

Supplementary document: Deviation-based multiple coefficient item mixer for heterogeneous set-to-set matching

Hirotaka Hachiya

HHACHIYA@WAKAYAMA-U.AC.JP

Graduate School of Systems Engineering, Wakayama University/Center for AIP, RIKEN

Yukito Kajishiro

KAJISHIRO.YUKITO@G.WAKAYAMA-U.JP

Graduate School of Systems Engineering, Wakayama University

Editors: Hung-yi Lee and Tongliang Liu

1. Details of experimental evaluation

1.1. Comparison methods

Table 1: Comparative methods with different operations for each component in the network architecture (Fig. 3).

methods	encoder	backbone decoder	aggregation	set similarity measure
CSeFT (Saito et al., 2020)	selfAtt(\bar{X}, \bar{X}) (Eq. 1)	crossAtt(\bar{X}, \bar{Y}) (Eq. 1)	—	CSS (Saito et al., 2020)
pivot-attention (Hachiya and Saito, 2024)	bi-PMA(X_s) (Eq. 2)	pivot-cross(X_s, Y_s) (Eq. 2)	$\widehat{X}_s''[0, :]$ $\widehat{Y}_s''[0, :]$	dot product (Eq. 20)
poolformer + setRepVec	avgPool(X_s) (Eq. 3)	avgPool([$X_s; Y_s$]) (Eq. 3)	$\widehat{X}_s''[0, :]$ $\widehat{Y}_s''[0, :]$	dot product (Eq. 20)
poolformer (Yu et al., 2022)	avgPool(X) (Eq. 3)	avgPool([$X; Y$]) (Eq. 3)	avgPool(\widehat{X}'') $\widehat{X}_s''[0, :]$	dot product (Eq. 20)
Janossy pooling (Murphy et al., 2019)	janossyMixer(X) (Eq. 4)	janossyCrossMixer([$X; Y$]) (Eq. 4)	avgPool(\widehat{X}'') $\widehat{X}_s''[0, :]$	dot product (Eq. 20)
proposed (Sec. 3)	self-mixer(X_s) (Eq. 18)	cross-mixer(Z_s) (Eq. 19)	$\widehat{X}_s''[0, :]$ $\widehat{Y}_s''[0, :]$	dot product (Eq. 20)

We compared the performance of the proposed methods with several existing methods. Table 1 depicts the type of process at each component: backbone (encoder and decoder) and head (aggregation and set-similarity) in the network architecture (in Fig. 3 in the main manuscript).

- proposed (Sec. 3 in the main manuscript): As shown in Fig. 3 (main manuscript), the proposed architecture uses the self-mixer (Eq. 18 in the main manuscript) and cross-mixer (Eq. 19 in the main manuscript) modules in the encoder and decoder, respectively. The final set similarity score is computed as the dot product between the set-rep vectors, i.e., $\widehat{s}_X'' \widehat{s}_Y''^\top$.

The detailed settings are as follows. The number of hidden nodes in each MLP (i.e., MLP_{ch_1} , MLP_{ch_2} , MLP_{coef} , and MLP_{head}) is set to $D_h = D/2$, and GELU is used as the activation function for all hidden layers. The output layer of MLP_{coef}

employs a $\tanh(\cdot)$ activation function to constrain the generated coefficients within the range $[-1, 1]$. Additionally, the dimensionality of each head-wise item vector is set to $D' = \frac{D}{N_{\text{head}}}$, and the number of coefficients is set at $N_{\text{coef}} = 8$ in order to match the number of parameters with the attention mechanism, as described in Sec. 2.1.

