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

Abstract

Heterogeneous set-to-set matching tasks—such as fashion outfit recommendation—require permutation-invariant and dynamic item-wise transformations to bring compatible sets closer while pushing incompatible ones apart. While attention-based methods satisfy the permutation invariance requirement, they often suffer from convex hull limitations due to their reliance on softmax-based dot-product operations. On the other hand, MLP-based methods like DuMLP-Pin avoid such constraints but tend to lose critical item-wise structure through global aggregation. To address these limitations, we propose DeviMix (Deviation-based multiple coefficient item Mixer), a novel MLP-based architecture that performs item-wise dynamic transformations. Our approach generates multiple item-mixing coefficients by applying MLPs to cross-deviation vectors computed from all possible item pairs in sets. Extensive experiments on fashion outfit and furniture coordination matching tasks demonstrate that DeviMix consistently outperforms attention-based and global pooling-based baselines, validating the effectiveness of our MLP-based item-wise aggregation using cross-deviation for heterogeneous set matching.

Keywords: MLP; permutation-invariance; heterogeneous set-to-set matching

1. Introduction

Matching between image sets has significantly advanced in recent years due to progress in deep learning, enabling a wide range of applications such as video-based face recognition (Yang et al., 2017; Liu et al., 2019b), group re-identification (group Re-ID) (Saito et al., 2020; Xiao et al., 2018; Huang et al., 2021), and fashion outfit recommendation (Hsiao and Grauman, 2018; Vasileva et al., 2018; Saito et al., 2020).

In set-to-set matching, it is crucial that the matching results are permutation invariant with respect to the input items within each set and set-exchangeable between the two sets. To ensure these properties, various permutation-invariant pooling functions have been proposed to aggregate item vectors within a set into a single vector, allowing the matching score to be computed via dot products (Zaheer et al., 2017; Lee et al., 2019; Hachiya and Saito, 2024). Notable examples include sum pooling in DeepSets (Zaheer et al., 2017), max pooling in PointNet (Qi et al., 2017), pooling by multi-head attention (PMA) in Set Transformer (Lee et al., 2019), Janossy Pooling (Murphy et al., 2019), DuMLP-Pin (Fei et al., 2022), and the set-representative vector (Hachiya and Saito, 2024). Several methods directly compute a matching score between two sets of item vectors without collapsing

© 2025 H. Hachiya & Y. Kajishiro.

them into single vectors (Liu et al., 2019b; Saito et al., 2020). These include permutation-invariant feature restructuring (PIFR) (Liu et al., 2019b), CATSET (Sharma et al., 2021) and cross-similarity score (CSS) (Saito et al., 2020).

In addition, self-attention and cross-attention mechanisms (Lee et al., 2019) in the backbone network provide item-specific permutation-invariant transformation by aggregating each item vector using coefficients generated by cross-dot product between all possible item pairs, and they are widely applied in set-to-set matching (Sharma et al., 2021; Saito et al., 2020; Hachiya and Saito, 2024). Tasks such as face recognition and group Re-ID typically involve homogeneous matching, where both sets contain items of the same or similar types. In contrast, fashion coordination recommendation requires **heterogeneous** matching, where compatibility must be identified between sets of different item types. Heterogeneous matching poses greater challenges because it demands more dynamic transformations across a broader feature space to match target items of different types.

It is known that attention-based item vector transformations are performed within the convex hull formed by the linearly projected item vectors. In set-to-set matching scenarios involving a much smaller number of items in comparison with text data, the expressive capacity of standard self- and cross-attention mechanisms becomes inherently constrained. To overcome this limitation, an additional trainable item vector, called **set-rep** (set representative) vector, is added to each set, and asymmetric attention mechanisms called bi-PMA and pivot-cross have been proposed (Hachiya and Saito, 2024). This approach effectively extends the original convex hulls and allows the attention mechanism to encode more discriminative item information, allowing the matching score to be computed via dot products. However, despite this extension, dot-product attention remains constrained by (1) scale-sensitive convex hull limitations, and (2) the loss of explicit directional and magnitude information due to scalar similarity compression.

DuMLP-Pin (Fei et al., 2022) offers an alternative approach for dynamically computing coefficients by applying item-wise multilayer perceptrons (MLPs). However, DuMLP-Pin aggregates all items into a single global representation, inevitably discarding fine-grained item-wise information that is critical for heterogeneous set matching.

To address the limitations of both dot-product-based attention mechanisms and global pooling approaches like DuMLP-Pin, we propose a novel item-wise set aggregation method called **Deviation-based multiple coefficient item Mixer (DeviMix)**. Unlike attention, which computes scalar similarities through dot products, or DuMLP-Pin, which outputs a single global vector, DeviMix dynamically computes item-specific coefficients by applying MLPs to deviation vectors—defined as the difference between each item and the rest of the set. These coefficients capture rich, directional, and context-aware relationships, enabling flexible and discriminative transformations of individual item vectors.

We demonstrate the effectiveness of our proposed method on heterogeneous set-to-set matching tasks, including fashion outfit matching using the Shift15M dataset (Kimura et al., 2021) and furniture coordination matching using the DeepFurniture dataset (Liu et al., 2019a).

2. Set-to-set matching and related works

This section reviews the formulation of set-to-set matching and its related works.

2.1. Formulation

Let us denote $\mathcal{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_{N_X}\}$ and $\mathcal{Y} = \{\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_{N_Y}\}$ as the set of item vectors $\boldsymbol{x}_i$ and $\boldsymbol{y}_i \in \mathbb{R}^{1\times D}$, respectively. For computational convenience, we represent these sets in matrix form as $X \equiv [\boldsymbol{x}_1; \boldsymbol{x}_2; \ldots; \boldsymbol{x}_{N_X}] \in \mathbb{R}^{N_X \times D}$ and $\equiv [\boldsymbol{y}_1; \boldsymbol{y}_2; \ldots; \boldsymbol{y}_{N_Y}] \in \mathbb{R}^{N_Y \times D}$, which stack the item vectors as rows, where ';' indicates the row breaking, D is the dimension of an item vector, and N_X and N_Y are the numbers of items in $\mathcal{X}$ and $\mathcal{Y}$, respectively.

Let $\mathcal{D}_{\mathrm{tr}}$ and $\mathcal{D}_{\mathrm{te}}$ denote training and test data consisting of pairs of compatible sets $\mathcal{X}$ and $\mathcal{Y}$ as follows:

$$\mathcal{D}_{\mathrm{tr}} = \{(\mathcal{X}_n, \mathcal{Y}_n)\}_{n=1}^{N_{\mathrm{tr}}}, \quad \mathcal{D}_{\mathrm{te}} = \{(\mathcal{X}_n, \mathcal{Y}_n)\}_{n=1}^{N_{\mathrm{te}}}, \tag{1}$$

where N_{tr} and N_{te} are the numbers of training and test data.

