Exchangeable Deep Neural Networks for Set-to-Set Matching and Learning

Yuki Saito[1,2][0000−0003−0492−414X], Takuma Nakamura[1][0000−0001−7904−4724],
Hirotaka Hachiya[3][0000−0003−3748−4101], and
Kenji Fukumizu[4,2][0000−0002−3488−2625]

[1] ZOZO Research, Jingumae, Shibuya, Tokyo, Japan
{yuki.saito,takuma.nakamura}@zozo.com
[2] The Graduate University for Advanced Studies, SOKENDAI, Tachikawa, Tokyo,
Japan
[3] Wakayama University, Wakayama-shi, Wakayama, Japan
hhachiya@wakayama-u.ac.jp
[4] The Institute of Statistical Mathematics, Tachikawa, Tokyo, Japan
fukumizu@ism.ac.jp

Abstract. Matching two different sets of items, called heterogeneous set-to-set matching problem, has recently received attention as a promising problem. The difficulties are to extract features to match a correct pair of different sets and also preserve two types of exchangeability required for set-to-set matching: the pair of sets, as well as the items in each set, should be exchangeable. In this study, we propose a novel deep learning architecture to address the abovementioned difficulties and also an efficient training framework for set-to-set matching. We evaluate the methods through experiments based on two industrial applications: fashion set recommendation and group re-identification. In these experiments, we show that the proposed method provides significant improvements and results compared with the state-of-the-art methods, thereby validating our architecture for the heterogeneous set matching problem.

Keywords: set to set matching; deep learning; permutation invariance

1   Introduction

Matching pairs of data is a crucial part of many machine learning tasks, including recommendation [61, 56, 36], person re-identification (re-id) [87], image search [74], and face recognition [54], as typical industrial applications. Over the past decade, a deep learning framework for matching up data, e.g., images, has served as the core of such systems.

Aside from these tasks, set-to-set matching, which is an extension of multiple instance matching, has recently been identified as an important element in various applications required by emerging web technologies or services. A representative example in e-commerce is fashion recommendation, where a group of fashion items deemed to match the collection of fashion items already owned
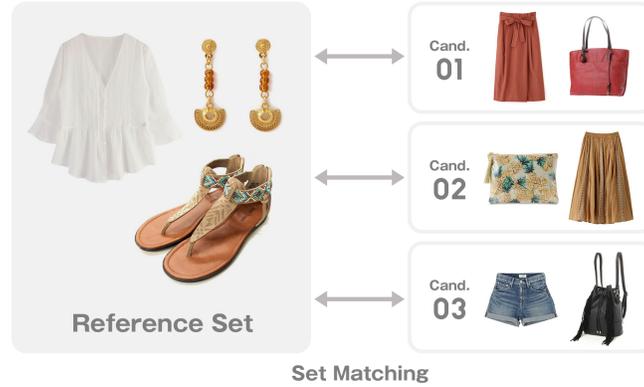
Fig. 1: One of the main questions that set-to-set matching attempts to answer is as follows: which candidate is more compatible than others with the reference set? Here, we consider the matching of the reference set and the respective candidate set and then selecting the best pair.

by a user is recommended. Regarding the group as an unordered set, we can consider this task a set-to-set matching problem, as shown in Figure 1. Another example is group re-identification (group re-id) in surveillance systems [41, 77, 40], which has recently started implementing a function to track known groups of suspicious persons or criminals, a task that can also be simplified as a set-to-set matching problem. Other examples include image-set retrieval [11, 9], image-set classification [45], image-set reconstruction [42], person re-id [44], taxonomy matching [59], cross-lingual matching [29], relational data matching [30], and face verification [43, 78]. Earlier studies have also explored face recognition as a set-to-set matching problem [63, 1, 6, 79] and next-basket recommendation [56].

Set matching scenarios can be grouped into two classes: homogeneous set matching and heterogeneous set matching. In the former, two positive sets comprising the same instances, such as the images of the face of the same person, are to be matched. Except for variations such as differences in illumination or pose in the images, both sets contain similar instances. This scenario has been investigated in several studies [11, 45, 9, 44, 43, 78, 63, 1, 6, 79, 42]. In the heterogeneous case, the instances within paired sets can be considerably different, as is the case in fashion recommendation and group re-id. To the best of our knowledge, there are very few studies on constructing deep learning models for heterogeneous set matching. We consider that matching heterogeneous sets is a more difficult task and requires a strong learning architecture to match different sets.

Furthermore, another fundamental difficulty in set-to-set matching, compared with ordinary data matching, lies in the two types of exchangeability required: exchangeability between the pair of sets and invariance across different permutations of the items in each set. A function that calculates a matching

score should provide an invariant response, regardless of the order of the two sets, or the permutations of the items.

The main focus of this paper is an architecture that preserves the aforementioned exchangeability properties, and at the same time, realizes a high performance in heterogeneous set matching tasks. In this study, we argue that allowing the feature extractor and matching layer to include interactions between the two sets is crucial to identify matching pairs among different items. We propose a deep learning model for (1) feature extraction, named *cross-set feature transformation* (CSeFT), which iteratively provides the interactions between the pair of sets to each other in the intermediate layers. Our novel functions, *attention-* and *affinity-based functions*, organize the CSeFT spanning two different sets in the feature spaces, thereby improving the feature representations. The proposed architecture also includes (2) a matching layer, named *cross-similarity function* (CS function), that calculates the matching score between the features of the set members across the two sets accurately. Our model guarantees both types of exchangeability in the modules. Figure 2 shows the proposed architecture.