- Cross-Set Feature Transformation (CSeFT) (Saito et al., 2020): standard attention-based set-to-set matching method where self- and cross-attention (Eq. 7 in the main manuscript) are used in the backbone for item vector transformation as follows:

$$\begin{aligned} \text{Encoder: } & \widehat{X}' = X + \text{selfAtt}(\text{snorm}(X)), \quad \widehat{X}'' = \widehat{X}' + \text{FC}(\text{norm}(\widehat{X}')), \\ \text{Decoder: } & \widehat{X}' = X + \text{crossAtt}(\text{cnorm}(X, Y), \text{cnorm}(Y, X)), \\ & \widehat{X}'' = \widehat{X}' + \text{FC}(\text{norm}(\widehat{X}')), \end{aligned} \quad (1)$$

where $\text{FC}(\cdot)$ indicates item-wise (channel-direction) fully connected network. As a set-similarity measure in the head, cross-similarity score (CSS) where the average dot product of all possible item pairs between sets \widehat{X}'' and \widehat{Y}'' is computed—the number of heads in the CSS is set at the same value as the number of heads of the attention mechanisms in the backbone, i.e., N_{head} . As for the implementation, we used the code available on GitHub (Hachiya, 2024). Following the implementation of multi-head attention (Vaswani et al., 2017), the dimension of linearly projected item vectors is set as $D' = \frac{D}{N_{\text{head}}}$.

- Pivot-Attention (Hachiya and Saito (2024), in Sec. 2.3 in the main manuscript): A state-of-the-art set-to-set matching method based on the attention mechanism. In this method, bi-PMA (Eq. 9 in the main manuscript) and pivot-attention (Eq. 10 in the main manuscript) are used in the encoder and decoder, respectively, as follows:

$$\begin{aligned} \text{Encoder: } & \widehat{X}'_s = X_s + \text{bi-PMA}(\text{snorm}(X_s)), \quad \widehat{X}''_s = \widehat{X}'_s + \text{FC}(\text{norm}(\widehat{X}'_s)), \\ \text{Decoder: } & \widehat{X}'_s = X_s + \text{pivot-cross}(\text{cnorm}(X_s, Y_s), \text{cnorm}(Y_s, X_s)), \\ & \widehat{X}''_s = \widehat{X}'_s + \text{FC}(\text{norm}(\widehat{X}'_s)). \end{aligned} \quad (2)$$

As the set similarity metric, the dot product between the final set-rep vectors, i.e., $\widehat{s}_X'' \widehat{s}_Y''^\top$, is used. As for the implementation, we used the official code available on GitHub (Hachiya, 2024). Similarly, with CSeFT, the dimension of linearly projected item vectors is set as $D' = \frac{D}{N_{\text{head}}}$.

- poolFormer + set-rep vector (proposed method w/o self-/cross-mixer): To evaluate the effectiveness of the proposed item-specific mixer modules, e.g., self-mixer and cross-mixer (Eq. 17 in the main manuscript), we constructed a baseline method where the mixers are replaced with global average pooling—a variant of poolFormer (Yu et al., 2022) as follows:

$$\begin{aligned} \text{Encoder: } & \widehat{X}'_s = X_s + \text{avgPool}(\text{snorm}(X_s)), \quad \widehat{X}''_s = \widehat{X}'_s + \text{FC}(\text{norm}(\widehat{X}'_s)) \\ \text{Decoder: } & [\widehat{X}'_s; \widehat{Y}'_s] = [X_s, Y_s] + \text{avgPool}(\text{cnorm}(X_s, Y_s)), \\ & \widehat{X}''_s = \widehat{X}'_s + \text{FC}(\text{norm}(X'_s)), \quad \widehat{Y}''_s = \widehat{Y}'_s + \text{FC}(\text{norm}(\widehat{Y}'_s)), \end{aligned} \quad (3)$$

where $\text{avgPool}(\cdot)$ is the operation of averaging in the direction of items. The final set similarity is computed as the dot product between the transformed set-rep vectors, i.e., $\widehat{\mathbf{s}}_X'' \widehat{\mathbf{s}}_Y''^\top$.

