Conditional Cross Attention Network for Multi-Space Embedding without Entanglement in Only a SINGLE Network

Chull Hwan Song[1] Taebaek Hwang[1] Jooyoung Yoon[1] Shunghyun Choi[1] Yeong Hyeon Gu[2]*
[1]Dealicious Inc. [2]Sejong University

Abstract

Many studies in vision tasks have aimed to create effective embedding spaces for single-label object prediction within an image. However, in reality, most objects possess multiple specific attributes, such as shape, color, and length, with each attribute composed of various classes. To apply models in real-world scenarios, it is essential to be able to distinguish between the granular components of an object. Conventional approaches to embedding multiple specific attributes into a single network often result in entanglement, where fine-grained features of each attribute cannot be identified separately. To address this problem, we propose a Conditional Cross-Attention Network that induces disentangled multi-space embeddings for various specific attributes with only a single backbone. Firstly, we employ a cross-attention mechanism to fuse and switch the information of conditions (specific attributes), and we demonstrate its effectiveness through a diverse visualization example. Secondly, we leverage the vision transformer for the first time to a fine-grained image retrieval task and present a simple yet effective framework compared to existing methods. Unlike previous studies where performance varied depending on the benchmark dataset, our proposed method achieved consistent state-of-the-art performance on the FashionAI, DARN, DeepFashion, and Zappos50K benchmark datasets.

1. Introduction

ImageNet [2] is a representative benchmark dataset to verify the visual feature learning effects of deep learning models in the vision domain. However, each image has only one label, which cannot fully explain the various features of real objects. For example, a car can be identified with various attributes such as category, color, and length, as in Figure 1. As shown in Figure 1 (a), the general method of forming embeddings for objects' various attributes involves constructing neural networks equal to the number of specific attributes, and creating multiple embeddings for vision tasks such as image classification [6, 22, 8] and retrieval [10, 20]. Unlike conventional methods, this study presents a technique that embeds various attributes into a single network. We refer to this technique as multi-space attribute-specific embedding Figure 1 (b).

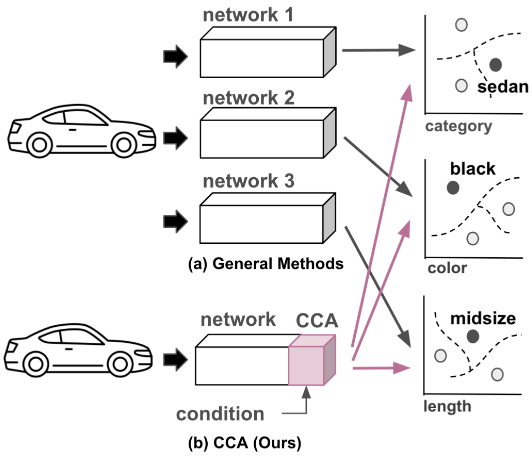

Figure 1: Multiple Networks vs Single Network for Multi-space embedding. CCA means our proposed Conditional Cross Attention Network.

Embedding space aims to encapsulate feature similarities by mapping similar features to close points and dissimilar ones to farther points. However, when the model attempts to learn multiple visual and semantic concepts simultaneously, the embedding space becomes complex, resulting in entanglement; thus, points corresponding to the same semantic concept can be mapped in different regions. Consequently, embedding multiple concepts in an image into a single network is very challenging. Although previous studies attempted to solve this problem using convolutional neural networks (CNNs) [26, 13, 4, 20], they have required intricate frameworks, such as the incorporation of multiple attention modules or stages, in order to identify specific local regions that contain attribute information.

Recently, there has been an increase in research related to ViT [11], which outperforms existing CNN-based mod-

*Corresponding author

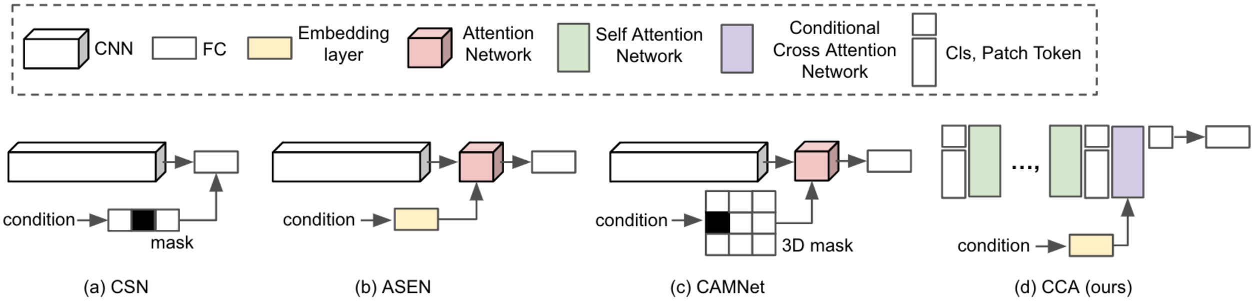

Figure 2: Previous works (CSN, ASEN, CAMNet) *vs*. Ours (CCA)

els in various vision tasks, such as image classification [11], retrieval [21], and detection [1]. In addition, research analyzing how ViT learns representations compared to CNN is underway [17, 15, 14]. Raghu *et al*. [17] demonstrated that the higher layers of ViT are superior in preserving spatial locality information, providing improved spatially discriminative representation than CNN. Some attributes of an object are more easily distinguished when focusing on specific local areas. So, we tailor the last layer of ViT to recognize specific attributes based on their spatial locality, which provides fine-grained information about a particular condition. Figure 2 summarizes the difference between existing CNN-based and proposed ViT-based methods. This study makes the following contributions:

1. Entanglement occurs when embedding an object containing multiple attributes using a single network. The proposed CCA that applies a cross-attention mechanism can solve this problem by adequately fusing and switching between the different condition information (specific attributes) and images.
2. This is the first study to apply ViT to multi-space embedding-based image retrieval tasks. In addition, it is a simple and effective method that can be applied to the ViT architecture with only minor modification. Moreover, it improves memory efficiency by forming multi-space embeddings with only one ViT backbone rather than multiple backbones.
3. Most prior studies showed good performance only on specific datasets. However, the proposed method yields consistently high performance on most datasets and effectively learns interpretable representations. Moreover, the proposed method achieved state-of-the-art (SOTA) performance on all relevant benchmark datasets compared to existing methods.