We examine the set-to-set matching problem in a supervised setting, where examples of correctly paired sets are deployed as training data. The objective is to train the feature extractor and matching layer in an end-to-end manner such that the appropriate sets of features to be matched can be extracted. To train the model efficiently, we also propose a novel training framework, *K-pair-set loss*. Following training, the model is then used to find correct pairs of sets among a group of candidates.

The effectiveness of our approach is demonstrated in two real-world applications. First, we consider fashion set matching, where provided examples of the outfits are used as correct combinations of items (clothes). Using a large number of examples of the outfits in the form of images, we aim to match the correct pair of defined sets using the IQON dataset [51]. Since two positive sets include images of different fashion items, we regard this case as heterogeneous set matching. Next, we evaluate our methods through group re-id experiments using two datasets, a new extension of the Market-1501 dataset [87] (Market-1501 Group) and the Road Group dataset [77]. We use the Market-1501 Group dataset to analyze sensitivity to noises or outliers in set matching and the Road Group dataset as a more practical search task. Considering group membership change caused by the noises, we regard group re-id as a heterogeneous set matching problem. In the fashion set matching and group re-id experiments performed on the Market-1501 Group dataset, our methods show significant improvements compared with the results of baseline methods. Moreover, using the data augmentation method that we developed for the pair set (set-aug), our methods show competitive results without using any external datasets or spatial layout information on the Road Group dataset.

The main contributions of this paper are as follows. (i) A novel deep learning architecture is proposed to provide the two types of exchangeability required for set-to-set matching. (ii) The proposed feature extractors using the interactions between two sets are shown to extract better features for heterogeneous set

matching. (iii) A new loss function, K-pair-set loss, is proposed and provide better performances in our tasks. (iv) We introduce set-input methods into group re-id tasks (Road Group) using a new set-data augmentation, thereby showing competitive results without using external datasets or spatial relations. (v) The proposed models show state-of-the-art results for the fashion set matching and group re-id, supporting the claim that the interactions and exchangeability improve the accuracy and robustness of the set-matching procedure.

2 Preliminaries: Set-to-Set Matching

We introduce the necessary notation as follows. Let $\boldsymbol{x}_n, \boldsymbol{y}_m \in \mathfrak{X} = \mathbb{R}^d$ be feature vectors representing the features of each individual item. Let $\mathcal{X} = \{\boldsymbol{x}_1, ..., \boldsymbol{x}_N\}$ and $\mathcal{Y} = \{\boldsymbol{y}_1, ..., \boldsymbol{y}_M\}$ be *sets* of these feature vectors, where $\mathcal{X}, \mathcal{Y} \in 2^{\mathfrak{X}}$.

The function $f : 2^{\mathfrak{X}} \times 2^{\mathfrak{X}} \to \mathbb{R}$ calculates a matching score between the two sets $\mathcal{X}$ and $\mathcal{Y}$. Guaranteeing the exchangeability of the set-to-set matching requires that the matching function $f(\mathcal{X}, \mathcal{Y})$ is *symmetric* and *invariant* under any permutation of items within each set.

We consider tasks where the matching function f is used per pair of sets [92] to select a correct matching. Given candidate pairs of sets $(\mathcal{X}, \mathcal{Y}^{(k)})$, where $\mathcal{X}, \mathcal{Y}^{(k)} \in 2^{\mathfrak{X}}$ and $k \in \{1, \cdots, K\}$, we choose $\mathcal{Y}^{(k^*)}$ as a correct one so that $f(\mathcal{X}, \mathcal{Y}^{(k^*)})$ achieves the maximum score from amongst the K candidates. In this study, a supervised learning setting is considered, where the function f is trained to classify the correct pair and unmatched pairs.

2.1 Mappings of Exchangeability

We present a brief review on several notions of exchangeability, which are used in building our models.

Permutation Invariance. A set-input function f is said to be *permutation invariant* if

$$f(\mathcal{X}, \mathcal{Y}) = f(\pi_x \mathcal{X}, \pi_y \mathcal{Y}) \tag{1}$$

for permutations π_x on $\{1, \ldots, N\}$ and π_y on $\{1, \ldots, M\}$.

Permutation Equivariance. A map $f : \mathfrak{X}^N \times \mathfrak{X}^M \to \mathfrak{X}^N$ is said to be *permutation equivariant* if

$$f(\pi_x \mathcal{X}, \pi_y \mathcal{Y}) = \pi_x f(\mathcal{X}, \mathcal{Y}) \tag{2}$$

for permutations π_x and π_y, where π_x and π_y are on $\{1, \ldots, N\}$ and $\{1, \ldots, M\}$, respectively. Note that f is permutation invariant for permutations within $\mathcal{Y}$.