- poolFormer (proposed method w/o self-/cross-mixer and set-rep vector \mathbf{s}): The set-rep vector \mathbf{s} is further removed from Eq. 3— X_s and Y_s are replaced with X and Y , respectively. In the head, following the concept of poolFormer, the global average pooling is used to aggregate the items in each set to a single vector, and the dot product is used as the set similarity measure.
- Janossy pooling (Murphy et al., 2019): To compare with a method with a different strategy for permutation invariance than attention or global average pooling, we combined Janossy Pooling (Murphy et al., 2019) with MLP-Mixer (Tolstikhin et al., 2021). Janossy pooling ensures permutation invariance by explicitly enumerating all permutations of the input set and applying a function over each of them. Specifically, we generate the set of all r -permutations of the input set X , denoted by $P_r(X)$, and apply an channel-wise MLP (item mixer) $\text{MLP}_{\text{item}}(\cdot) \in \mathbb{R}^{1 \times D}$ to each permutation. The outputs are then averaged to aggregate information across all possible permutations as follows:

$$\begin{aligned} \text{Encoder: } \widehat{X}' &= X + \frac{1}{|P_r(\text{snorm}(X))|} \sum_{Z \in P_r(\text{snorm}(X))} \text{MLP}_{\text{item}}(Z), \\ \widehat{X}'' &= \widehat{X}' + \text{FC}(\text{norm}(\widehat{X}')), \\ \text{Decoder: } [\widehat{X}'; \widehat{Y}'] &= [X; Y] + \frac{1}{|P_r(\text{cnorm}(X, Y))|} \sum_{Z \in P_r(\text{cnorm}(X, Y))} \text{MLP}_{\text{item}}(Z), \\ \widehat{X}'' &= \widehat{X}' + \text{FC}(\text{norm}(\widehat{X}')), \quad \widehat{Y}'' = \widehat{Y}' + \text{FC}(\text{norm}(\widehat{Y}')), \end{aligned} \quad (4)$$

where $\text{FC}(\cdot)$ corresponds to a channel mixer. By choosing a smaller r such that $r < N_X$, we can reduce the number of possible permutations and thereby reduce the computational cost, although there are trade-off between representational power and computational cost—we set $r = 2$. Even in the head, Janossy pooling is used to aggregate the items in each set to a single vector, and the dot product is used as the set similarity measure.

2. Details of furniture-coordination dataset

The DeepFurniture dataset (Liu et al., 2019) consists of approximately 24,000 interior design coordinations, each formed by combining over 20,000 unique furniture item images with different types, e.g., cabinet, table, chair, and sofa. These coordinations represent compatible combinations curated from COOHOM, a widely used online interior design platform.

To construct the set matching dataset, we first removed duplicate items within each coordination and then filtered coordinations to include only those containing between 4 and 16 furniture item images. As a result, the number of coordinations was reduced from 24,182 to 13,385. We note that the upper bound of 16 items was chosen because coordinations with

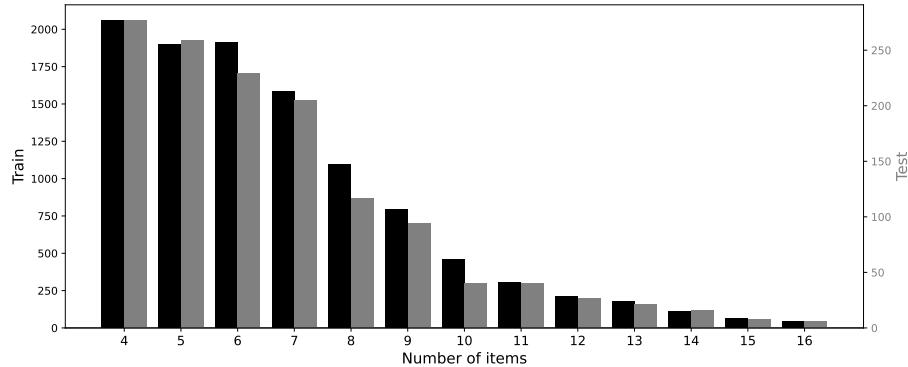


Figure 1: Distribution of the numbers of items N_X/N_Y in coordinations in training and test data of Furniture-coordination dataset.

Table 2: Examples of item images in set \mathcal{X} and \mathcal{Y} with different numbers of items N_X/N_Y .