Given a query set X and a gallery $\mathcal{G} \equiv \{Y^1, Y^2, \ldots, Y^{N_{\mathcal{G}}}\}$, the task of set-to-set matching is to select the most matching set $Y \in \mathcal{G}$ with the maximum score value based on a score function $f(\cdot,\cdot) \in [0,1]$ as follows:

$$\mathrm{matching}(X, \mathcal{G}) = \operatorname*{argmax}_{Y \in \mathcal{G}} f(X, Y), \tag{2}$$

where Y^g is the g-th candidate and $N_{\mathcal{G}}$ is the number of candidate sets in $\mathcal{G}$.

The score function $f(\cdot,\cdot)$ predicts the score $s_{XY} \in [0,1]$, where $s_{XY} = 1$ if X and Y match, and is 0 otherwise. Here, we consider the score function is modeled by the combination of the backbone network $\mathrm{backbone}(\cdot,\cdot)$ for transforming item vectors and the head network $\mathrm{set\text{–}sim}(\cdot,\cdot)$ for computing the between-set similarity function as follows:

$$\widehat{s_{XY}} = f(X, Y) = \sigma\big(\mathrm{set\text{–}sim}(\widehat{X}, \widehat{Y})\big), \quad \widehat{X}, \widehat{Y} = \mathrm{backbone}(X, Y), \tag{3}$$

where $\sigma(\cdot)$ is a sigmoid function. The parameters of both backbone and head networks are tuned using the training data $\mathcal{D}_{\mathrm{tr}}$.

2.2. Attention-based permutation-invariant aggregation

The main approach for transforming vectors for set-to-set matching in the backbone is based on an attention mechanism (Lee et al., 2019; Saito et al., 2020; Sharma et al., 2021; Hachiya and Saito, 2024). In this mechanism, the normalized coefficient $\boldsymbol{a}_i \in \mathbb{R}^{1\times N_K}$ is dynamically obtained based on the similarity between a query item vector $\boldsymbol{q}_i \in \mathbb{R}^{1\times D'}$ and item vectors in a key set $K \in \mathbb{R}^{N_K \times D'}$. Then, $\boldsymbol{q}_i$ is transformed by the aggregation of item vectors in a value set $V \in \mathbb{R}^{N_K \times D'}$ using a linear combination (Vaswani et al., 2017) as follows:

$$\boldsymbol{a}_i = \mathrm{softmax}\Big(\frac{\boldsymbol{q}_i K^\top}{\sqrt{D'}}\Big), \quad \widehat{\boldsymbol{q}}_i = \mathrm{Att}(\boldsymbol{q}_i, K, V) = \boldsymbol{a}_i V, \tag{4}$$

where, since $\mathrm{sum}(\boldsymbol{a}_i) = 1$ and $\boldsymbol{a}_{ij} \geq 0\ \forall j$, $\widehat{\boldsymbol{q}}_i \in \mathbb{R}^{1\times D'}$ is the projection onto the convex hull of item vectors in V. In addition, since the orders of items in the key K and value V sets are the same, the resulting $\widehat{\boldsymbol{q}}_i$ is permutation-invariant over items in the key and value sets.

With trainable weights W_Q^h, W_K^h, and $W_V^h \in \mathbb{R}^{D'\times D'}$ at each head h, the item vectors are linearly projected as follows:

$$\boldsymbol{q}_i^h = \boldsymbol{q}_i W_Q^h, \quad K^h = K W_K^h, \quad V^h = V W_V^h. \tag{5}$$

Then, multiple transformed query items $\{\widehat{\boldsymbol{q}_i^h}\}_{h=1}^{N_{\text{head}}}$ obtained by the attention $\text{Att}(\boldsymbol{q}_i^h, K^h, V^h)$ (Eq. 4) are linearly combined using trainable weights $W_{\text{head}} \in \mathbb{R}^{(D' \times N_{\text{head}}) \times D}$ as follows:

$$\widehat{\boldsymbol{q}_i} = \text{multiAtt}(\boldsymbol{q}_i, K, V) = \left[\widehat{\boldsymbol{q}_i^1}, \widehat{\boldsymbol{q}_i^2}, \cdots, \widehat{\boldsymbol{q}_i^{N_{\text{head}}}}\right] W_{\text{head}}, \tag{6}$$

where N_{head} is the number of heads.

To extract effective transformation of X and Y for set-to-set matching, the backbone for alternatively applying the following self-attention and cross-attention operations has been proposed (Saito et al., 2020; Sharma et al., 2021):

$$\begin{aligned} &\widehat{X} = \text{selfAtt}(X) = \text{multiAtt}(X, X, X), \ \widehat{Y} = \text{selfAtt}(Y) = \text{multiAtt}(Y, Y, Y), \\ &\widehat{X} = \text{crossAtt}(X, Y) = \text{multiAtt}(X, Y, Y), \ \widehat{Y} = \text{crossAtt}(Y, X) = \text{multiAtt}(Y, X, X). \end{aligned} \tag{7}$$

However, this approach has two main limitations. First, since self- and cross-attention output transformed sets $\widehat{X}$ and $\widehat{Y}$, permutation-invariant pooling function is necessary in the head network $f(\cdot, \cdot)$, such as sum pooling (Zaheer et al., 2017), PMA (Lee et al., 2019), CATSET (Sharma et al., 2021), and CSS (Saito et al., 2020). Second, because the number of item vectors in each set is typically small in set-to-set matching tasks, the ability of attention layers to transform these vectors is inherently limited. Moreover, the use of softmax-normalized dot-product attention restricts the output representations to lie within the convex hull of the linearly projected item vectors.