Symmetric Function. A map $f : 2^{\mathfrak{X}} \times 2^{\mathfrak{X}} \to \mathbb{R}$ is said to be *symmetric* if

$$f(\mathcal{X}, \mathcal{Y}) = f(\mathcal{Y}, \mathcal{X}). \tag{3}$$

Two-Set-Permutation Equivariance. Given $\mathcal{Z}^{(1)} \in \mathfrak{X}^N$ and $\mathcal{Z}^{(2)} \in \mathfrak{X}^M$, a map $f : \mathfrak{X}^* \times \mathfrak{X}^* \to \mathfrak{X}^* \times \mathfrak{X}^*$ is said to be *two-set-permutation equivariant* if

$$pf(\mathcal{Z}^{(1)}, \mathcal{Z}^{(2)}) = f(\mathcal{Z}^{(p(1))}, \mathcal{Z}^{(p(2))}) \tag{4}$$

for any permutation operator p exchanging the two sets, where $\mathfrak{X}^* = \cup_{n=0}^{\infty} \mathfrak{X}^n$ indicates a sequence of arbitrary length such as $\mathfrak{X}^N$ or $\mathfrak{X}^M$.
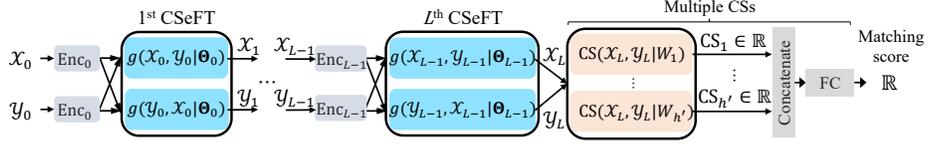
Fig. 2: Our model calculates a matching score between the paired sets. Enc_i, CSeFT, CS, and FC indicate an $(i+1)$-th (one-layered) encoder sharing weights within the same layer, cross-set feature transformation, cross-similarity function, and fully connected layer, respectively. We exclude the multihead structure in g.

3 Matching and Learning for Sets

3.1 Cross-Set Feature Transformation

We construct the architecture of the feature extractor, which transforms sets of features using the interactions between the pair of sets, and extracts the desired features to be matched in the post-processing stages (Section 3.2).

Here, consider the transformation of a pair of set-feature vectors $(\mathcal{X}, \mathcal{Y})$ into new feature representations on $\mathfrak{X}^N \times \mathfrak{X}^M$, using two-set-permutation equivariant functions. Let i be the iteration (layer) number of the CSeFT layers. Our feature extraction then can be described as a map of $(\mathcal{X}_i, \mathcal{Y}_i) \rightarrow (\mathcal{X}_{i+1}, \mathcal{Y}_{i+1})$, where $\mathcal{X}_{i+1}, \mathcal{X}_i \in \mathfrak{X}^N$, $\mathcal{Y}_{i+1}, \mathcal{Y}_i \in \mathfrak{X}^M$, $\mathcal{X}_{i+1} = (\boldsymbol{x}_{(n,i+1)})_{n=1}^N$, $\mathcal{X}_i = (\boldsymbol{x}_{(n,i)})_{n=1}^N$, $\mathcal{Y}_{i+1} = (\boldsymbol{y}_{(m,i+1)})_{m=1}^M$, and $\mathcal{Y}_i = (\boldsymbol{y}_{(m,i)})_{m=1}^M$. For example, $\boldsymbol{x}_{(n,i)} \in \mathfrak{X}$ denotes the feature vector extracted by the i-th layer representing the n-th item, $\boldsymbol{x}_n$, and $\boldsymbol{y}_{(m,i)}$ is defined similarly. Note that the initial feature vectors with $i = 0$ are found with a typical feature extractor, i.e., a deep convolutional neural network (CNN) for the image of each item. Then, we construct a parallel architecture of CSeFT, with an asymmetric transformation g, as follows:

$$\text{cross-set feature transformation (CSeFT)} : \begin{cases} \mathcal{X}_{i+1} = g(\mathcal{X}_i, \mathcal{Y}_i | \boldsymbol{\Theta}_i) \\ \mathcal{Y}_{i+1} = g(\mathcal{Y}_i, \mathcal{X}_i | \boldsymbol{\Theta}_i), \end{cases} \quad (5)$$

where $g : \mathfrak{X}^* \times \mathfrak{X}^* \rightarrow \mathfrak{X}^*$ is a permutation equivariant function and $\boldsymbol{\Theta}_i$ is learnable weights shared in the same layer. Also, we introduce the respective residual paths [17] to Eq. (5). Figure 3 shows the model of our CSeFT.

We propose two possible feature extractors for g: an *attention-based function*, and an *affinity-based function*. Both are constructed to assign the *matched* feature vectors to the *reference* feature vector, taking account of interactions between the two sets. For simplicity, we provide an explanation via the case of extracting the features for $\mathcal{X}$ as follows (we can easily exchange $\mathcal{X}$ and $\mathcal{Y}$ for $\mathcal{Y}$).

The attention-based function of g maps $\boldsymbol{x}_{(n,i)} \rightarrow \boldsymbol{x}_{(n,i+1)}$ as follows:

$$\boldsymbol{x}_{(n,i+1)} = \frac{1}{|\mathcal{Y}_i|} \sum_{\boldsymbol{y} \in \mathcal{Y}_i} \left(\frac{l_i^{(1)}(\boldsymbol{x}_{(n,i)})^{\mathrm{T}} l_i^{(2)}(\boldsymbol{y})}{\sqrt{d_g}} \right)_+ l_i^{(3)}(\boldsymbol{y}), \quad (6)$$
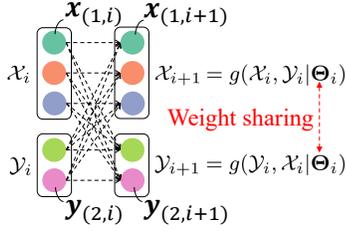
Fig. 3: A diagram of CSeFT. Here, we assume $|\mathcal{X}| = 3$ and $|\mathcal{Y}| = 2$. The colors indicate the respective set members.
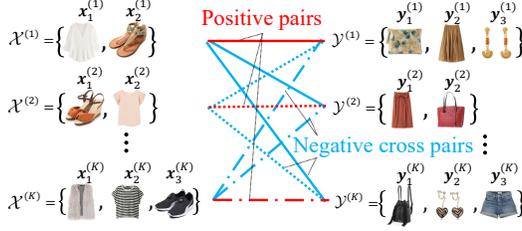
Fig. 4: K-pair-set-based matching candidates. Red and blue lines indicate correct pairs ($\mathcal{X}^{(k)}$, $\mathcal{Y}^{(k)}$) and negative cross pairs ($\mathcal{X}^{(k)}$, $\mathcal{Y}^{(k')}$) : $\forall k' \neq k$, where $k, k' \in \{1, \cdots, K\}$, respectively.