N_X/N_Y	set \mathcal{X}				set \mathcal{Y}			
	2	4	8	16	2	4	8	16
2								
4								
8								
16								

more than 16 items are rare. The lower bound of 4 items was adopted to ensure that each coordination can be split into two subsets, X and Y , each containing at least two items. We used the output of the fc7 layer of the pretrained VGG16 ([Simonyan and Zisserman, 2015](#)) to convert each furniture image into a 4,096 dimensional vector. We randomly divided the full set of 13,385 coordinations into training, validation, and test splits with a ratio of 8:1:1. Fig. 1 shows the histogram of item counts in sets from the training and test splits, indicating a tendency that sets with fewer items occur more frequently.

Examples of positive pairs with varying numbers of items (N_X/N_Y) are shown in Fig. 2 (main manuscript). Since the types of items in sets X and Y are different, the matching problem is heterogeneous.

For evaluation, we used 1,339 pre-split test sets. Query-gallery pairs were constructed in the same way as Fashion-outfit dataset. The detailed settings of the dataset are summarized in Table 1 (main manuscript).

2.1. Analysis on computational complexity and parameters

The complexity of the attention mechanism for the sets Q , K , and V with the numbers of items N and heads N_{head} , and the dimensions of the item vector D and linearly projected item vector D' at each layer is as follows:

$$\mathcal{O}(4NN_{\text{head}}DD' + 2N^2N_{\text{head}}D'), \quad (5)$$

where the first term corresponds to linear projections using W_Q^h , W_K^h , W_V^h , and W_{head} (Eqs. 5 and 6 in the main manuscript), and the second term corresponds to computing coefficients $\{A^h\}_{h=1}^{N_{\text{head}}}$ and item-specific aggregation of the set V (Eq. 4 in the main manuscript).

Here, the number of items N is usually small in the set-to-set matching tasks, e.g., 2 to 8 in the fashion-outfit and furniture-coordination matching, and is significantly smaller than the dimension of the item vector D , i.e., 64 to 512 dimensions; therefore, $D \gg N$ and the complexity of the attention Eq. 5 is expressed using only the first term as follows:

$$C_{\text{att}} = \mathcal{O}(4ND^2), \quad (6)$$

where $D' = \frac{D}{N_{\text{head}}}$ by following the implementation of multi-head attention (Vaswani et al., 2017).

Meanwhile, the complexity of the DeviMix is a similar structure to the attention mechanism with additional components. The complexity of the DeviMix with the numbers of item mixing coefficients N_{coef} and hidden nodes D_h in the MLP, at each layer is as follows:

$$\begin{aligned} &\mathcal{O}\left(2NN_{\text{head}}DD' + N^2N_{\text{head}}D' + N^2N_{\text{head}}(D'D_h + D'N_{\text{coef}}) \right. \\ &\quad \left. + 2NDD_h + N^2N_{\text{head}}N_{\text{coef}}D + NN_{\text{head}}N_{\text{coef}}D\right), \end{aligned} \quad (7)$$

where the first and second terms correspond to linear projections using W_Q^h and W_K^h , and computing cross-deviation among all item pairs, respectively. The third and fourth terms correspond to MLP_{ch_1} and MLP_{ch_2} . Finally, the fifth and sixth terms correspond to computing item-specific aggregation of the set V and the multiple coefficients and head integration MLP_{head} .

Similarly with attention, with $D \gg N$ and $D' = \frac{D}{N_{\text{head}}}$, the complexity of the DeviMix Eq. 7 is expressed using only the first, third and fourth terms, as follows:

$$C_{\text{mix}} = \mathcal{O}\left((2 + 2\gamma + \gamma N)ND^2\right) \quad (8)$$

where $\gamma \in (0, 1]$ is the item to hidden dimension parameter, i.e., $D_h = \gamma D$. In the experiment, we set $\gamma = 1/2$.

The difference between DeviMix and the attention mechanism can be expressed as $\frac{C_{\text{mix}}}{C_{\text{att}}} = \frac{2+2\gamma+\gamma N}{4}$, indicating that the overall cost does not significantly increase.

