustrate that our model can recommend outfits with internal compatibility and outfit text compliance qualitatively and quantitatively.

REFERENCES

[1] T. Ye, L. Hu, Q. Zhang, Z. Y. Lai, U. Naseem, and D. D. Liu, "Show me the best outfit for a certain scene: A scene-aware fashion recommender system," in *Proc. ACM Web Conf.*, Austin, TX, USA, Apr. 2023, pp. 1172–1180.

[2] D. Banerjee, L. Dhakad, H. Maheshwari, M. Chelliah, N. Ganguly, and A. Bhattacharya, "Recommendation of compatible outfits conditioned on style," in *Advances in Information Retrieval*, vol. 13185, M. Hagen, S. Verberne, C. Macdonald, C. Seifert, K. Balog, K. Nørvåg, and V. Setty, Eds. Cham, Switzerland: Springer, 2022, pp. 35–50.

[3] Y.-L. Lin, S. Tran, and L. S. Davis, "Fashion outfit complementary item retrieval," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 3308–3316.

[4] R. Sarkar, N. Bodla, M. I. Vasileva, Y.-L. Lin, A. Beniwal, A. Lu, and G. Medioni, "OutfitTransformer: Learning outfit representations for fashion recommendation," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2023, pp. 3590–3598.

[5] R. Tan, M. Vasileva, K. Saenko, and B. Plummer, "Learning similarity conditions without explicit supervision," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 10372–10381.

[6] M. I. Vasileva, B. A. Plummer, K. Dusad, S. Rajpal, R. Kumar, and D. Forsyth, "Learning type-aware embeddings for fashion compatibility," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 390–405.

[7] K. Li, C. Liu, and D. Forsyth, "Coherent and controllable outfit generation," 2019, *arXiv:1906.07273*.

[8] P. J. Chia, G. Attanasio, F. Bianchi, S. Terragni, A. R. Magalhães, D. Goncalves, C. Greco, and J. Tagliabue, "Contrastive language and vision learning of general fashion concepts," *Sci. Rep.*, vol. 12, no. 1, p. 18958, Nov. 2022.

[9] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, New Orleans, LA, USA, Jun. 2022, pp. 10674–10685.

[10] F. Zeng, M. Zhao, Z. Zhang, S. Gao, and L. Cheng, "Joint clothes detection and attribution prediction via anchor-free framework with decoupled representation transformer," in *Proc. 31st ACM Int. Conf. Inf. Knowl. Manage.*, Atlanta, GA, USA, Oct. 2022, pp. 2444–2454.

[11] M. Zhao, S. Gao, J. Ma, and Z. Zhang, "Joint clothes image detection and search via anchor free framework," *Neural Netw.*, vol. 155, pp. 84–94, Nov. 2022.

[12] X. Han, Z. Wu, Y.-G. Jiang, and L. S. Davis, "Learning fashion compatibility with bidirectional LSTMs," in *Proc. 25th ACM Int. Conf. Multimedia*, Mountain View, CA, USA, Oct. 2017, pp. 1078–1086.

[13] M. Zhao, Y. Liu, X. Li, Z. Zhang, and Y. Zhang, "An end-to-end framework for clothing collocation based on semantic feature fusion," *IEEE MultimediaMag.*, vol. 27, no. 4, pp. 122–132, Oct. 2020.

[14] Y. Ding, P. Y. Mok, Y. Ma, and Y. Bin, "Personalized fashion outfit generation with user coordination preference learning," *Inf. Process. Manage.*, vol. 60, no. 5, Sep. 2023, Art. no. 103434.

[15] W. Guan, X. Song, H. Zhang, M. Liu, C.-H. Yeh, and X. Chang, "Bi-directional heterogeneous graph hashing towards efficient outfit recommendation," in *Proc. 30th ACM Int. Conf. Multimedia*, Oct. 2022, pp. 268–276.

[16] Y. Zhu, Q. Lin, H. Lu, K. Shi, D. Liu, J. Chambua, S. Wan, and Z. Niu, "Recommending learning objects through attentive heterogeneous graph convolution and operation-aware neural network," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 4, pp. 4178–4189, Apr. 2023.

[17] S. C. Hidayati, C.-C. Hsu, Y.-T. Chang, K.-L. Hua, J. Fu, and W.-H. Cheng, "What dress fits me best? Fashion recommendation on the clothing style for personal body shape," in *Proc. 26th ACM Int. Conf. Multimedia*, Oct. 2018, pp. 438–446.

[18] H. Tan and M. Bansal, "LXMERT: Learning cross-modality encoder representations from transformers," 2019, *arXiv:1908.07490*.

[19] D. Gao, L. Jin, B. Chen, M. Qiu, P. Li, Y. Wei, Y. Hu, and H. Wang, "FashionBERT: Text and image matching with adaptive loss for cross-modal retrieval," in *Proc. 43rd Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, China, Jul. 2020, pp. 2251–2260.

[20] W. Su, X. Zhu, Y. Cao, B. Li, L. Lu, F. Wei, and J. Dai, "VL-BERT: Pre-training of generic visual-linguistic representations," 2019, *arXiv:1908.08530*.

[21] J. Lu, D. Batra, D. Parikh, and S. Lee, "ViLBERT: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks," in *Advances in Neural Information Processing Systems*, vol. 32. Red Hook, NY, USA: Curran Associates, 2019.