where $n \in \{1, \cdots, N\}$, $\boldsymbol{\Theta}_i = \{\Theta_i^{(1)}, \Theta_i^{(2)}, \Theta_i^{(3)}\}$, $\Theta_i^{(j)} \in \mathbb{R}^{d_g \times d}$, $|\mathcal{Y}_i| = M$, $l_i^{(j)}$ denote a linear transformation, i.e., $l_i^{(j)}(\boldsymbol{x}) := \Theta_i^{(j)} \boldsymbol{x}$, and $()_+$ is a non-negative mapping, i.e., ReLU [13], which introduces nonlinear interactions between the two elements.

Note that Eq. (6) relates to other attention models [31, 28, 81, 22, 70, 37], especially the dot-product attention [70, 37]. The dot-product attention has been introduced to calculate the weighted average on $\mathcal{Y}$ using softmax as the coefficients. However, the softmax operation would be inconsistent with our matching objective, as through normalization it increases the coefficients even in unmatched cases of $\mathcal{X}$ and $\mathcal{Y}$. To preserve non-linearity, we use the non-negative weighted sum instead and average it.

The affinity-based function of g maps $\boldsymbol{x}_{(n,i)} \to \boldsymbol{x}_{(n,i+1)}$ as follows:

$$\boldsymbol{x}_{(n,i+1)} = \frac{1}{2} \left(\bar{\boldsymbol{x}}_{(n,i)} + \frac{1}{|\bar{\mathcal{Y}}_i|} \sum_{\bar{\boldsymbol{y}} \in \bar{\mathcal{Y}}_i} \left(\frac{\bar{\boldsymbol{x}}_{(n,i)}^{\mathrm{T}} \bar{\boldsymbol{y}}}{\sqrt{d_g}} \right)_+ \bar{\boldsymbol{y}} \right), \qquad (7)$$

where $\boldsymbol{\Theta}_i = \{\Theta_i^{(1)}, \Theta_i^{(2)}\}$, $\bar{\boldsymbol{x}}_{(n,i)} = l_i^{(1)}(\boldsymbol{x}_{(n,i)})$, and $\bar{\mathcal{Y}}_i = \{l_i^{(2)}(\boldsymbol{y}_{(m,i)})\}_{m=1}^M$. Using the two linear transformations $l_i^{(1)}$ and $l_i^{(2)}$, the affinity-based function combines the resembling feature vectors within different sets so that the feature vectors for $\mathcal{X}$ have similar representations to the linearly transformed vectors in $\mathcal{Y}$.

Other simple permutation equivariant functions of g, e.g., $\boldsymbol{x}_{(n,i+1)} = \boldsymbol{x}_{(n,i)} + \frac{1}{|\mathcal{Y}_i|} \sum_{\boldsymbol{y} \in \mathcal{Y}_i} \boldsymbol{y}$, may be utilized. However, we consider it a function incapable of extracting appropriate enough features without any rich interactions between the two sets to yield accurate matching for two sets.

Instead of performing g singly, we introduce a multihead structure [70] to our feature extractor g, which is also a permutation equivariant function. Denoting the output of $g(\mathcal{X}_i, \mathcal{Y}_i|\boldsymbol{\Theta}_i^{(j)})$ as $g_{\mathcal{X}_i}^{(j)}$, the multihead version of g is defined as $\Theta_h \text{Concat}\left(g_{\mathcal{X}_i}^{(1)}, \cdots, g_{\mathcal{X}_i}^{(h)}\right)$, where Concat indicates a concatenation for each corresponding set member in $g_{\mathcal{X}_i}^{(j)}$, $\Theta_h \in \mathbb{R}^{d \times h d_g}$, and $h d_g = d$. Note that the

multihead structure is related to recent models such as MobileNet [20], which isolates and places the convolutional operations in parallel to reduce the calculation costs whilst preserving the accuracy of the recognition. We assume that the multihead structure provides various interactions between the set members, reducing the calculation costs as well.

Note that we can stack CSeFTs or combine it with other networks that operate upon the sets or items independently, which preserves the symmetric architecture. Although a function of our CSeFT does not entail interactions within a set, stacking CSeFTs takes account of higher-order interactions between multiple elements involving the intra-set, which is a similar way of overlaying convolution layers for CNNs. We discuss the overall architecture in Section 5.1.