2. Related Works

Similarity Embedding Triplet Network [24, 18] uses distance calculation to embed images into a space; images in the same category are placed close and those in different categories are far apart. This algorithm has been widely used for diverse subjects such as face recognition and image retrieval. However, as it learns from a single embedding space, it is unsuitable for embedding multiple subjects with multiple categories. Multiple learning models must be created separately according to the number of categories to increase the sophistication level.

Image Retrieval via CNN-based Embedding Image Retrieval is a common task in computer vision, which is finding relevant images based on a query image. Recent works have explored the CNN-based embedding and attention mechanisms to improve image retrieval. Some works leverage attention mechanisms according to the channel-wise [8, 28, 29] and spatial-wise [29] concepts to assign more importance to attended object in the image. Understanding the detailed characteristics of objects is crucial in image retrieval. This is particularly significant in the fashion domain, where even the same type of clothing can have various attributes such as color, material, and length. Therefore, to excel in attribute-based retrieval,, it is required to recognize disentangled representation for each attribute. The nature of this task is suitable for demonstrating the effectiveness of multi-space embedding. Thus, we show the efficacy of CCA through a fashion attribute-specific retrieval task.

CNN based Attributes-Specific Embedding Figure 2 outlines the concepts of existing attribute-specific embedding, similar to our current study. CSN [26] converts the condition into a mask-like representation for multi-space embedding. The mask can be easily applied to the fully connected layer (FC). ASEN [13] joins the attention mechanism with a condition for multi-space embedding. A variation, ASEN++ [4], extended ASEN to 2 stages. These multi-stage techniques are excluded from this study for a fair comparison. M2Fashion [27] adds a classifier to the ASEN base. Unlike CSN, CAMNet [19] was extended to 3D feature maps and applied to the spatial attention mechanism, thus enhancing performance. These studies are CNN-based, not self-attention-based like the present study. The recent ViT [11] has been successfully applied to many vision tasks. However, there has been no technique of multi-space embedding for specific attributes, as described in this study.

3. Methods

Figure 3 presents the proposed CCA architecture, which is mostly similar to that of ViT [11] because it was designed to embed specific attributes through a detailed analysis of the ViT architecture. Hence, CCA is easily applicable under the ViT architecture. Moreover, as described in subsection 4.5, it yields excellent performance. The proposed architectures comprise self-attention and CCA modules. The following sections explain these networks.

3.1. Self Attention Networks

The self-attention module learns a common representation containing the information necessary for multi-space embedding. The self-attention modules are nearly identical to ViT [11]. ViT divides the image into specific patch sizes and converts it into continuous patch tokens (`[PATCH]`). Here, classification tokens (`[CLS]`) [3] are added to the input sequence. As self-attention in ViT is position-independent, position embeddings are added to each patch token for vision applications requiring position information. All tokens of ViT are forwarded through a stacked transformer encoder and used for classification using `[CLS]` of the last layer. The transformer encoder consists of feed-forward (FFN) and multi-headed self-attention (MSA) blocks in a continuous chain. FFN has two multilayer perceptrons; layer normalization (LN) is applied at the beginning of the block, followed by residual shortcuts. The following equation is for the l-th transformer encoder.

$$\begin{aligned} \mathbf{x}_0 &= [\mathbf{x}_{\texttt{[CLS]}}; \mathbf{x}_{\texttt{[PATCH]}}] + \mathbf{x}_{\texttt{[POS]}} \\ \mathbf{x'}_l &= \mathbf{x}_{l-1} + \texttt{MSA}(\texttt{LN}(\mathbf{x}_{l-1})) \\ \mathbf{x}_l &= \mathbf{x'}_l + \texttt{FFN}(\texttt{LN}(\mathbf{x'}_l)) \end{aligned} \tag{1}$$

where $\mathbf{x}_0$ is inital ViT input. $\mathbf{x}_{\texttt{[CLS]}} \in \mathbb{R}^{1\times D}$, $\mathbf{x}_{\texttt{[PATCH]}} \in \mathbb{R}^{N\times D}$ and $\mathbf{x}_{\texttt{[POS]}} \in \mathbb{R}^{(1+N)\times D}$ are the classification, patch, and positional embedding, respectively. The output of the $L-1$ repeated encoder is used as input to the CCA module, as explained in subsection 3.2

3.2. Conditional Cross Attention Network

In this study, the transformer must fuse the concept of attributes and mapped condition information for the network to learn. Drawing inspiration from Vaswani *et al.* [25], we propose CCA to enable learning in line with the transformer's self-attention mechanism. CCA uses a common representation obtained from the self-attention module and cross-attention of the mask according to the given condition to learn nonlinear embeddings that effectively express the semantic similarity based on the condition. Though existing techniques, such as CSN [26] and ASEN [13], have applied condition information to the embedding, these methods are CNN-based rather than transformer-based.

Conditional Token Embedding A network switch based on the condition is required to embed multiple attributes under a single network. In other words, attributes must be learned according to the condition. We propose two conditional token embedding methods, as illustrated in Figure 3.