Furthermore, while the attention mechanism contains trainable parameters only in the linear projection weights W_Q^h , W_K^h , and W_V^h , totaling $4N_{\text{head}}DD' = 4D^2$ parameters, the proposed DeviMix involves additional parameters due to the use of multiple MLPs. The total number of parameters in DeviMix is:

$$\begin{aligned} &2N_{\text{head}}DD' + 2DD_h + D_h(D' + N_{\text{coef}}) + D_h(N_{\text{coef}}N_{\text{head}} + 1) \\ &= 2D^2 + 2\gamma D^2 + \gamma D\left(\frac{D}{N_{\text{head}}} + N_{\text{coef}}\right) + \gamma D(N_{\text{coef}}N_{\text{head}} + 1) \end{aligned} \quad (9)$$

Table 3: Number of parameters and processing time (ms)

model	(L, N_{head})	Parameters	Time (ms)
CSeFT	(1, 8)	65,544	2.03
	(3, 8)	131,080	5.78
pivot-attention	(1, 8)	32,768	1.95
	(3, 8)	98,304	5.25
poolFormer + set-rep vector	(1, -)	0	1.31
	(3, -)	0	3.19
poolFormer	(1, -)	0	1.26
	(3, -)	0	3.14
Janossy pooling	(1, -)	481	3.58
	(3, -)	1,377	9.10
proposed (DeviMix)	(1, 8)	30,098	4.00
	(3, 8)	90,294	13.06

where the first term corresponds to the linear projections for Q and K , the second term arises from MLP_{ch_2} used to transform the value set V , the third term represents MLP_{ch_1} which generates multiple coefficients from the deviation vectors, and the fourth term comes from MLP_{head} , which integrates outputs across coefficient and head dimensions.

Although DevMix includes more parameter components, the total increase in parameter count remains moderate in practice, as the item vector dimension D is typically much larger than the number of coefficients N_{coef} and the number of heads N_{head} . For example, when $L = 3$, $N_{\text{head}} = 3$, $D = 64$, $N_{\text{coef}} = 8$, $D' = \frac{D}{N_{\text{head}}} = 8$, $D_h = \gamma D = \frac{1}{2} \times 64 = 32$, the number of parameters in the attention mechanism is $4 \times 64^2 = 16,384$. Meanwhile, the number of parameters in the DeviMix is $2 \times 64^2 + 2 \times \frac{1}{2} \times 64^2 + \frac{1}{2} \times 64(\frac{64}{8} + 8) + \frac{1}{2} \times 64(8 \times 8 + 1) = 14,880$.

Table 3 shows the number of parameters and inference time for each model used in the experiments of Table 3 (main manuscript). Each model has a structure in which the self/cross-mixer and head modules in Fig. 3 (main manuscript) are replaced with the model-specific operation described in Table 1. Therefore, the reported number of parameters corresponds to those used in the model-specific operation. For example, in the case of the proposed method, it represents the total number of parameters in the self/cross-mixer and dot product modules, while for CSeFT, it represents the total number of parameters in the self/cr-attention and CSS modules. The inference time indicates the average processing time (in milliseconds) required to compute the between-set similarity \hat{X} for one query set and five gallery sets, measured over all 1,339 query sets in the test dataset using the entire architecture shown in Fig. 3 (main manuscript).

Similar to the theoretical analysis described above, Table 3 shows that the proposed method has fewer parameters than Pivot-Attention and CSeFT. Regarding the computational cost, from Eqs. 5 and 7, when $N = 8$ and $\gamma = \frac{1}{2}$, the theoretical ratio between the DevMix and the attention mechanisms is given as $\frac{C_{\text{mix}}}{C_{\text{att}}} = \frac{2+2\gamma+\gamma N}{4} = \frac{7}{4}$. From the table, the ratio of processing times between the proposed method and Pivot-Attention for a single layer is $\frac{4.00}{1.95} \approx \frac{7}{4}$, which is almost identical to the theoretical value. Therefore, the inference time is consistent with the theoretical analysis.

Table 4: Summary of the experimental results evaluating robustness to imbalance in the number of items using Furniture coordination dataset.