3.2   Calculating Matching Score for Sets

We introduce a matching layer to calculate the matching score between two given sets, mapping $2^{\mathfrak{X}} \times 2^{\mathfrak{X}} \to [0, \infty]$. It is designed to calculate the inner product for every combination of set members across sets, so we call this *cross-similarity* (CS), defined as follows:

$$\mathrm{CS}(\mathcal{X}, \mathcal{Y}|W) := \frac{1}{|\mathcal{X}||\mathcal{Y}|} \sum_{\boldsymbol{x} \in \mathcal{X}} \sum_{\boldsymbol{y} \in \mathcal{Y}} \left(\frac{l(\boldsymbol{x}|W)^{\mathrm{T}} l(\boldsymbol{y}|W)}{\sqrt{d_w}} \right)_+ , \qquad (8)$$

where x and y are feature vectors in $\mathcal{X}$ and $\mathcal{Y}$, respectively, l is a linear function allowing conversions into a lower-dimensional space using learnable weights $W \in \mathbb{R}^{d_w \times d}$, i.e., $l(\boldsymbol{x}|W) := W\boldsymbol{x}$, and d_w is the number of dimensions of the lower-dimensional space. CS can be seen as a calculation of the average similarity in the linear subspaces created by the dimensionality reduction l, or the normalized and non-negative inner product if both sets contain only one set member.

While CS function is based on a simple pair relationship, considering the feature extraction provided by CSeFT layers that involves multiple elements to represent each feature, thereby higher-order relationships are included.

Instead of calculating CS singly, we utilize multiple CSs (mCS) to combine the CSs calculated with different linear mappings. The procedure runs as follows:

$$\mathrm{mCS}(\mathcal{X}, \mathcal{Y}|\mathbf{W}) = l(\mathrm{Concat}\,(\mathrm{CS}_1, \cdots, \mathrm{CS}_{h'})\,|W_o), \qquad (9)$$

where $\mathbf{W} = \{W_1, \cdots, W_{h'}, W_o\}$, $\mathrm{CS}_j = \mathrm{CS}(\mathcal{X}, \mathcal{Y}|W_j) \in \mathbb{R}$, the linear function l with learnable weights W_o maps $\mathbb{R}^{h'} \to \mathbb{R}$, and $h'd_w = d$.

Since CS and mCS are symmetric and permutation invariant functions, combined with the fact that CSeFT is a two-set-permutation equivariant function, our model is symmetric and invariant under any permutation of items within each set in these properties.

3.3   Training for Pairs of Sets

Next, the task of maximizing the matching score is translated into a minimization of a loss function of set matching, allowing for comparison against the scores

for other matching candidates. Although several studies have investigated loss functions for point-to-point [67, 48] or point-to-set [84, 90] metric learning, the loss function for set matching has not been well studied, and the efficient approach to preparing K candidates for each query set is non-trivial.

To train our model efficiently, we create matching candidates from the correct pairs, as described in Figure 4. Let $(\mathcal{X}^{(k)}, \mathcal{Y}^{(k)})$ be a correct pair of sets, where $k \in \{1, \cdots, K\}$. From those K-pair, by extracting all $\mathcal{Y}^{(k)}$, we create the set of $\mathcal{Y}^{(k)}$ as $\mathbf{Y} = \{\mathcal{Y}^{(1)}, \cdots, \mathcal{Y}^{(K)}\}$. That is, $\mathbf{Y}$ is composed of sets exhibiting correct relations to the respective $\mathcal{X}^{(k)}$, and $\mathbf{Y}$ can be used as a set of candidates for each $\mathcal{X}^{(k)}$ in the training stage. We construct positive pairs and negative cross pairs from these candidates by assuming that one correct pair exists for the respective sets, as described in Section 2. Then, we train our models using these pairs with a conventional softmax cross-entropy loss. We consider the above training method as a set version of N-pair loss [67], so we call this K-pair-set loss.

4   Related Works

Set-Input Neural Networks. Deep learning architectures for set data have been well studied [72, 37, 85, 55], and several studies have investigated its representation universality [82, 73, 60, 62]. In the work of Lee et al. [37], the state-of-the-art set-feature model, Set Transformer, was introduced by applying a self-attention based Transformer [70] to a set data. Set Transformer is trained through supervised/unsupervised learning and transforms a set data into a vector/matrix representation to recognize set features. However, constructing a deep learning model that can manage multiple sets has not been well studied.

Set-to-Set Matching. Various studies have suggested modeling a set as a hull [6, 23, 80, 92], hyperplanes [71, 12], linear subspace [79, 34, 76, 15], convex cone [66], covariance matrix [75, 4], Gaussian model [63, 1], among others, for matching sets. The methods above do not include feature learning schemes for paired sets, and most of them require specific computations involving optimization methods to measure the similarity/distance between the set models. Compared with these optimization-based methods, our models are based on a feed-forward function, thus are potentially easier to scale up.

Applications. Many fashion item recommendation studies have investigated natural combinations of fashion items, the so-called visual fashion compatibility, to recommend fashion items or outfits [16, 18, 21, 69, 39]. In this study, the main difficulties of the fashion set matching procedures lie in satisfying the fashion compatibility requirements of the matched sets.

In the applications of group re-id [41, 77, 40, 88, 4, 24, 91], problems of multiple instance matching arise. One group re-id scenario has been proposed that the detection of known groups from videos [40] is required. Also, two group re-id datasets, the Road Group dataset, and the DukeMTMC Group dataset[5] have been constructed [40], which include bounding box annotations for each person. Our experiments focus on set-to-set matching using these given cropped images.

---

[5] Note that the DukeMTMC [57] is no longer available.