First, Condition c is converted into a one-hot vector, after which conditional token embedding is performed, similar to that used in multi-modal studies such as DeViSE [5], which learns text and image information having the same meaning using heterogeneous data in the same space, as follows:

$$\mathbf{q_c} = \texttt{FC}(\texttt{onehot}(\texttt{c})) \tag{2}$$

where $\mathbf{q_c} \in \mathbb{R}^{D\times 1}$, c is condition of size K.

Second is the CSN [26] technique, presented in Figure 2 (a). To express K conditions, CSN applies a mask $\in \mathbb{R}^{K\times D}$ to one of the features and uses element-wise multiplication to fuse and embed two CNN features $\in \mathbb{R}^{D}$. This study uses this step only for conditional feature embedding without fusing the features. To this end, we initialize the mask $\in \mathbb{R}^{K\times D}$ for all attributes. This mask can be expressed as a learnable lookup table. The conditional token embedding using the mask is expressed as follows:

$$\mathbf{q_c} = \texttt{FC}(\phi(\mathbf{M}_\theta[c,:])) \tag{3}$$

where ϕ refers to ReLU, the activation function. Accordingly, the dimensions must be the same as the feature to apply self-attention. The result of **FC** in Equation 2 and Equation 3 is embedded while matching the dimension of C.

Finally, the result of both equations must equal the dimensions of the token embedding in subsection 3.1. Therefore, the same vector $\mathbf{q_c} \in \mathbb{R}^{D\times 1}$ is repeated times to expand the result of both equations as follows:

$$Q_c = [\mathbf{q_c}; \mathbf{q_c}; ...; \mathbf{q_c}] \tag{4}$$

Conditional Cross Attention Finally, the transformer architecture must effectively fuse the conditional token embedding vector Q_c, for which we use CCA. The MSA process in Equation 1 uses a self-attention mechanism with the vector query (Q), key (K), and Value (V) as input and is expressed as follows:

$$\texttt{Attention}(Q_i, K_i, V_i) = \texttt{softmax}(\frac{Q_i K_i^\top}{\sqrt{d}})V_i \tag{5}$$

These vectors generated from image i can be expressed using $K_i, Q_i, V_i \in \mathbb{R}^{N\times D}$, consistent with the tokens mentioned above. The inner product of Q and K is calculated, which scales and normalizes with the softmax function to obtain weight N.

In contrast, though CCA is nearly identical to self-attention, Query, Q_c in Equation 4, is generated to have condition information. K_i and V_i, which are the same as above,

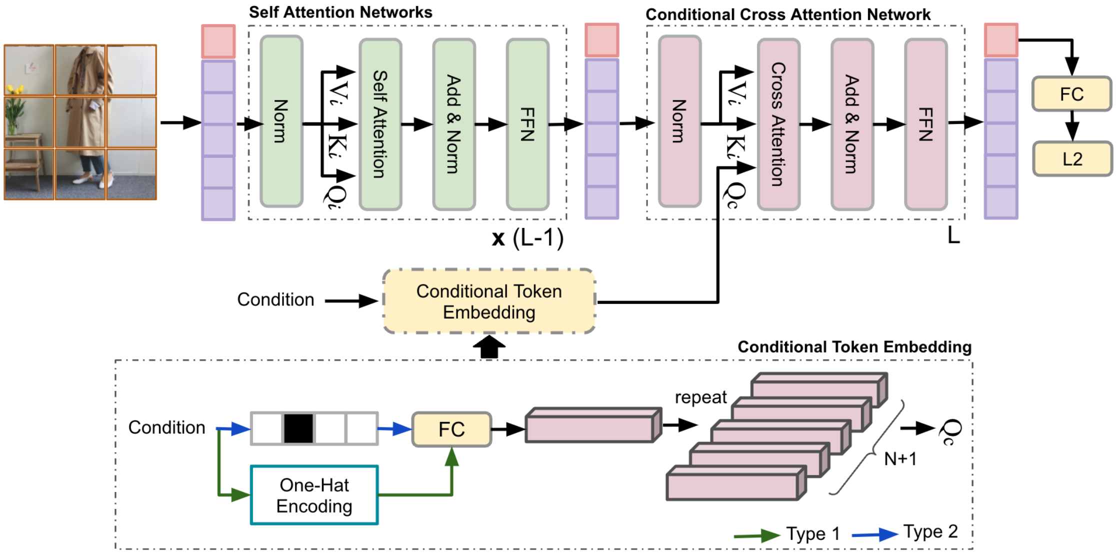

Figure 3: The Architecture of Conditional Cross Attention Network (CCA)

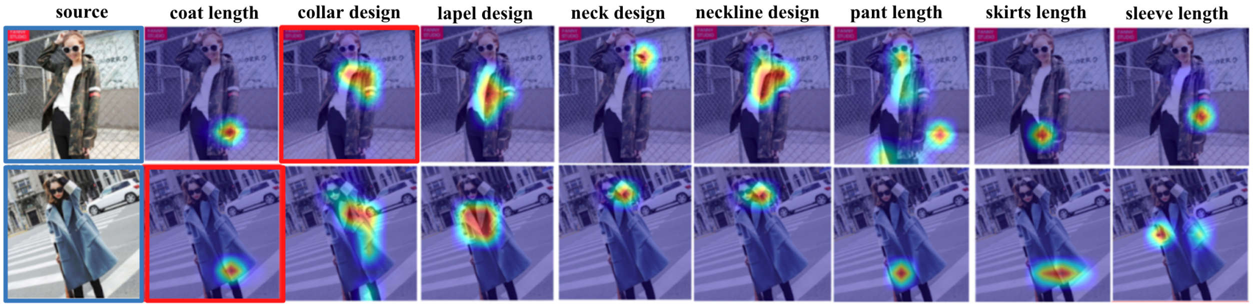

Figure 4: Visualization of attention heat maps for each attribute. Red outlines denote actual annotated attributes in FashionAI.