Train	Test	model	(L, N_{head})	CMC 1↑	CMC 2↑	CMC 3↑	loss ↓
equal	equal	proposed (DeviMix)	(3, 8)	73.5 (3.6)	91.1 (1.5)	97.6 (0.5)	0.375 (0.016)
	multiple	proposed (DeviMix)	(3, 8)	41.9 (2.0)	58.5 (1.6)	69.9 (2.3)	2.892 (0.300)
random	equal	proposed (DeviMix)	(3, 8)	63.6 (1.0)	84.4 (0.4)	93.9 (0.4)	0.479 (0.008)
	multiple	proposed (DeviMix)	(3, 8)	65.4 (1.4)	86.0 (0.5)	94.7 (0.5)	0.458 (0.003)

2.2. Evaluation of Robustness to Imbalance in the Number of Items

In practical applications such as furniture or fashion coordination recommendation systems, a query corresponds to a set of items already purchased by a user, and the positive set represents a set of items required to complete the coordination. In such cases, there can be an imbalance in the number of items between the query and its corresponding positive set.

In the experiment shown in Table 3 (main manuscript), the query–positive set pairs were created by evenly dividing coordination sets for simplicity. Here, to evaluate the effectiveness of the proposed method under item-number imbalance, we conducted experiments by changing the division ratio of the Furniture-coordination dataset. Specifically, the coordination sets in the test dataset \mathcal{D}_{te} were divided into query and positive sets using five different ratios: 5:5, 2:8, 4:6, 6:4, and 8:2. In each case, both sets contained at least two items, and there was no item number overlap. As a result, the number of test samples N_{te} increased from 1,339 to 3,643 and for each query, we randomly constructed five gallery sets including the corresponding positive set; the number of queries is $3,643 \times 5 = 18,215$ (Table 1, main manuscript).

For model training, two types of data-splitting strategies were used. The first is the equal-division setting, corresponding to the model trained for the experiments in Table 3 (main manuscript). The second is the random-division setting, where coordination sets are randomly split such that both the query and target sets contain at least two items. We evaluated four combinations consisting of the two training strategies (equal and random division) and the two test datasets (equal and diverse division). The experimental results are summarized in Table 4.

The results show that the model trained with equal-division data significantly degrades in accuracy when tested on the diverse-division dataset. In contrast, the model trained with random-division data achieves substantially higher accuracy on the diverse-division test dataset, while maintaining only a slight decrease in accuracy for the equal-division case. These results indicate that the proposed method can achieve stable performance regardless of the imbalance in the number of items between the query and target sets.

2.3. Examples of sets and predicted scores

Tables 5 and 6 show examples of images $\{I_{x_1}, I_{x_2}, \dots\}$ and $\{I_{y_1}, I_{y_2}, \dots\}$ in query \mathcal{X} and candidate sets \mathcal{Y}_k in the gallery \mathcal{G} for the furniture-coordination matching task when the number of candidate sets in the gallery \mathcal{G} is $N_{\mathcal{G}} = 5$ and the numbers of layers and heads are set as $L = 3$ and $N_{\text{head}} = 8$. We note that the scene image in the left bottom is an example of the layout of furniture combining those in the query \mathcal{X} and the positive candidate \mathcal{Y}_1 , and not used in the matching task.

Table 5: Examples of images in sets \mathcal{X} and $\{\mathcal{Y}_1, \mathcal{Y}_2, \dots, \mathcal{Y}_5\} = \mathcal{G}$ on furniture-coordination matching, and predicted scores $\widehat{s_{XY}}$ using the proposed methods, DeviMix, when $L = 3$, $N_{\text{head}} = 8$, and $N_{\text{coef}} = 8$.

query \mathcal{X}				
				
scene image 				
	positive \mathcal{Y}_1 : $\widehat{s_{XY}} = 0.985$			
				
	negative \mathcal{Y}_2 : $\widehat{s_{XY}} = 0.0$			
				
	negative \mathcal{Y}_3 : $\widehat{s_{XY}} = 0.001$			
				
	negative \mathcal{Y}_4 : $\widehat{s_{XY}} = 0.771$			
				
	negative \mathcal{Y}_5 : $\widehat{s_{XY}} = 0.376$			
				

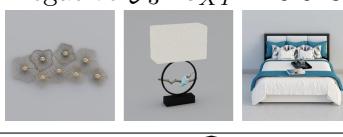
2.4. Ablation study

To evaluate the effectiveness of each component of the proposed DeviMix module, we conducted an ablation study using fashion-outfit and furniture-coordination datasets. Three variants of the proposed method were evaluated as follows:

- **w/o self-/cross-mixer:** To assess the contribution of item-specific mixing coefficients W_i^h in Eq. 13 (main manuscript), we replaced it with a global item mixing coefficients computed by DuMLP-Pin in Sec. 2.4 (main manuscript). Specifically, we replaced self- and cross-mixer in the encoder and decoder (Eqs. 18 and 19 in the main manuscript) with Du-MLP (Eq. 12 in the manuscript) as follows:

$$\begin{aligned}
 \text{Encoder: } & \widehat{X}'_s = X_s + \text{DuMLP}(\text{snorm}(X_s)), \quad \widehat{X}''_s = \widehat{X}'_s + \text{FC}(\text{norm}(\widehat{X}'_s)), \\
 \text{Decoder: } & [\widehat{X}'_s; \widehat{Y}'_s] = [X_s, Y_s] + \text{DuMLP}(\text{cnorm}(X_s, Y_s)), \\
 & \widehat{X}''_s = \widehat{X}'_s + \text{FC}(\text{norm}(\widehat{X}'_s)), \quad \widehat{Y}''_s = \widehat{Y}'_s + \text{FC}(\text{norm}(\widehat{Y}'_s)),
 \end{aligned} \tag{10}$$

Table 6: Examples of images in sets \mathcal{X} and $\{\mathcal{Y}_1, \mathcal{Y}_2, \dots, \mathcal{Y}_5\} = \mathcal{G}$ on furniture-coordination matching, and predicted scores $\widehat{s_{XY}}$ using the proposed methods, DeviMix, when $L = 3$, $N_{\text{head}} = 8$, and $N_{\text{coef}} = 8$.

 	<p>query \mathcal{X}</p> <p>positive $\mathcal{Y}_1 : \widehat{s_{XY}} = 0.965$</p> 
	<p>negative $\mathcal{Y}_2 : \widehat{s_{XY}} = 0.453$</p> 
	<p>negative $\mathcal{Y}_3 : \widehat{s_{XY}} = 0.013$</p> 
	<p>negative $\mathcal{Y}_4 : \widehat{s_{XY}} = 0.035$</p> 
	<p>negative $\mathcal{Y}_5 : \widehat{s_{XY}} = 0.064$</p> 

- **w/o set-req vector:** To assess the impact of the set-req vector \mathbf{s} in the proposed MLP-based set-to-set matching method—it has been shown effective in the attention-based method [Hachiya and Saito \(2024\)](#), we replaced the input X_s with X without \mathbf{s} . Accordingly, the set similarity metric in the head network is computed using a global average pooling followed by a dot product.
- **w/o deviation vectors:** To assess the contribution of cross-deviation vectors in the proposed DeviMix (Eq. 13 in the main manuscript), we replaced it with the scalar values of cross-dot product and cross-Euclidean distances as follows:

$$\text{dot: } A_i^h = \text{MLP}_{\text{ch}_1}(\mathbf{q}_i^h K^h) \in \mathbb{R}^{N_K \times N_{\text{coef}}}, \quad (11)$$

$$\text{Euclid: } A_i^h = \text{MLP}_{\text{ch}_1}\left(\|\mathbf{q}_i^h - K^h\|_2\right) \in \mathbb{R}^{N_K \times N_{\text{coef}}}. \quad (12)$$

The ablation study is summarized in Table 7. The table demonstrates that removing any individual component of the proposed DeviMix method leads to a decrease in the resulting performance. In particular, when the cross-deviation vectors were replaced with a scalar