2.3. Set representative vector and asymmetric attention

To address the limitations of the self- and cross-attention mechanism, a trainable vector $\boldsymbol{s}$, called the set representative (set-rep) vector, is introduced into sets as one of the items as follows:

$$X_{\boldsymbol{s}} \equiv [\boldsymbol{s}_X; X], \quad Y_{\boldsymbol{s}} \equiv [\boldsymbol{s}_Y; Y], \tag{8}$$

where $\boldsymbol{s}_X$ and $\boldsymbol{s}_Y$ are set-rep vectors for sets X and Y respectively, and initially have the same values, i.e., $\boldsymbol{s}_X = \boldsymbol{s}_Y = \boldsymbol{s}$. Then, set-rep and item vectors in $X_{\boldsymbol{s}}$ and $Y_{\boldsymbol{s}}$ are transformed through an asymmetric self- and cross-attention mechanism, i.e., bi-PMA and pivot-cross, in the backbone to embed discriminative information among sets $X_{\boldsymbol{s}}$ and $Y_{\boldsymbol{s}}$ into each of the set-rep vectors $\widehat{\boldsymbol{s}_X}$ and $\widehat{\boldsymbol{s}_Y}$ as follows:

1. bi-PMA: self-attention where the convex hull of X is expanded by adding $\boldsymbol{s}$ into key and value sets as follows:

$$\widehat{X_{\boldsymbol{s}}} = \text{bi–PMA}(X_{\boldsymbol{s}}) \equiv \text{multiAtt}(X_{\boldsymbol{s}}, X_{\boldsymbol{s}}, X_{\boldsymbol{s}}). \tag{9}$$

2. pivot-cross: cross-attention where the convex hull of X is further expanded by using the union of two sets X and Y as key and value sets as follows:

$$\widehat{X_{\boldsymbol{s}}} = \text{pivot–cross}(X_{\boldsymbol{s}}, Y_{\boldsymbol{s}}) \equiv \text{multiAtt}(X_{\boldsymbol{s}}, [X_{\boldsymbol{s}}; Y_{\boldsymbol{s}}], [X_{\boldsymbol{s}}; Y_{\boldsymbol{s}}]). \tag{10}$$

With transformed set-rep vectors $\widehat{\boldsymbol{s}_X}$ and $\widehat{\boldsymbol{s}_Y}$, the set-similarity can be simply computed using the inner product $\widehat{\boldsymbol{s}_X}\widehat{\boldsymbol{s}_Y}^\top$ in the head.

However, its reliance on dot-product similarity for computing attention weights $\boldsymbol{a}_i$ (Eq. 4) introduces two potential limitations:

1. **Scale sensitivity and convex hull constraint:** Dot-product attention produces scalar similarity scores that are sensitive to the input dimensionality, typically requiring scaling (e.g., by $\sqrt{D'}$) and softmax normalization for numerical stability. However, this normalization restricts the output to lie within the convex hull of the linearly projected item vectors, limiting the ability of the model to extrapolate beyond the span of the input set.

2. **Loss of directional and magnitude information:** The dot product reduces the interaction between item vectors to a single scalar value, discarding explicit information about their directional and distance-based relationships. This simplification hinders the ability of the model to capture fine-grained or asymmetric dependencies-particularly important for heterogeneous set matching.

2.4. MLP-based permutation-invariant aggregation

DuMLP-Pin (Fei et al., 2022) offers an alternative approach for dynamically computing coefficients to aggregate item vectors by applying item-wise multilayer perceptrons (MLPs) as follows:

$$A = \mathrm{MLP}_{\mathrm{ch}_1}\big(X\big) \in \mathbb{R}^{N_X \times N_{\mathrm{coef}}}, \quad \widetilde{X} = \big[\widehat{\boldsymbol{x}^1}; \widehat{\boldsymbol{x}^2}; \ldots; \widehat{\boldsymbol{x}^{N_{\mathrm{coef}}}}\big] = A^\top \mathrm{MLP}_{\mathrm{ch}_2}\big(X\big) \in \mathbb{R}^{N_{\mathrm{coef}} \times D}, \tag{11}$$

where $\mathrm{MLP}_{\mathrm{ch}_1}(X)$ and $\mathrm{MLP}_{\mathrm{ch}_2}(X) \in \mathbb{R}^{N_Q \times D}$ are individually applied to the channel direction of each item $\boldsymbol{x} \in \mathcal{X}$ like the channel-mixer (Tolstikhin et al., 2021b). Here, the output of $\mathrm{MLP}_{\mathrm{ch}_1}(X)$ is interpreted as N_{coef} sets of aggregation coefficients. The resulting N_{coef} aggregated vectors are then fused using an $\mathrm{MLP}_{\mathrm{coef}}(\cdot)$ for coefficient direction:

$$\widehat{\boldsymbol{x}} = \mathrm{DuMLP}(X) = \mathrm{MLP}_{\mathrm{coef}}\Big(\big[\widehat{\boldsymbol{x}^1}^\top, \widehat{\boldsymbol{x}^2}^\top, \ldots, \widehat{\boldsymbol{x}^{N_{\mathrm{coef}}}}^\top\big]\Big) \in \mathbb{R}^{1 \times D}. \tag{12}$$

This formulation allows the model to compute aggregation coefficients without relying on dot-product similarity and softmax normalization, thereby removing constraints such as convex hull limitations on the transformation space discussed in Secs. 2.2 and 2.3.

However, DuMLP-Pin produces only a single aggregated vector $\widehat{\boldsymbol{x}}$ for the set X, which does not retain item-specific information, i.e., $\boldsymbol{x}_i$. As a result, it limits the ability to perform dynamic item-wise transformations that are essential for heterogeneous set-to-set matching tasks.

3. Proposed method

We introduce a novel set aggregation method called **Deviation-based multiple coefficient item Mixer (DeviMix)** to address limitations found in both dot-product-based attention (Secs. 2.2 and 2.3) and global pooling approaches like DuMLP-Pin (Sec. 2.4).

Unlike attention, which computes scalar similarity via dot products, or DuMLP-Pin, which generates a single global aggregation, DeviMix dynamically computes item-specific coefficients using an MLP and cross deviation between each item and the rest of the set and provides several advantages:

- **Directional and positional awareness:** Deviation vectors encode explicit dimension-wise differences, unlike cosine similarity, which captures only angular closeness.

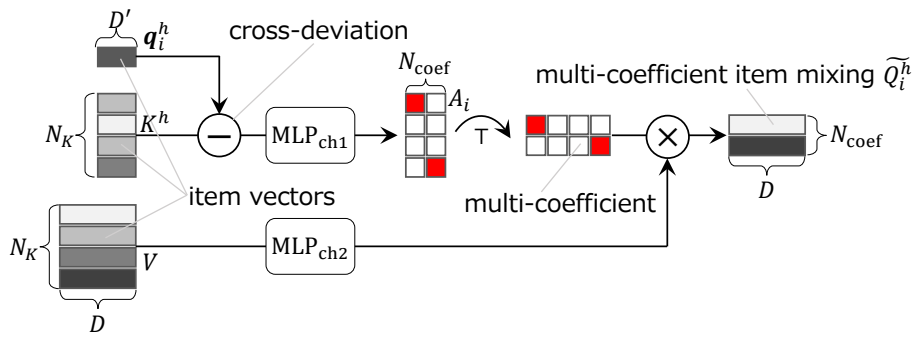


Figure 1: DeviMix module using cross-deviation and channel-wise MLP for generating multiple coefficient and multiple item mixing.