Methods for Non-Exchangeable Data. Many powerful data-processing methods, such as graph matching [2, 14, 38, 3, 86, 10, 83], graph classification [47], entity matching [50], and sequence matching [64, 5] have been proposed based on the specific data structures. In natural language processing, Devlin et al. achieved state-of-the-art results in various tasks using the bidirectional encoder representations from transformers (BERT) [8]. Furthermore, Cucurull et al. applied graph neural networks (GNNs) to predict fashion compatibility between related fashion items using graph structures [7]. Although the data in those tasks are known to be non-exchangeable, we still consider that comparing these promising models with our model is possible and necessary.

5   Experiments

5.1   Overall Architecture

In this section, we briefly describe our models. Borrowing from the encoder–decoder structure [70, 37, 52], we construct our architecture by combining the encoder [37], called a self-attention block, with the decoder of our CSeFT. We apply a feed-forward network comprising two fully connected layers with a leaky ReLU [46] to the first argument of each function g. We then repeat this structure L times in succession, as described in Figure 2. We set h, h', L, and d to 8, 8, 2, and 512, respectively. To combine it with CNN features, for the fashion task, we use the Inception-v3 [68], which is pre-trained using the ILSVRC-2012 ImageNet [58], and finetune it. We extract the feature vectors on $\mathbb{R}^{2048}$ from the global average pooling layer and linearly transform it into $\mathbb{R}^{512}$. For the group re-id tasks, we utilize a simple CNN that maps $3 \to 64 \to 128 \to 256 \to 512$ channels using 3×3 kernels and train it from scratch.

5.2   Baselines for Comparisons

We validate our architecture through comparison with other set-matching models. However, to the best of our knowledge, studies using deep neural networks for matching two heterogeneous sets are non-existent. Instead, we use extensions from the state-of-the-art set-input method and the promising models in other related domains to a set-to-set matching procedure as described below, and consider this acceptable for the comparison. We also present ablation studies including other ordinary set matching functions.
Set Transformer. The Set Transformer [37] transforms a set of feature vectors into a vector on $\mathbb{R}^d$. Denoting the Set Transformer model ST, we perform the extension by calculating the matching score between the two sets $\mathcal{X}$ and $\mathcal{Y}$ via the inner product $\mathrm{ST}(\mathcal{X})^{\mathrm{T}}\mathrm{ST}(\mathcal{Y})$, sharing the weights between the two ST.
BERT. We consider a union of two sets as a set-input for the extension of BERT [8] and omit the individual token embedding, i.e., the position embedding. We use the segment embedding to designate items of $\mathcal{X}$ and $\mathcal{Y}$. We use three variants: $\mathrm{BERT}_{\mathrm{BASE}}$ is the same model as described in [8]; $\mathrm{BERT}_{\mathrm{BASE-AP}}$ uses

the average pooling in the last layer; and $\text{BERT}_{\text{SMALL}}$ is a four-layered version of $\text{BERT}_{\text{BASE}}$ with eight heads, and the hidden size is 512.

GNN. We combine two sets as one input for the extension of GNN [7]. Because this model is not presented to train in an end-to-end with the feature extractor, we do not finetune the CNN in fashion set matching, where pre-trained CNNs are used, but train it in an end-to-end manner for the group re-id task. Note that we omit the context provided from the external graphs in the evaluation stage to apply this model in the same scenarios of our tasks. We set the training epoch to 256 in the group re-id to enhance the training results of the GNN.

The properties in the above models are different from ours. The extension of Set Transformer satisfies the exchangeability criteria; however, no interactions between paired sets are provided. Both extensions of BERT and GNN provide the interactions but do not facilitate the exchangeability of two sets.

Additionally, in our first experiments, we introduce a conventional CNN, trained by Hard-Aware Point-to-Set loss (HAP2S) [84] as a minimum configuration. We use the exponential weighting and the same parameter setting described in [84]. We also use the *batch all* strategy [19] to train the CNN effectively.

5.3 Training Settings

In this section, we briefly describe the training settings. We use a stochastic gradient descent method with a learning rate of 0.005, a momentum of 0.5, and a weight decay of 0.00004. We set the numbers of matching candidates and the training epochs to 4 and 32, 16 and 128, and 81 and 3500, for the tasks of fashion set matching, Market-1501 Group dataset, and Road Group dataset, respectively. We train both the CNN and set-matching model in an end-to-end manner. In each iteration, we randomly swap pairs of sets and items in each set, and randomly flip images horizontally, to learn all the methods stably.

5.4 Fashion Set Matching

Dataset. We examine the set matching task for fashion recommendation, using the *IQON dataset* [51]. The dataset consists of recently created, high-quality outfits, including 199,792 items grouped into 88,674 outfits. We split these outfits into groups, using 70,997 for training, 8,842 for validation, and 8,835 for testing.

Our task can be considered an extended version of a standard task, Fill-In-The-Blank [7], which requires us to select an item that best extends an outfit from among four candidates. Because selecting a set corresponds to filling multiple blanks, we consider the set matching problem as Fill-In-The-N-Blank.