are input, and the cross-attention mechanism is applied to construct the final CCA as follows:

$$\texttt{Attention}(Q_c, K_i, V_i) = \texttt{softmax}(\frac{Q_c K_i^\top}{\sqrt{d}})V_i \tag{6}$$

The cross-attention mechanism is nearly identical to general self-attention; except for the part of Equation 6, it is the same as Equation 1. The output is the embedding values of `[CLS]` and `[PATCH]`. In our proposed CCA, only `[CLS]` is used for the loss calculation. For the final output, FC and l2 normalization are applied to the embedding feature $x_{\texttt{[CLS]}} \in \mathbb{R}^D$ of `[CLS]` as follows:

$$f^{final} = \texttt{l2}(\texttt{FC}(x_{\texttt{[CLS]}})) \tag{7}$$

Self-attention, explained in subsection 3.1, executes the transformer encoder until step $1 \sim (L-1)$, while CCAN, explained in subsection 3.2, applies only to the final step L. In other words, during inference, as shown in Figure 1, if step $1 \sim (L-1)$ is executed only once and the condition in the final step L is changed and repeated, then several specific features can be obtained under various conditions. Figure 4 shows related experimental results. Eight attributes in the FashionAI dataset are attended in regions matched to each attribute. In addition, step $1 \sim (L-1)$ in the network model can apply the existing ViT-based pre-trained model without modification for learning.

3.3. Triplet Loss with Conditions

We use triplet loss for learning specific attributes, different from the previous general triplet loss in that a conditioned triplet must be constructed. If a label with image I and condition c exists, then the Pair can be denoted as (I, L_c). When expanded to triplets, this is expressed as follows:

$$\mathcal{T} = \{((I^a, L_c^a), (I^+, L_c^+), (I^-, L_c^-)|c)\} \tag{8}$$

where a indicates the anchor, $+$ means that it has the same class in the same condition as the anchor, and $-$ means that it does not have the same class. Using negative samples with the same condition in triplet learning can be interpreted as a hard negative mining strategy. As shown in

DataSets	#Attributes	#Classes	#Images
FashionAI [32]	8	55	180,335
DARN [9]	9	185	195,771
DeepFashion [12]	6	1000	289,222
Zappos50k [31]	4	34	50,025

Table 1: Statistics of the banchmark datasets.

[30], randomly selected negatives are easily distinguished from anchors, enabling the model to learn only coarse features. However, for negative samples with the same condition, the model must distinguish more fine-grained differences. Hence, informative negative samples more suitable for specific-attributes learning are provided. The equation of triplet loss $\mathcal{L}$ is as follows:

$$\begin{aligned}&\mathcal{L}(I^a, I^+, I^-, c) = \\ &\quad \max\{0, \textsc{Dist}(I^a, I^+|c) - \textsc{Dist}(I^a, I^-|c) + m\}\end{aligned} \tag{9}$$

m uses a predefined margin, and $\textsc{Dist}()$ refers to cosine distance. In the Appendix, we present Algorithm 1, which outlines the pseudo-code of our proposed method.

4. Experiments

Table 1 shows the statistics of the datasets, including the number of attributes, classes within the attributes, and the total number of images. The difficulty increases as the number of attributes increases and for higher classes. These results can also be seen in the evaluation results of Table 2, Table 3, and Table 4.

4.1. Metrics

For FashionAI, DARN, and DeepFashion, we used the experimental setting information of ASEN [13] and applied the mean average precision (mAP) metric for evaluation. For Zappos50K, we followed the experimental setting of CSN [26] and applied the triplet prediction metric for evaluation. This metric verifies the efficiency of attribute specific embedding learning for predicting triplet relationships.

4.2. Implementation Details

The experimental environment was implemented using 8 RTX 3090 GPUs. We used Pytorch [16] for all implementations. The backbone network was initialized with pretrained R50+ViT-B/16 [11]. A batch size of 64 and learning rate of 0.0001 was applied for learning. We trained the models up to 200 epochs and selected the trained model that yielded the best results. Triplet loss, described in subsection 3.3, was used with a margin of 0.2.

4.3. Visualization of Multi-Space Embedding and Ranking Results

Entangled *vs*. Disentangled Multi-space Embedding The proposed method enables multi-space embedding for various specific attributes with only one backbone network. When using the general learning method, entanglement in the embedding space inevitably occurs. To solve the entanglement problem and verify whether multi-space embeddings were formed, t-SNE [23] was used to examine the results. The t-SNE visualization results in Figure 5 show whether each attribute class of the FashionAI dataset is properly embedded. The t-SNE visualization results at the center are for the FashionAI dataset with 8 fashion attributes. For the proposed method, excellent embedding results are found for all 8 attributes in the center, and each attribute on the edges. However, training a single model for multiple attributes with the non-conditional method, which is the triplet network in Table 2, Table 3, and Table 4, do not solve the entanglement problem. These findings offer strong evidence that the proposed method achieved multi-space embedding with only one backbone network.

Ours *vs*. Previous Works' Multi-space Embedding Figure 6 compares the embedding results between the proposed and previous (ASEN [13], CAMNet [19]) methods for the FashionAI dataset. The comparison results for 3 of the 8 detailed categories (Neck Design, Sleeve Length, and Coat Length) in FashionAI are shown. Our method yielded better embedding results than ASEN and CAMNet. For example, in the ASEN and CAMNet results, entanglement occurred in the embedding space for the Wrist Length, Long Sleeves, and Extra-long Sleeves classes of Sleeve Length, whereas entanglement is resolved with our proposed method. Figure A3 presents the embedding results for the 8 attributes in the FashionAI dataset.