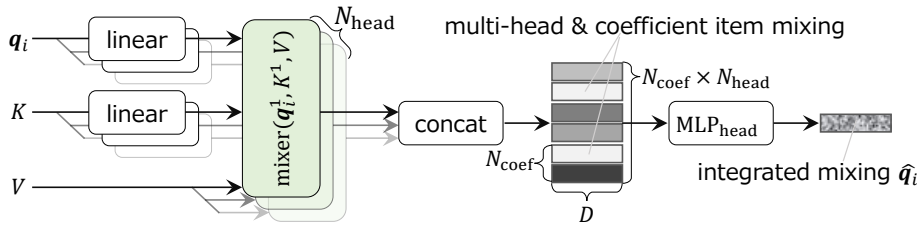


Figure 2: Overview of multi-head and multi-coefficient item mixing in DeviMix.

- **No normalization required:** The original scale of the item vector is preserved, eliminating the need for softmax and its associated convex hull constraint.
- **Fine-grained aggregation:** Item-wise cross-deviation allows dynamic coefficient generation for transforming each item individually.

3.1. Deviation-based multiple coefficient item Mixer (DeviMix)

Given a query item $\boldsymbol{q}_i^h = \boldsymbol{q}_i W_Q^h$ and a set of keys $K^h = K W_K^h$ (from Eq. 5), we compute N_{coef} item mixing coefficients by applying a channel-wise MLP to the deviations:

$$A_i^h = \text{MLP}_{\text{ch}_1}\left(\boldsymbol{q}_i^h - K^h\right) \in \mathbb{R}^{N_K \times N_{\text{coef}}}, \tag{13}$$

where MLP_{ch} is a three-layer MLP with the number D_h of hidden nodes. For the set Q as the query input, the cross-deviation Eq. 13 is extended to compute deviations for all possible item pairs between the sets Q and K using a tiled expression as follows:

$$A^h = \text{MLP}_{\text{ch}_1}\left(\text{tile}\big(Q^h, N_K, 1\big) - \text{tile}\big(K^h, N_Q, 0\big)\right) \in \mathbb{R}^{N_Q \times N_K \times N_{\text{coef}}}, \tag{14}$$

where $\text{tile}(X, n, d)$ denotes repeating the matrix X n times along dimension d. For example, $\text{tile}\big(Q^h, N_K, 1\big) \in \mathbb{R}^{N_Q \times N_K \times D'}$.

These coefficients are then used for multiple item-specific mixing over the value set V as described in Fig. 1 and follow:

$$\widetilde{Q_i^h} = \left[\widehat{\boldsymbol{q}_i^{h1}}; \widehat{\boldsymbol{q}_i^{h2}}; \ldots; \widehat{\boldsymbol{q}_i^{hN_{\text{coef}}}}\right] = \text{mixer}\big(\boldsymbol{q}_i^h, K^h, V\big) = A_i^{h\top} \text{MLP}_{\text{ch}_2}(V) \in \mathbb{R}^{N_{\text{coef}} \times D}, \tag{15}$$

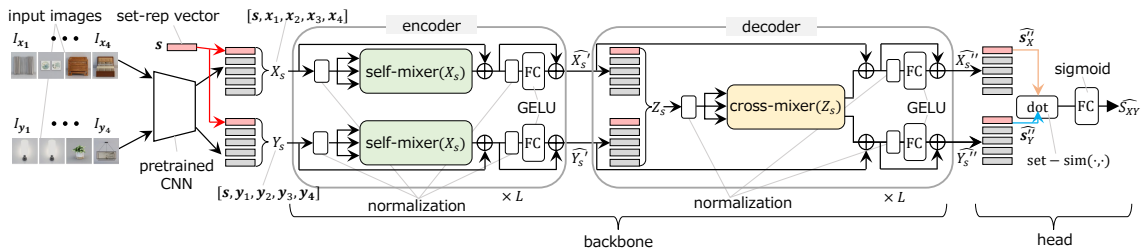


Figure 3: Architecture of set-to-set matching with proposed DeviMix.

where let us note that only query item $\boldsymbol{q}^h$ and key set K^h depend on the head index h but not the value set V.

Because the coefficients A^h are computed for all possible item pairs, i.e., $\boldsymbol{q}_i^h$ vs. K^h (Eq. 13), or Q^h vs. K^h (Eq. 14), using the fully tiled cross-deviation between Q and K, the subsequent aggregation over the value set V via matrix multiplication preserves permutation invariance over items in set V.

Finally, to integrate the item mixing over $N_{\text{coef}} \times N_{\text{head}}$ combinations of coefficients and heads, we apply a final aggregation step using a head-and-coefficient-wise mixer MLP with the number D_h of hidden nodes. This operation produces the item-specific transformed vector $\widehat{\boldsymbol{q}}_i$ for each query item $\boldsymbol{q}_i$ as depicted in Fig. 2 and as follows:

$$\widehat{\boldsymbol{q}}_i = \text{multi–mixer}(\boldsymbol{q}_i, K, V) = \text{MLP}_{\text{head}}\Big([\widetilde{Q_i^1}^\top, \widetilde{Q_i^2}^\top, \cdots, \widetilde{Q_i^{N_{\text{head}}}}^\top] \Big) \in \mathbb{R}^{1\times D}. \tag{16}$$

3.2. Architecture of entire network

To effectively transform the input sets X and Y and extract the set-rep vector $\boldsymbol{s}$ (Sec. 2.3) for set-to-set matching, we construct a backbone module backbone$(\cdot)$ that alternately applies two types of DeviMix operations: self–mixer$(\cdot)$ for intra-set interactions and cross–mixer$(\cdot)$ for inter-set interactions as follows:

$$\begin{aligned} \widehat{X_{\boldsymbol{s}}} &= \text{self–mixer}(X_{\boldsymbol{s}}) = \text{multi–mixer}(X_{\boldsymbol{s}}, X_{\boldsymbol{s}}, X_{\boldsymbol{s}}), \\ [\widehat{X_{\boldsymbol{s}}}; \widehat{Y_{\boldsymbol{s}}}] &= \text{cross–mixer}(Z_{\boldsymbol{s}}) = \text{multi–mixer}(Z_{\boldsymbol{s}}, Z_{\boldsymbol{s}}, Z_{\boldsymbol{s}}), \quad Z_{\boldsymbol{s}} = [X_{\boldsymbol{s}}; Y_{\boldsymbol{s}}], \end{aligned} \tag{17}$$

where due to the permutation invariance of the proposed DeviMix architecture, the output of the cross-mixer, e.g., $\widehat{X_{\boldsymbol{s}}}$, remains unchanged even if the concatenated input set $Z_{\boldsymbol{s}} = [X_{\boldsymbol{s}}; Y_{\boldsymbol{s}}]$ is reordered as $Z_{\boldsymbol{s}} = [Y_{\boldsymbol{s}}; X_{\boldsymbol{s}}]$.