Preparing Set Pairs. To construct the correct pair of sets to be matched, we randomly halve the given outfit $\mathcal{O}$ into two non-empty proper subsets $\mathcal{X}$ and $\mathcal{Y}$ as follows: $\mathcal{O} \rightarrow \{\mathcal{X}, \mathcal{Y}\}$, where $\mathcal{X} \cap \mathcal{Y} = \emptyset$. Here, we extend this setting to include more general situations. We select Q outfits $\{\mathcal{O}^{(1)}, \cdots, \mathcal{O}^{(Q)}\}$ randomly and split the respective outfits in half $\mathcal{O}^{(q)} \rightarrow \{\mathcal{X}^{(q)}, \mathcal{Y}^{(q)}\}$, where $q \in \{1, \cdots, Q\}$. We regard the two sets $\{\mathcal{X}^{(1)}, \cdots, \mathcal{X}^{(Q)}\}$ and $\{\mathcal{Y}^{(1)}, \cdots, \mathcal{Y}^{(Q)}\}$ as the correct pair, which consists of Q fashion styles. In the training phase, we set $Q = 4$.

Table 1: Accuracy of fashion set matching (%). Cand and Mix indicate the number of matching candidates and number of mixed outfits (Q), respectively.

Method	Cand:4			Cand:8		
	Mix:1	Mix:2	Mix:4	Mix:1	Mix:2	Mix:4
Set Transformer	68.0	73.5	65.3	50.5	57.5	49.6
BERT$_{\mathrm{SMALL}}$	82.1	87.3	69.7	69.9	77.0	53.0
BERT$_{\mathrm{BASE}}$	81.4	86.6	66.1	69.2	76.3	50.8
BERT$_{\mathrm{BASE-AP}}$	80.8	86.4	65.4	68.6	75.7	49.5
GNN	35.4	32.4	25.5	19.9	17.5	13.4
HAP2S	39.3	36.6	32.0	23.3	20.8	17.8
Cross Attention (ours)	80.8	88.8	74.3	68.9	80.6	58.9
Cross Affinity (ours)	**85.1**	**90.6**	**75.9**	**73.8**	**82.8**	**61.9**

Fashion Set Matching. We discuss the experimental results of the fashion set matching. Table 1 shows significantly different results between our models and the baselines. Here, Cross Attention and Cross Affinity denote our models with the attention-based and affinity-based functions, respectively. Comparing the performance of Cross Affinity and BERT$_{\mathrm{SMALL}}$, which is the most accurate among the baselines, the differences in their accuracy were 5.2%, on average, 8.9%, at maximum, where the differences were relatively significant in the complicated setting on Mix:4. Furthermore, Table 1 shows that the affinity-based function performed better than the attention-based one.

In this experiment, we consider that the components on which the comparative effectiveness of the proposed models depended were potentially three-fold. Compared with the extensions of BERT, (a) our model preserves the exchangeability in two sets, which may ensure that the set features to be matched are accurately represented. Furthermore, (b) our model preserves two set features explicitly, whereas BERT provides a set of features with segment embedding that may have a limitation. Compared with the results of the Set Transformer, our models and BERT yielded accurate results is made possible by (c) providing the strength of interactions between two sets. Therefore, we conclude that these results justify the fine aspects of our architecture.

5.5   Group Re-Identification

We present the results of a group re-id on the Market-1501 Group dataset, a new extension of a well-known person re-id dataset, Market-1501 [87], and the Road Group dataset [77]. The task is to identify the pairs of sets that consist of individual images of the (mostly) same multiple persons under noisy situations. **Evaluation on Market-1501 Group Dataset.** We constructed the training/validation data based on query/gallery splits provided in [87]. Because there are few person images provided for each camera position, we did not consider camera information. We regard sets of gallery and query data as $\mathcal{X}$ and $\mathcal{Y}$, respectively, where each set contains three persons in non-noisy cases, and each person is represented by three different images.

We investigated noise robustness through the experiments to show that our models do not over-fit on the data; here, the *noise* means that random persons that accidentally contained into the group additionally or that the label noise [32] for paired sets generated based on the given noise fraction. Note that the noise persons and label noise have some relations, e.g., a candidate set composed of only noise persons corresponds to a set mislabelled by label noise.

Table 2 presents the comparison results. In the non-noisy case, many models showed almost perfect accuracies; we consider that *averaging feature vectors in sets* achieves high accuracy in this homogeneous case. In the case the noise person included, the noise ratio was inversely proportional to the accuracy across all the models; however, our models yielded more accurate results, e.g., the average accuracy of Cross Affinity, Set Transformer, and $BERT_{BASE-AP}$ was 87.0, 80.6, and 72.4%, respectively. Because the main differences between the architectures exist in the interactions for paired sets or the exchangeability, the results support the claim that considering these properties improves both the accuracy and robustness. Furthermore, in the case of label noise fraction is 0.8, the permutation invariance would be essential to preserve high accuracy.

Evaluation on Road Group Dataset. We conduct experiments on the Road Group dataset [77, 40], which consists of 162 group pairs taken from a 2-camera-view of a crowded road scene. One image per group for each camera is provided, where most groups do not have the same person's image in common with the different group pairs. Following the experimental protocol described in [77, 40], we construct training/validation datasets, splitting the 162 group pairs randomly in half into two different 81 group pairs, and reporting the accuracies calculated by the cumulative matching characteristic (CMC) metric [49].

Because group re-id is a newly emerging task, most datasets, including the Road Group dataset, contain a small number of groups and images, and training on such datasets is difficult [27]. Specifically, our set-to-set matching method extracts features that rely on input set pairs, thus, the variations in the set pairs are crucial. Considering the difference in appearances or camera parameters, however, importing external data [27, 26] is also a challenging task itself.

To relax the data limitation, we introduce our novel set-data augmentation (set-aug) method that significantly enhances the learning results of the proposed set-to-set matching modules by increasing the training data. Given positive person image pairs and several negative person images, creating set pairs randomly on each training iteration, our set-aug effectively increases the group member variations (please refer to supplementary materials for details).