Ranking Results Figure A2 in Appendix B presents the Top 3 ranking results for the 8 attributes in the FashionAI dataset. The order in the figure is lapel design (notched), neckline design (round), skirt length (floor), pant length (midi), sleeve length (short), neck design (low turtle), coat length (midi), and collar design (peter pan). The features of each attribute are reflected accurately in the ranking. This is also demonstrated in the attention heat map.

4.4. Memory Efficiency

The ViT used in this study has 98M parameters. Individual networks are required to learn attributes with the existing naive method, which necessitates 98M $\times K$ parameters. However, our proposed method can form multi-space embeddings with only one backbone network, thus requiring approximately 98M $\times 1$ parameters. As shown in Figure 2, only the last layer of the ViT model is modified in the proposed CCA, and fewer than 0.1M parameters are added for conditional token embedding. Thus, the proposed method achieves SOTA performance with very few parameters, indicating high efficiency of the algorithm.

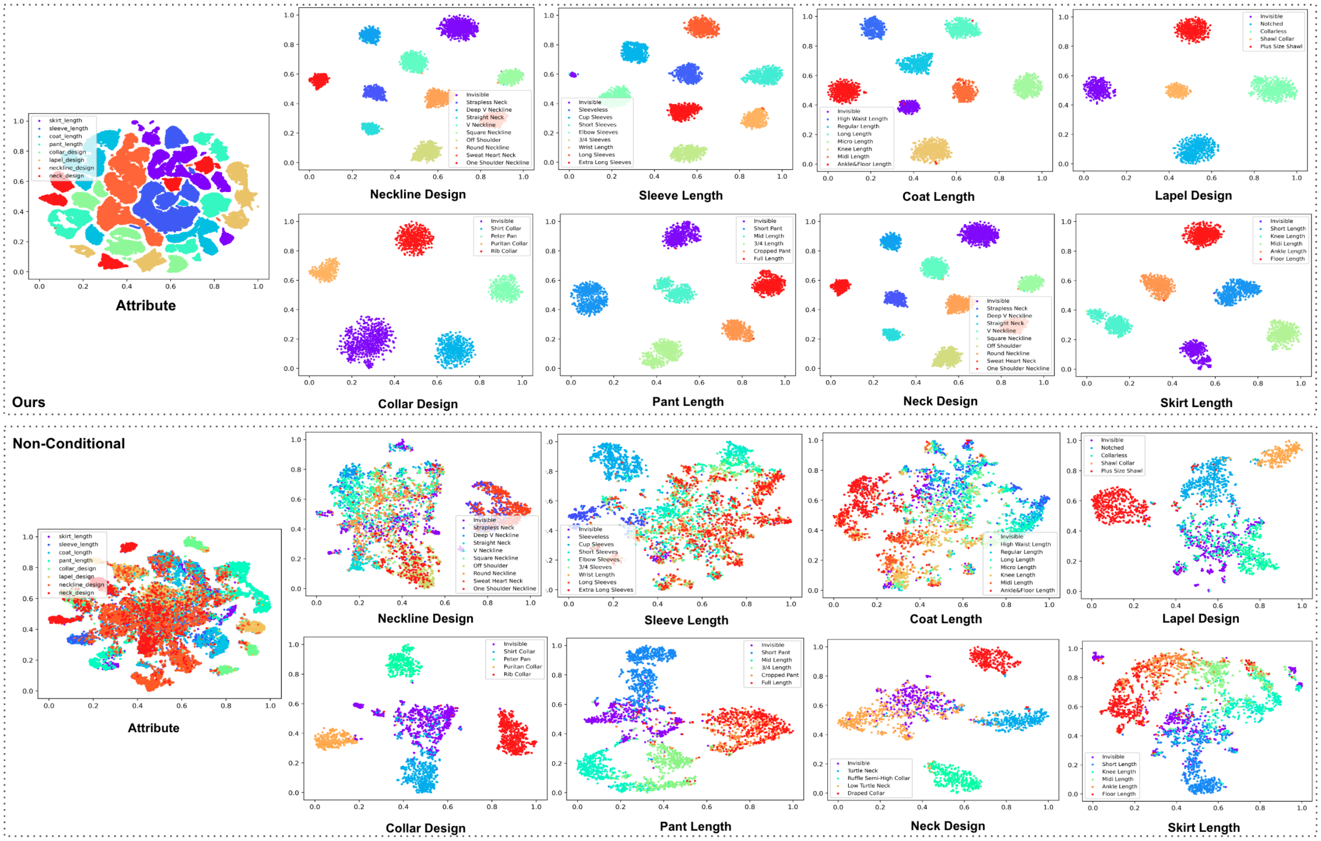

Figure 5: Comparison of multi-space embedding : Conditional (Ours) *vs*. Non-Conditional (Triplet network)

Method	Backbone	mAP	mAP for each attribute							
			skirt length	sleeve length	coat length	pant length	collar design	lapel design	neckline design	neck design
Random baseline [13]	R50	15.79	17.20	12.50	13.35	17.45	22.36	21.63	11.09	21.19
Triplet network [13]	R50	38.52	48.38	28.14	29.82	54.56	62.58	38.31	26.64	40.02
CSN [13]	R50	53.52	61.97	45.06	47.30	62.85	69.83	54.14	46.56	54.47
ASEN [13]	R50	61.02	64.44	54.63	51.27	63.53	70.79	65.36	59.50	58.67
CAMNet [19]	R50	61.97	64.14	56.22	53.05	65.67	72.60	67.74	63.05	61.97
ASEN++ [4]	R50	64.31	66.34	57.53	55.51	68.77	72.94	66.95	66.81	**67.01**
TF-CSN†	ViT	64.86	66.73	59.58	59.94	70.91	71.45	68.17	64.92	62.33
TF-ASEN†	ViT	64.21	65.86	60.11	59.74	70.20	70.80	67.01	64.08	59.48
Ours										
CCA (Type-1)	ViT	66.06	67.20	62.34	60.47	70.29	**75.93**	70.32	65.76	61.04
CCA (Type-2)	ViT	**69.03**	**69.55**	**65.92**	**64.43**	**72.74**	75.39	**71.89**	**70.42**	63.85

Table 2: mAP comparisons of our methods against other studies on FashionAI. Bold: the best results among all methods. Bold black: the best results among the counterparts. TF is Transformer. R50 is ResNet50. † indicates our reproduced results.