An example of the network structure for the set of images $\mathcal{X}$ and $\mathcal{Y}$, when applying our proposed methods, DeviMix with the set-rep vector, is depicted in Fig. 3. First, images in the set $\mathcal{X}$ and $\mathcal{Y}$, i.e., $\{I_{\boldsymbol{x}_1}, I_{\boldsymbol{x}_2}, \ldots\}$ and $\{I_{\boldsymbol{y}_1}, I_{\boldsymbol{y}_2}, \ldots\}$, are input into a pre-trained convolutional neural network (CNN). Given the CNN feature maps, the global average pooling is used to obtain the feature vector of items, i.e., $[\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots]$ and $[\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots]$.

Next, a common set-rep vector $\boldsymbol{s}$ is added to each set, and then item and set-rep vectors, i.e., $X_{\boldsymbol{s}}$ and $Y_{\boldsymbol{s}}$, are transformed through the encoder and decoder networks, each of which is repeated at L times in the backbone. More specifically, as encoder and decoder, the following residual block is used as follows:

Table 1: Statistics of each dataset used for experimental evaluation where N_{tr}, N_{val}, and N_{te} are numbers of training, validation, and test sets, respectively. N, $N_{\mathcal{G}}$, and D are the maximum number of items in sets, the number of candidate sets in the gallery $\mathcal{G}$, and the dimension of item vector, respectively.

dataset	N_{tr}	N_{val}	N_{te}	N	# of queries in test	$N_{\mathcal{G}}$	D
Fashion	30,815	3,851	3,851	5	19,255	5	4,096
Furniture	10,708	1,338	1,339	8	6,695	5	4,096

- For encoder with DivMix:

$$\widehat{X'_{\boldsymbol{s}}} = X_{\boldsymbol{s}} + \mathrm{self\text{–}mixer}\big(\mathrm{snorm}(X_{\boldsymbol{s}})\big), \quad \widehat{X''_{\boldsymbol{s}}} = \widehat{X'_{\boldsymbol{s}}} + \mathrm{FC}\big(\mathrm{norm}(\widehat{X'_{\boldsymbol{s}}})\big), \tag{18}$$

- For decoder with DiVMix:

$$\begin{aligned} Z_{\boldsymbol{s}} &= \big[X_{\boldsymbol{s}}; Y_{\boldsymbol{s}}\big], \quad \big[\widehat{X'_{\boldsymbol{s}}}; \widehat{Y'_{\boldsymbol{s}}}\big] = Z_{\boldsymbol{s}} + \mathrm{cross\text{–}mixer}\big(\mathrm{cnorm}(Z_{\boldsymbol{s}})\big), \\ \widehat{X''_{\boldsymbol{s}}} &= \widehat{X'_{\boldsymbol{s}}} + \mathrm{FC}\big(\mathrm{norm}(\widehat{X'_{\boldsymbol{s}}})\big), \quad \widehat{Y''_{\boldsymbol{s}}} = \widehat{Y'_{\boldsymbol{s}}} + \mathrm{FC}\big(\mathrm{norm}(\widehat{Y'_{\boldsymbol{s}}})\big), \end{aligned} \tag{19}$$

where $\mathrm{FC}(\cdot)$ indicates item-wise (channel-direction) fully connected network, and $\mathrm{snorm}(\cdot)$ and $\mathrm{cnorm}(\cdot,\cdot)$ denote set normalization and cross-set normalization (Hachiya and Saito, 2024), respectively, and $\mathrm{norm}(\cdot)$ denotes Layer Normalization.

Given the set-rep vectors of $\widehat{\boldsymbol{s}''_X}$ and $\widehat{\boldsymbol{s}''_Y}$, in the head, the set-similarity (Eq. 3) is computed using the dot product as follows:

$$\mathrm{set\text{–}sim}(\widehat{X''}, \widehat{Y''}) = \widehat{\boldsymbol{s}''_X}\,\widehat{\boldsymbol{s}''_Y}^{\top}. \tag{20}$$

For training the backbone and head networks, the mini-batch data are created by combining all set-pairs $(X, Y) \in \mathcal{D}_b \times \mathcal{D}_b$ in the randomly selected set-data $\mathcal{D}_b = \{X^b\}_{b=1}^{B}$ from the training sets where B is the size of mini-batch. Parameters in the backbone and head networks, including the vector $\boldsymbol{s}$, are initialized randomly, e.g., using Xavier initialization, and are then updated to minimize the mean cross-entropy $\mathcal{L}_b$ as follows:

$$\mathcal{L}_b \equiv -\frac{1}{B(B-1)} \sum_{(X,Y)\in\mathcal{D}_b\times\mathcal{D}_b, X\neq Y} s_{XY} \log \widehat{s_{XY}}. \tag{21}$$

4. Experimental evaluation

In this section, we demonstrate the effectiveness of the proposed methods through experiments on heterogeneous set-to-set matching tasks: fashion-outfit matching and furniture-coordination matching.

4.1. Datasets

We used two datasets:

- **Fashion-outfit dataset using Shift15M** (Kimura et al., 2021; Research, 2023): Shift15M is the dataset collected by a fashion social networking service, IQON, with approximately two million users in the past decade. The outfit (set) is a highly rated combination of multiple item images of different types, e.g., outerwear, tops, bottoms, shoes, and bags; thus, the matching problem is heterogeneous. Each item image is provided as a 4,096-dimensional feature vector.

  Following the experimental setting (Hachiya, 2024), we used 38,517 outfits collected in 2017 with a data split seed of 0—a ratio of train, validation, and test is 8:1:1, and limited the maximum number of items per set to $N = 5$.

  For evaluation, we used 3,851 pre-split test sets. Query-gallery pairs were constructed by selecting each test set as a query five times and pairing it with a gallery composed of one positive set and four randomly sampled negative sets. The detailed settings of the dataset are summarized in Table 1.