Table 7: Ablation study evaluating the contributions of each component of the proposed DeviMix method, i.e., item-wise self-/cross-mixer, set-rep vector, and cross-deviation vectors in fashion-outfit and furniture-coordination matching tasks. Each result shows the mean and standard deviation of the top three trials selected from five runs based on CMC1. The best scores for each metric are highlighted in bold.

dataset	model	(L, N_{head})	CMC 1↑	CMC 2↑	CMC 3↑	loss ↓
Fashion	proposed (DeviMix)	(3, 8)	84.5 (0.3)	96.7 (0.4)	99.4 (0.1)	0.292 (0.013)
	w/o self-/cross-mixer	(3, 8)	83.3 (1.9)	96.7 (0.5)	99.2 (0.1)	0.289 (0.014)
	w/o set-rep vector	(3, 8)	81.5 (0.9)	96.0 (0.4)	99.1 (0.1)	0.302 (0.004)
	w/o deviation vectors (dot)	(3, 8)	80.0 (0.5)	95.4 (0.2)	99.0 (0.1)	0.332 (0.023)
	w/o deviation vectors (Euclid)	(3, 8)	82.7 (0.5)	96.3 (0.3)	99.4 (0.1)	0.295 (0.011)
Furniture	proposed (DeviMix)	(3, 8)	73.5 (3.6)	91.1 (1.5)	97.6 (0.5)	0.375 (0.016)
	w/o self-/cross-mixer	(3, 8)	72.7 (1.5)	91.2 (0.4)	97.5 (0.4)	0.384 (0.012)
	w/o set-rep vector	(3, 8)	72.9 (2.0)	90.8 (1.1)	97.1 (1.1)	0.392 (0.014)
	w/o deviation vectors (dot)	(3, 8)	56.8 (7.0)	80.9 (4.1)	92.2 (1.5)	0.490 (0.043)
	w/o deviation vectors (Euclid)	(3, 8)	72.3 (1.4)	90.6 (0.4)	97.1 (0.4)	0.381 (0.009)

cross-dot product value, i.e., using only cosine similarity, the performance dropped significantly. This indicates that the dot product-based similarity fails to capture dimension-wise direction and positional differences between item vectors, making it difficult for the MLP to generate effective item-specific coefficients, as discussed in Secs. 2.3 in the main manuscript.

Overall, the ablation study validates the effectiveness of the proposed DeviMix method, especially cross-deviation vectors combined with MLP-based dynamic coefficient generation for item-specific aggregation in heterogeneous set-to-set matching.

References

- Hirotaka Hachiya. set_rep_vec_asym_attention. https://github.com/hhachiya/set_rep_vec_asym_attention, 2024. Accessed: 2025-05-23.
- Hirotaka Hachiya and Yuki Saito. Set representative vector and its asymmetric attention-based transformation for heterogeneous set-to-set matching. In *Neurocomputing*, volume 578, 2024.
- Bingyuan Liu, Jiantao Zhang, Xiaoting Zhang, Wei Zhang, Chuanhui Yu, and Yuan Zhou. Furnishing your room by what you see: An end-to-end furniture set retrieval framework with rich annotated benchmark dataset. In *Computer Vision and Pattern Recognition (CVPR)*, 2019.
- Ryan L. Murphy, Balasubramaniam Srinivasan, Vinayak Rao, and Bruno Ribeiro. Janossy pooling: Learning deep permutation-invariant functions for variable-size inputs. 2019.
- Yuki Saito, Takuma Nakamura, Hirotaka Hachiya, and Kenji Fukumizu. Exchangeable deep neural networks for set-to-set matching and learning. In *Proceedings of European Conference on Computer Vision (ECCV)*, pages 626–646, 2020.
- Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. 2015.

Ilya Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Andreas Steiner, Jessica Yung, Daniel Keysers, Jakob Uszkoreit, Mario Lucic, and Alexey Dosovitskiy. Mlp-mixer: An all-mlp architecture for vision. 2021.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, volume 30, 2017.

Weihao Yu, Mi Luo, Pan Zhou, Chenyang Si, Yichen Zhou, Xinchao Wang, Jiashi Feng, and Shuicheng Yan. Metaformer is actually what you need for vision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10819–10829. IEEE, 2022.