4.5. Benchmarking

Table 2, Table 3, and Table 4 present the evaluations for mAP using the metrics in subsection 4.1. Table 5 shows the triplet prediction metric results. In all tables, our method outperforms the SOTA models CSN [26] and ASEN [13].

FashionAI In Table 2, our method achieves SOTA performance for all categories except neck design. Overall, we achieve a +4.72% performance improvement.

DARN In Table 3, the proposed model yields SOTA performance for all items. Averaged across the board, it shows a significant performance improvement of +12.15%.

DeepFashion In Table 4, the proposed model yields SOTA performance for all items. Overall, we achieve a performance improvement of +1.4%. As shown in Table 1, although it consists of only five attributes, these contain many

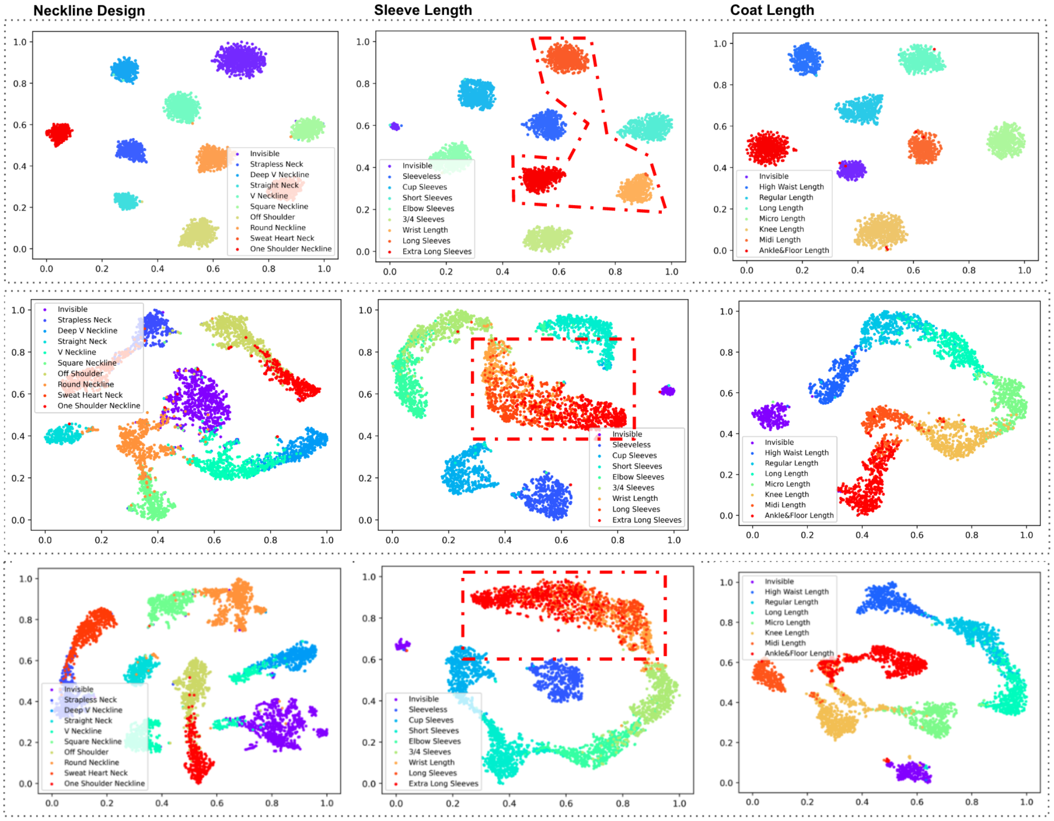

Figure 6: Comparison of multi-space embeddings (Ours vs. ASEN and CAMNet) for FashionAI. The top row corresponds to our method, the middle row to ASEN, and the bottom row to CAMNet. The embeddings are shown for three categories (Neck Design, Sleeve Length, Coat Length) out of eight attributes.

Method	Backbone	mAP	mAP for each attribute								
			clothes category	clothes button	clothes color	clothes length	clothes pattern	clothes shape	collar shape	sleeve length	sleeve shape
Random baseline [13]	R50	32.26	8.49	24.45	12.54	29.90	43.26	39.76	15.22	63.03	55.54
Triplet network [13]	R50	40.14	23.59	38.07	16.83	39.77	49.56	47.00	23.43	68.49	56.48
CSN [13]	R50	50.86	34.10	44.32	47.38	53.68	54.09	56.32	31.82	78.05	58.76
ASEN [13]	R50	53.31	36.69	46.96	51.35	56.47	54.49	60.02	34.18	80.11	60.04
CAMNet [19]†	R50	44.32	25.24	38.02	47.01	45.25	48.35	45.57	23.33	71.69	55.89
M2Fashion [27]	R50	54.29	36.91	48.03	51.14	57.51	56.09	60.77	35.05	81.13	62.23
ASEN++ [4]	R50	55.94	40.15	50.42	53.78	60.38	57.39	59.88	37.65	83.91	60.70
TF-CSN†	ViT	62.85	48.65	60.71	53.27	66.18	63.70	72.75	45.95	88.36	66.35
TF-ASEN†	ViT	33.52	6.20	23.28	31.24	31.37	41.16	39.02	15.57	60.88	54.16
Ours											
CCA (Type-1)	ViT	66.78	51.56	65.55	55.94	72.95	66.97	75.80	51.37	90.08	**71.44**
CCA (Type-2)	ViT	**68.09**	**53.04**	**68.21**	**56.65**	**74.71**	**70.12**	**77.03**	**52.51**	**90.23**	70.99

Table 3: mAP comparisons of our methods against other studies on DARN. † indicates our reproduced results.

more classes than FashionAI and DARN at 1000, resulting in a relatively low mAP value.