- **Furniture-coordination dataset using DeepFurniture**: The DeepFurniture dataset (Liu et al., 2019a) 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 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 pre-trained 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 (supplementary material) 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.

  To create positive set pair, we randomly shuffle the order of items in each coordinate and split into two subsets, i.e., X and Y. Examples of positive pairs with varying numbers of items (N_X/N_Y) are shown in Fig. 2 (supplementary material). 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.

4.2. Comparison methods

We compared the performance of the proposed methods with several existing methods.

- proposed (Sec. 3): As shown in Fig. 3, the proposed architecture uses the self-mixer (Eq. 18) and cross-mixer (Eq. 19) 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., $\widehat{\boldsymbol{s}''_X}\widehat{\boldsymbol{s}''_Y}^\top$.

The detailed settings are as follows. The number of hidden nodes in each MLP (i.e., $\mathrm{MLP}_{\mathrm{ch}_1}$, $\mathrm{MLP}_{\mathrm{ch}_2}$, $\mathrm{MLP}_{\mathrm{coef}}$, and $\mathrm{MLP}_{\mathrm{head}}$) is set to $D_h = D/2$, and GELU is used as the activation function for all hidden layers. The output layer of $\mathrm{MLP}_{\mathrm{coef}}$ 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 $D' = \frac{D}{N_{\mathrm{head}}}$, and the number of coefficients is set at $N_{\mathrm{coef}} = 8$ in order to match the number of parameters with the attention mechanism, as described in Sec. 2.1 (supplementary material).

- Cross-Set Feature Transformation (CSeFT) (Saito et al., 2020): standard attention-based set-to-set matching method where self- and cross-attention (Eq. 7) are used in the backbone. 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}. Following the implementation of multi-head attention (Vaswani et al., 2017), the dimension of linearly projected item vectors is set as $D' = \frac{D}{N_{\mathrm{head}}}$. The detailed information is described in the supplementary document (Sec. 1.1 in the supplementary material).

- Pivot-Attention (Hachiya and Saito (2024), in Sec. 2.3): A state-of-the-art set-to-set matching method based on the attention mechanism. In this method, bi-PMA (Eq. 9) and pivot-attention (Eq. 10) are used in the encoder and decoder, respectively. As the set similarity metric, the dot product between the final set-rep vectors, i.e., $\widehat{\boldsymbol{s}''_X}\widehat{\boldsymbol{s}''_Y}^\top$, 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 $D' = \frac{D}{N_{\mathrm{head}}}$. The detailed information is described in Sec. 1.1 (supplementary material).

- poolFormer + set-rep vector: To evaluate the effectiveness of the proposed item-specific mixer modules, e.g., self- and cross-mixer (Eq. 17), we constructed a baseline method where the mixers are replaced with global average pooling—a variant of poolFormer (Yu et al., 2022). The final set similarity is computed as the dot product between the transformed set-rep vectors, i.e., $\widehat{\boldsymbol{s}''_X}\widehat{\boldsymbol{s}''_Y}^\top$. The detailed information is described in Sec. 1.1 (supplementary material).

- poolFormer (Yu et al., 2022): The set-rep vector $\boldsymbol{s}$ is further removed from Eq. 3 (supplementary material)—$X_{\boldsymbol{s}}$ and $Y_{\boldsymbol{s}}$ 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.,

2021a). Specifically, we generate the set of all r-permutations of the input set X, denoted by $P_r(X)$, and apply an item mixer $\mathrm{MLP}_{\mathrm{item}}(\cdot) \in \mathbb{R}^{1\times D}$ to each permutation. Then, the outputs are then averaged to aggregate information across all possible permutations—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. The detailed information is described in Sec. 1.1 (supplementary material).

4.2.1. Hyperparameter and metrics

For training, to create the mini-batch of set-data $\mathcal{D}_b$, we randomly selected $B = 15$ sets from $\mathcal{D}_{\mathrm{tr}}$ without replacement. For each selected set, we randomly permuted the item order and split it into two subsets to generate a positive pair of sets, X and Y, resulting in B^2 set pairs: B positive and $B(B-1)$ negative. In addition, we applied negative down-sampling by randomly removing negative pairs to balance the ratio between positive and negative pairs when computing the loss Eq. 21.

The number of layers in the backbone network is set to $L \in \{1, 3\}$. The number of heads for self-attention, cross-attention, self-mixer, cross-mixer, and CSS is set to $N_{\mathrm{head}} = 8$, and the number of dimensions D of the item vector is set to $D = 64$.

For the validation, we applied early stopping with 10-epoch patience based on the validation accuracy. For the quantitative evaluation, we used the averaged cumulative match characteristic (CMC) (Moon and Jonathon, 2001) given a gallery $\mathcal{G}$ with the number of candidates $N_{\mathcal{G}} = 5$.

To mitigate the effects of random initialization, we conducted five independent training and evaluation trials for each method, using different random seeds (1 through 5) to initialize model parameters. For each method, we selected the top three trials based on their CMC1 scores and computed the mean and standard deviation of the averaged CMC values across those selected trials.

4.3. Main results

The experimental results are summarized in Table 2. Firstly, as shown in Table 2, the baseline models—poolFormer and poolFormer + set-rep vector—which replace the item-specific aggregation in our self- and cross-mixer modules with global average pooling, suffer from notable performance degradation and training instability. While the introduction of set-rep vectors partially mitigates these issues, the results highlight the critical role of item-specific dynamic aggregation in achieving robust performance for heterogeneous set-to-set matching tasks.

The proposed method, DeviMix, addresses this need by computing multiple aggregation coefficients per item using MLPs applied to cross-deviation vectors. As a result, it achieves the best or competitive performance compared to the state-of-the-art pivot-attention method, as shown in the table.