[22] M. Zhuge, D. Gao, D.-P. Fan, L. Jin, B. Chen, H. Zhou, M. Qiu, and L. Shao, "Kaleido-BERT: Vision-language pre-training on fashion domain," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 12642–12652.

[23] G.-P. Ji, M. Zhuge, D. Gao, D.-P. Fan, C. Sakaridis, and L. V. Gool, "Masked vision-language transformer in fashion," *Mach. Intell. Res.*, vol. 20, no. 3, pp. 421–434, Jun. 2023.

[24] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, "Learning transferable visual models from natural language supervision," in *Proc. Int. Conf. Mach. Learn.*, Feb. 2021, pp. 8748–8763.

[25] Y. Du, Z. Liu, J. Li, and W. X. Zhao, "A survey of vision-language pre-trained models," 2022, *arXiv:2202.10936*.

[26] A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, and M. Chen, "Hierarchical text-conditional image generation with CLIP latents," 2022, *arXiv:2204.06125*.

[27] A. Agostinelli, T. I. Denk, Z. Borsos, J. Engel, M. Verzetti, A. Caillon, Q. Huang, A. Jansen, A. Roberts, M. Tagliasacchi, M. Sharifi, N. Zeghidour, and C. Frank, "MusicLM: Generating music from text," 2023, *arXiv:2301.11325*.

[28] W. Hong, M. Ding, W. Zheng, X. Liu, and J. Tang, "CogVideo: Large-scale pretraining for text-to-video generation via transformers," 2022, *arXiv:2205.15868*.

[29] A. Jaegle, F. Gimeno, A. Brock, O. Vinyals, A. Zisserman, and J. Carreira, "Perceiver: General perception with iterative attention," in *Proc. 38th Int. Conf. Mach. Learn.*, Jul. 2021, pp. 4651–4664.

[30] A. Jaegle, S. Borgeaud, J.-B. Alayrac, C. Doersch, C. Ionescu, D. Ding, S. Koppula, D. Zoran, A. Brock, E. Shelhamer, O. Hénaff, M. M. Botvinick, A. Zisserman, O. Vinyals, and J. Carreira, "Perceiver IO: A general architecture for structured inputs & outputs," 2021, *arXiv:2107.14795*.

[31] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence embeddings using Siamese BERT-networks," 2019, *arXiv:1908.10084*.

[32] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16×16 words: Transformers for image recognition at scale," 2020, *arXiv:2010.11929*.

[33] G. Cucurull, P. Taslakian, and D. Vazquez, "Context-aware visual compatibility prediction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Long Beach, CA, USA, Jun. 2019, pp. 12609–12618.

[34] N. Hegde, J. D. Hipp, Y. Liu, M. Emmert-Buck, E. Reif, D. Smilkov, M. Terry, C. J. Cai, M. B. Amin, C. H. Mermel, P. Q. Nelson, L. H. Peng, G. S. Corrado, and M. C. Stumpe, "Similar image search for histopathology: SMILY," *NPJ Digit. Med.*, vol. 2, no. 1, p. 56, Jun. 2019.

[35] Z. Tabatabaei, Y. Wang, A. Colomer, J. O. Moll, Z. Zhao, and V. Naranjo, "WWFedCBMIR: World-wide federated content-based medical image retrieval," *Bioengineering*, vol. 10, no. 10, p. 1144, Sep. 2023.

[36] Y. Zheng, Z. Jiang, H. Zhang, F. Xie, Y. Ma, H. Shi, and Y. Zhao, "Size-scalable content-based histopathological image retrieval from database that consists of WSIs," *IEEE J. Biomed. Health Informat.*, vol. 22, no. 4, pp. 1278–1287, Jul. 2018.

[37] X. Wang, Y. Du, S. Yang, J. Zhang, M. Wang, J. Zhang, W. Yang, J. Huang, and X. Han, "RetCCL: Clustering-guided contrastive learning for whole-slide image retrieval," *Med. Image Anal.*, vol. 83, Jan. 2023, Art. no. 102645.

[38] X. Qin, Z. Zhang, C. Huang, M. Dehghan, O. R. Zaiane, and M. Jagersand, "U^2-Net: Going deeper with nested U-structure for salient object detection," *Pattern Recognit.*, vol. 106, Oct. 2020, Art. no. 107404.

XIN WANG is currently pursuing the Ph.D. degree with the College of Textiles, Donghua University, under the supervision of Prof. Yueqi Zhong. His research interests include the recognition and understanding of fashion images, recommendation systems of fashion products, and developing methods to combine both visual and textual information.

YUEQI ZHONG received the Ph.D. degree from Donghua University, in 2001. He joined the Faculty of the College of Textiles, Donghua University, in October 2005. He was a Postdoctoral Researcher at The University of Texas at Austin. He is currently a Professor with the College of Textiles, Donghua University. He is regarded as a Specialist in the area of virtual clothing and virtual human body. He was granted the National Natural Science Foundation of China (NSFC) funding many times to support his research work on digitalizing the physical world in cyberspace. He was also the PI of many projects granted with the Level of Province and Department. His research interests include virtual clothing, online sizing, fit evaluation, and virtual human body toward E-commerce. His patents on solving the problem of "virtual reality toward online dressing" won him the prize of the Shanghai Science and Technology Award, in 2013. In 2014, he was a recipient of the Nationwide Prize for his contribution to the textile and apparel industry.