Zappos50K Table 5 presents the triplet prediction metric results. Our method achieved SOTA performance, with a +3.61% improvement compared to the previous method. Unlike the aforementioned datasets, the Zappos50K dataset is relatively simple, as indicated by the category composition in Table 1 and the example in Figure A1.

4.6. Ablation Studies

SOTA models applied Transformer The results of the existing CSN [26], ASEN [13] models are obtained with the RestNet50 as the backbone. For a fair comparison, we apply the ViT backbone rather than CNN to these methods and present the experimental results. These models are indicated as TF-CSN and TF-ASEN, respectively. To apply this to CSN and ASEN, first, CSN must accept dimensions of size $\mathbb{R}^D$. Hence, it must be applied in `[CLS]`

Method	Backbone	mAP	mAP for each attribute				
			texture-related	fabric-related	shape-related	part-related	style-related
Random baseline [13]	R50	3.38	6.69	2.69	3.23	2.55	1.97
Triplet network [13]	R50	7.36	13.26	6.28	9.49	4.43	3.33
CSN [13]	R50	8.01	14.09	6.39	11.07	5.13	3.49
ASEN [13]	R50	8.74	15.13	7.11	12.39	5.51	3.56
ASEN++ [4]	R50	9.64	15.60	7.67	14.31	6.60	4.07
TF-CSN†	ViT	10.04	15.27	8.11	14.91	7.40	4.51
TF-ASEN†	ViT	8.53	13.98	6.56	13.39	5.61	3.13
Ours							
CCA (Type-1)	ViT	10.64	16.18	8.38	15.98	7.99	4.78
CCA (Type-2)	ViT	**11.04**	**16.76**	**8.42**	**16.83**	**8.47**	**4.92**

Table 4: mAP comparisons of our methods against other studies on DeepFashion. † indicates our reproduced results.

Method	Prediction Accuracy(%)
Random baseline [13]	50.00
Triplet network [26]	76.28
CSN [26]	89.27
ASEN [13]	90.79
ADDE-C [7]	91.37
TF-CSN†	94.78
TF-ASEN†	94.56
Ours	
CCA (Type-1)	**94.98**
CCA (Type-2)	94.85

Table 5: Performance of triplet prediction on Zappos50k. † indicates our reproduced results.

$\in \mathbb{R}^D$. In contrast, ASEN must accept a CNN feature map of $\in \mathbb{R}^{W \times H \times D}$ dimensions. For ViT, it must be applied in [PATCH] $\in \mathbb{R}^{N \times D}$, which can be applied because N can be reshaped to $W \times H$. One peculiarity is that ASEN outperforms CSN based on CNN but not that based on ViT. Overall, our proposed CCA, with the same transformer base as TF-CSN and TF-ASEN, outperforms both models.

Consistent Performance We found that previous studies yielded different performance results for the datasets. For example, Table 2, CAMNet [19] outperformed ASEN and CSN, whereas, in Table 3 and Table 4, there are no performance results. Similarly, in Table 3, M2Fashion [27] outperformed ASEN and CSN, whereas, in Table 2 and Table 4, there are no results. This suggests that the performance varies with the dataset. Accordingly, we applied the CAMNet study to the DARN dataset to reproduce it. In Table 3, the † symbol indicates our reproduced results. CAMNet model yielded lower performance than ASEN and CSN. Moreover, ASEN outperformed CSN based on CNN in the results of TF-CSN/ASEN when using the transformer. This is attributed to differences in learning according to the characteristics of each dataset. Thus, learning to form embeddings for objects with multiple attributes using a single network is very difficult. In contrast, our proposed CCA consistently yields high performance for all datasets.

Type-1 *vs.* Type-2 These results relate to Equation 2 and Equation 3 in subsection 3.2. Table 2 presents the results for CCA (Type-1) and CCA (Type-2); CCA (Type-2) yielded +2.97% higher performance than CCA (Type-1). In Table 3 and Table 4, CCA (Type-2) showed +1.31% and +0.4% higher performance, respectively. In Table 5, CCA (Type-1) yielded +0.42% higher performance. CCA (Type-2) was slightly higher in the previous three benchmark sets, whereas CCA (Type-1) was slightly higher by 0.13% in this dataset. However, CCA (Type-1) and CCA (Type-2) outperformed all results of the previous studies and TF-CSN and TF-ASEN described above, achieving SOTA performance.

5. Conclusion

This study investigates forming embeddings for an object with multiple attributes using a single network, which is generally difficult in practice. However, the proposed method can extract various specific attribute features using a single backbone network. The proposed network enables multi-space embedding for multiple attributes. Finally, our proposed algorithm achieved SOTA performance in all evaluation metrics for the benchmark datasets.