Furthermore, the performance of DeviMix tends to improve as the number of layers L increases. Importantly, even with a shallow configuration ($L = 1$), the model retains strong performance, indicating its robustness. This stability stems from the expressive power of the cross-deviation-based multiple coefficient generation, which enables discriminative

Table 2: Performance comparison in fashion-outfit and furniture-coordination matching. To mitigate the effects of random initialization, five independent training and evaluation trials and top three trials based on CMC1 are selected for each method, and the mean and standard deviation of three CMCs and losses for each method are listed. The best scores at each metric are highlighted in bold.

dataset	model	(L, N_{head})	CMC 1↑	CMC 2↑	CMC 3↑	loss ↓
Fashion	CSeFT	(1, 8)	58.0 (32.7)	75.9 (31.3)	85.7 (22.1)	2.801 (4.241)
		(3, 8)	83.0 (1.2)	96.6 (0.2)	99.2 (0.0)	0.295 (0.009)
	pivot-attention	(1, 8)	78.5 (1.4)	94.3 (0.2)	98.4 (0.2)	0.346 (0.011)
		(3, 8)	82.0 (2.0)	95.9 (0.8)	99.0 (0.3)	0.299 (0.016)
	poolFormer + set-rep vector	(1, –)	53.8 (0.9)	78.5 (0.7)	91.0 (0.5)	0.533 (0.006)
		(3, –)	77.4 (0.2)	93.8 (0.3)	98.4 (0.1)	0.360 (0.015)
	poolFormer	(1, –)	20.8 (0.6)	41.5 (1.4)	61.1 (1.4)	7.704 (0.072)
		(3, –)	37.3 (29.6)	57.5 (30.7)	72.8 (22.0)	5.235 (4.182)
	Janossy pooling	(1, –)	20.6 (0.6)	40.1 (0.6)	60.4 (0.4)	7.715 (0.023)
		(3, –)	40.6 (35.3)	58.5 (32.4)	73.2 (22.5)	5.246 (4.279)
	proposed (DeviMix)	(1, 8)	79.6 (3.2)	95.3 (1.1)	98.8 (0.4)	0.316 (0.032)
		(3, 8)	**84.5** (0.3)	**96.7** (0.4)	**99.4** (0.1)	**0.292** (0.013)
Furniture	CSeFT	(1, 8)	67.4 (1.5)	86.9 (0.7)	94.7 (0.4)	0.453 (0.007)
		(3, 8)	73.9 (0.8)	90.9 (1.2)	96.6 (0.4)	0.429 (0.004)
	pivot-attention	(1, 8)	72.5 (0.8)	89.6 (0.7)	96.7 (0.1)	0.395 (0.007)
		(3, 8)	**75.4** (2.2)	**91.3** (1.6)	97.2 (0.7)	0.390 (0.011)
	poolFormer + set-rep vector	(1, –)	47.1 (4.7)	71.7 (5.2)	86.1 (2.8)	0.590 (0.024)
		(3, –)	59.1 (3.2)	82.4 (2.4)	93.7 (0.7)	0.496 (0.025)
	poolFormer	(1, –)	47.3 (2.2)	72.7 (0.7)	86.3 (0.5)	0.609 (0.023)
		(3, –)	40.2 (17.2)	63.9 (19.7)	78.9 (16.1)	2.973 (4.157)
	Janossy Pooling	(1, –)	32.7 (18.6)	55.2 (20.8)	72.0 (17.9)	5.339 (4.159)
		(3, –)	50.9 (26.3)	72.6 (27.2)	84.5 (20.9)	2.835 (4.149)
	proposed (DeviMix)	(1, 8)	69.0 (1.9)	88.8 (0.8)	96.7 (0.4)	0.409 (0.015)
		(3, 8)	73.5 (3.6)	91.1 (1.5)	**97.6** (0.5)	**0.375** (0.016)

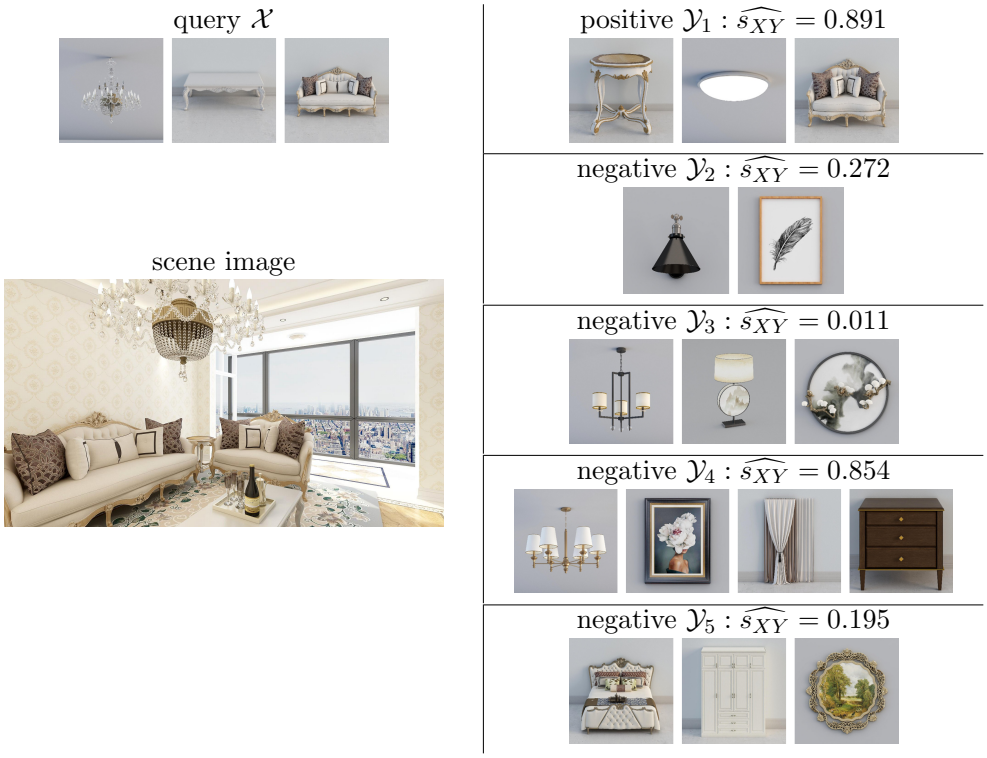
Table 3: Performance comparison with different numbers of item-mixing coefficients N_{coef} of the proposed method, DeviMix, in the furniture-coordination matching. The setting is same as Table 2.

model	$(L, N_{\text{head}}, N_{\text{coef}})$	CMC 1↑	CMC 2↑	CMC 3↑	loss ↓
proposed (DeviMix)	(3, 8, 1)	72.8 (1.0)	90.4 (0.8)	97.0 (0.6)	0.391 (0.011)
	(3, 8, 2)	73.3 (1.1)	**91.2** (0.7)	97.3 (0.3)	**0.373** (0.006)
	(3, 8, 4)	73.2 (1.6)	90.9 (0.5)	97.4 (0.3)	0.374 (0.005)
	(3, 8, 8)	**73.5** (3.6)	91.1 (1.5)	**97.6** (0.5)	0.375 (0.016)
	(3, 8, 16)	71.9 (1.5)	90.5 (1.1)	97.2 (0.7)	0.379 (0.014)

item-wise transformations and effectively overcomes the convex hull constraint and limited expressivity of dot-product attention, as discussed in Sections 2.2 and 2.3.

In addition, to evaluate the robustness of the proposed method with respect to the number of item-mixing coefficients, Table 3 presents the performance of DeviMix with different coefficient settings, $N_{\text{coef}} \in \{1, 2, 4, 8, 16\}$, in the furniture-coordination matching task. The results show that the proposed method consistently maintains high performance across dif-

Table 4: Examples of images in sets $\mathcal{X}$ and $\{\mathcal{Y}_1, \mathcal{Y}_2, \ldots, \mathcal{Y}_5\} = \mathcal{G}$ on furniture-coordination matching, and predicted scores $\widehat{s_{XY}}$ using the proposed methods, DeviMix, when $L = 3$, $N_{\text{head}} = 8$, and $N_{\text{coef}} = 8$.



ferent settings, demonstrating that DeviMix is robust to the choice of N_{coef}. This suggests that the dynamic item-mixing mechanism effectively captures discriminative relationships without relying heavily on a specific coefficient dimensionality.

The ablation study to evaluate each module of the proposed method is presented in the supplementary document.

Tables 4 shows examples of images $\{I_{\boldsymbol{x}_1}, I_{\boldsymbol{x}_2}, \ldots\}$ and $\{I_{\boldsymbol{y}_1}, I_{\boldsymbol{y}_2}, \ldots\}$ 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 furnitures combining ones in the query $\mathcal{X}$ and the positive candidate $\mathcal{Y}_1$, and not used in the matching task.

The table shows that the query set $\mathcal{X}$ and candidate sets $\mathcal{Y}_k$ contain various types of furniture, such as doors, wall art, cabinets, desks, tables, and chairs, resulting in a heterogeneous matching scenario. In such cases, the matching label cannot be determined solely based on the similarity of item vectors between sets. It can also be observed that positive candidate sets $\mathcal{Y}_1$ tend to include items that are not only functionally complementary to those in the query set X (e.g., desks paired with chairs), but also visually consistent in style (e.g., dark wooden finishes). This highlights the importance of both functional compatibil-

ity and visual coherence in determining set-level match quality, especially in applications such as furniture coordination and fashion outfit recommendation.

Overall, these experimental results indicate that the proposed DeviMix, where multiple item-mixing coefficients are generated using MLP based on the cross-deviation between item pairs, could be an effective approach for the heterogeneous set-to-set matching problems.

5. Conclusion

We have presented DeviMix, a novel MLP-based method for heterogeneous set-to-set matching that addresses the limitations of existing attention- and pooling-based approaches. By computing multiple aggregation coefficients through MLPs applied to cross-deviation vectors between items, DeviMix enables expressive and item-specific set transformations without relying on dot-product similarity or softmax normalization. Through experiments on fashion outfit and furniture coordination tasks, we demonstrated that DeviMix achieves superior performance over state-of-the-art methods. These results highlight the effectiveness of leveraging cross-deviation and MLP-based aggregation in capturing fine-grained compatibility within heterogeneous sets.

Acknowledgments

This work was supported by JSPS KAKENHI Grant Number JP23K11218.

References

Jiajun Fei, Ziyu Zhu, Zhidong Deng Wenlei Liu, Mingyang Li, Huanjun Deng, and Shuo Zhang. Dumlp-pin: A dual-mlp-dot-product permutation-invariant network for set feature extraction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 598–606, 2022.

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.

Wei-Lin Hsiao and Kristen Grauman. Creating capsule wardrobes from fashion images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7161–7170, 2018.

Ziling Huang, Zheng Wang, Chung-Chi Tsai, Shinichi Satoh, and Chia-Wen Lin. Dotscn: Group re-identification via domain-transferred single and couple representation learning. *IEEE Transactions on Circuits and Systems for Video Technology*, 31(7):2739–2750, 2021. doi: 10.1109/TCSVT.2020.3031303.

Masanari Kimura, Takuma Nakamura, and Yuki Saito. Shift15m: Multiobjective large-scale fashion dataset with distributional shifts, 2021.

Juho Lee, Yoonho Lee, Jungtaek Kim, Adam R. Kosiorek, Seungjin Choi, and Yee Whye Teh. Set transformer: A framework for attention-based permutation-invariant neural networks. In *Proceedings of International Conference on Machine Learning (ICML)*, volume 97, pages 3744–3753, 2019.

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)*, 2019a.

Xiaofeng Liu, Zhenhua Guo, Site Li, Lingsheng Kong, Ping Jia, Jane You, and B.V.K. Kumar. Permutation-invariant feature restructuring for correlation-aware image set-based recognition. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2019b.

Hyeonjoon Moon and Phillips P. Jonathon. Computational and performance aspects of pca-based face recognition algorithms. *Perception*, 30, 2001.

Ryan L. Murphy, Balasubramaniam Srinivasan, Vinayak Rao, and Bruno Ribeiro. Janossy pooling: Learning deep permutation-invariant functions for variable-size inputs. 2019.

Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, 2017.

ZOZO Research. Shift15m: Fashion-specific dataset for set-to-set matching with several distribution shifts. https://github.com/st-tech/zozo-shift15m, 2023. Accessed: 2025-05-23.

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.

Govind Sharma, Swyam Prakash Singh, Devi V. Susheela, and Murty M. Narasimha. The cat set on the mat: Cross attention for set matching in bipartite hypergraphs. *arXiv:2111.00243v1*, 2021.

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. 2021a.

Ilya Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Jessica Yung, Andreas Steiner, Daniel Keysers, Jakob Uszkoreit, Mario Lucic, and Alexey Dosovitskiy. Mlp-mixer: An all-mlp architecture for vision. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 34, pages 24261–24272, 2021b.

Mariya I. Vasileva, Bryan A. Plummer, Krishna Dusad, Shreya Rajpal, Ranjitha Kumar, and David Forsyth. Learning type-aware embeddings for fashion compatibility. In *Proceedings of the 15th European Conference on Computer Vision (ECCV)*, pages 390–405, 2018.

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

Hao Xiao, Weiyao Lin, Bin Sheng, Ke Lu, Junchi Yan, Jingdong Wang, Errui Ding, Yihao Zhang, and Hongkai Xiong. Group re-identification: Leveraging and integrating multi-grain information. In *Proceedings of the ACM International Conference on Multimedia (ICM)*, pages 192–200, 2018.

Jiaolong Yang, Peiran Ren, Dongqing Zhang, Dong Chen, Fang Wen, Hongdong Li, and Gang Hua. Neural aggregation network for video face recognition. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, 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.

Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Ruslan Salakhutdinov, and Alexander Smola. Deep sets. *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, 30, 2017.