6. Acknowledgment

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No.2022-0-00739, MLOps Platform for Machine learning pipeline automation)

References

[1] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, 2020. 2

[2] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 1

[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, June 2019. 3

[4] Jianfeng Dong, Zhe Ma, Xiaofeng Mao, Xun Yang, Yuan He, Richang Hong, and Shouling Ji. Fine-grained fashion similarity prediction by attribute-specific embedding learning. In *IEEE Transactions on Image Processing*, 2021. 1, 2, 6, 7, 8

[5] Andrea Frome, Greg Corrado, Jonathon Shlens, Samy Bengio, Jeffrey Dean, Marc'Aurelio Ranzato, and Tomas Mikolov. Devise: A deep visual-semantic embedding model. In *NIPS*, 2013. 3

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *CVPR*, June 2016. 1

[7] Yuxin Hou, Eleonora Vig, Michael Donoser, and Loris Bazzani. Learning attribute-driven disentangled representations for interactive fashion retrieval. In *ICCV*, 2021. 8

[8] Jie Hu, Li Shen, Gang Sun, and Samuel Albanie. Squeeze-and-Excitation Networks. In *TPAMI*, 2017. 1, 2

[9] Junshi Huang, Rogerio Feris, Qiang Chen, and Shuicheng Yan. Cross-Domain Image Retrieval with a Dual Attribute-Aware Ranking Network. In *ICCV*, 2015. 5

[10] Yannis Kalantidis, Clayton Mellina, and Simon Osindero. Cross-Dimensional Weighting for Aggregated Deep Convolutional Features. In *ECCV*, 2016. 1

[11] Alexander Kolesnikov, Alexey Dosovitskiy, Dirk Weissenborn, Georg Heigold, Jakob Uszkoreit, Lucas Beyer, Matthias Minderer, Mostafa Dehghani, Neil Houlsby, Sylvain Gelly, Thomas Unterthiner, and Xiaohua Zhai. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021. 1, 2, 3, 5

[12] Ziwei Liu, Ping Luo, Shi Qiu, Xiaogang Wang, and Xiaoou Tang. DeepFashion: Powering Robust Clothes Recognition and Retrieval with Rich Annotations. In *CVPR*, 2016. 5

[13] 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 *Thirty-fourth AAAI Conference on Artificial Intelligence*, 2020. 1, 2, 3, 5, 6, 7, 8

[14] Muhammad Muzammal Naseer, Kanchana Ranasinghe, Salman H Khan, Munawar Hayat, Fahad Shahbaz Khan, and Ming-Hsuan Yang. Intriguing properties of vision transformers. In *NeurIPS*, 2021. 2

[15] Namuk Park and Songkuk Kim. How do vision transformers work? In *ICLR*, 2022. 2

[16] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, and Soumith Chintala. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *NeurIPS*, 2019. 5

[17] Maithra Raghu, Thomas Unterthiner, Simon Kornblith, Chiyuan Zhang, and Alexey Dosovitskiy. Do vision transformers see like convolutional neural networks? 2021. 2

[18] Florian Schroff, Dmitry Kalenichenko, and James Philbin. FaceNet: A Unified Embedding for Face Recognition and Clustering. In *CVPR*, 2015. 2

[19] Chull Hwan Song and Hye Joo Han. Convolutional attribute mask with two-step attention for fashion image retrieval. In *26th International Conference on Pattern Recognition (ICPR), IEEE*, 2022. 2, 5, 6, 7, 8

[20] Chull Hwan Song, Hye Joo Han, and Yannis Avrithis. All the attention you need: Global-local, spatial-channel attention for image retrieval. In *WACV*, 2022. 1

[21] Chull Hwan Song, Jooyoung Yoon, Shunghyun Choi, and Yannis Avrithis. Boosting vision transformers for image retrieval. In *WACV*, 2023. 2

[22] Mingxing Tan and Quoc Le. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, pages 6105–6114, Long Beach, California, USA, June 2019. PMLR. 1

[23] Laurens van der Maaten and Geoffrey Hinton. Visualizing Data using t-SNE. In *Journal of Machine Learning Research*, 2008. 5

[24] Laurens van der Maaten and Kilian Weinberger. Stochastic triplet embedding. In *MLSP*, 2012. 2

[25] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, ukasz Kaiser, and Illia Polosukhin. Attention is All you Need. In I Guyon, U V Luxburg, S Bengio, H Wallach, R Fergus, S Vishwanathan, and R Garnett, editors, *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2017. 3

[26] Andreas Veit, Serge Belongie, and Theofanis Karaletsos. Conditional Similarity Networks. In *CVPR*, 2017. 1, 2, 3, 5, 6, 7, 8

[27] Yongquan Wan, Cairong Yan, Bofeng Zhang, and Guobing Zou. Learning image representation via attribute-aware attention networks for fashion classification. In *MultiMedia Modeling: 28th International Conference, MMM 2022, Phu Quoc, Vietnam, June 6–10, 2022, Proceedings, Part I*, 2022. 2, 7, 8

[28] Qilong Wang, Banggu Wu, Pengfei Zhu, Peihua Li, Wangmeng Zuo, and Qinghua Hu. ECA-Net: Efficient Channel Attention for Deep Convolutional Neural Networks. In *CVPR*, 2020. 2

[29] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. CBAM: Convolutional Block Attention Module. In *ECCV*, 2018. 2

[30] Hong Xuan, Abby Stylianou, Xiaotong Liu, and Robert Pless. Hard negative examples are hard, but useful. In *ECCV*, 2020. 5

[31] Aron Yu and Kristen Grauman. Fine-grained visual comparisons with local learning. In *CVPR*, 2014. 5

[32] Xingxing Zou, Xiangheng Kong, W. Wong, Congde Wang, Yuguang Liu, and Yuanpeng Cao. FashionAI: A Hierarchical Dataset for Fashion Understanding. In *CVPRW*, pages 296–304, 2019. 5