Table 3 shows the experimental results. The top block in Table 3 indicates the results of our methods with three types of data augmentation: (a) the horizontal flipping [35], which is used to train the baseline model; (b) image-based data augmentation (img-aug), which includes both scale augmentation [65, 17] and random erasing [89] on images; and (c) our set-aug. Using the 81 pre-defined groups, the baseline model was not very effective, even with img-aug. However, using the set-aug, our method exhibited significant improvements without applying img-aug. These results imply that generating combinations on sets is very

Table 2: Accuracy (%) for Market-1501 Group. The number of candidates is 5.

Method	Non-noisy	Ratio of *noise* persons in $\mathcal{X} \times \mathcal{Y}$						Label noise frac.		
		$(\frac{0}{3},\frac{1}{4})$	$(\frac{1}{4},\frac{1}{4})$	$(\frac{0}{3},\frac{3}{6})$	$(\frac{0}{3},\frac{5}{8})$	$(\frac{3}{6},\frac{3}{6})$	$(\frac{5}{8},\frac{5}{8})$	0.2	0.4	0.8
Set Transformer	99.5	95.1	89.9	85.7	80.4	65.7	48.1	**99.3**	98.8	95.6
BERT$_{\text{SMALL}}$	94.3	77.6	69.2	83.7	64.9	49.5	24.7	99.2	98.7	79.5
BERT$_{\text{BASE}}$	96.8	80.5	77.6	68.8	69.9	61.9	49.2	98.9	98.1	76.0
BERT$_{\text{BASE}-\text{AP}}$	97.3	84.4	74.7	70.7	69.3	62.8	47.7	**99.3**	97.5	77.9
GNN	82.0	29.3	46.0	23.7	22.1	29.3	21.1	81.7	73.0	76.7
Cross Attention (ours)	99.6	**96.9**	**94.8**	91.9	90.7	**72.9**	56.1	**99.3**	99.6	95.5
Cross Affinity (ours)	**99.7**	96.5	92.5	**94.4**	**92.4**	72.0	**61.7**	**99.3**	**99.9**	**98.4**

Table 3: Evaluation results (%) for Road Group dataset.

Method (detector-based)	CMC-1	CMC-5	CMC-10	CMC-15	CMC-20
Data augmentation ablation					
Cross Affinity (our baseline)	45.2 ± 3.5	77.5 ± 2.9	87.9 ± 3.8	91.9 ± 2.4	94.1 ± 2.1
Baseline + img-aug	47.7 ± 4.2	78.3 ± 3.2	87.7 ± 2.6	91.1 ± 2.4	93.3 ± 1.8
Baseline + set-aug	$\mathbf{84.0} \pm 3.6$	93.8 ± 0.8	$\mathbf{96.8} \pm 0.6$	$\mathbf{97.0} \pm 1.0$	97.5 ± 1.1
Baseline + set-aug + img-aug	81.7 ± 1.9	$\mathbf{94.1} \pm 1.3$	96.5 ± 1.1	$\mathbf{97.0} \pm 0.9$	$\mathbf{97.8} \pm 0.8$
Baseline + set-aug (ours)	$\mathbf{84.0} \pm 3.6$	93.8 ± 0.8	$\mathbf{96.8} \pm 0.6$	97.0 ± 1.0	97.5 ± 1.1
MGM w/ spatial layout [40]	80.2	93.8	96.3	**97.5**	97.5
MGM w/o spatial layout [40]	70.4	90.1	91.3	92.6	96.3
TSCN w/ external data [27]	**84.0**	**95.1**	96.3	-	**98.8**
GNN w/ external data [26]	74.1	90.1	92.6	-	**98.8**
Method (GT-based)	CMC-1	CMC-5	CMC-10	CMC-15	CMC-20
Baseline + set-aug (ours)	$\mathbf{85.7} \pm 3.7$	$\mathbf{96.3} \pm 0.8$	$\mathbf{97.8} \pm 0.5$	$\mathbf{98.3} \pm 0.6$	$\mathbf{98.3} \pm 0.6$
MGM w/ spatial layout [40]	82.4	95.1	96.3	97.5	98.0

beneficial to our models. The other parts in Table 3 show that our methods yield very competitive results, compared with the state-of-the-art methods that utilize a large transferred external dataset or auxiliary features such as spatial layout information within each group. Furthermore, compared with MGM w/o spatial layout [40], which also does not use the spatial layout information, our methods significantly improved the accuracy of CMC-1 by 13.6%.

5.6 Ablation Study

In this section, we report the results of an ablation study performed to highlight the importance of each proposed component. The top part in Table 4 shows the two results obtained when our models are trained using triplet loss with the soft-margin and batch all strategy [19] or the proposed K-pair-set loss. Triplet loss triggered significant accuracy degradation, even though the losses converged to near zero in the training stages. On the other hand, the proposed K-pair-set loss can manage to accurately train the models by considering the loss of selecting paired sets among multiple candidates. The second topmost part in Table 4 shows the results of ablations in the feature extractor. Reducing the number of layers and number of multiheads in the CSeFT, and excluding the encoder and CSeFT, the accuracies are degraded by 1.0, 1.1, 1.3, and 4.2%. Our model without the

encoder performed well (1.3% degradation) but showed a somewhat slow convergence. However, excluding the CSeFT module significantly reduced the accuracy (4.2% degradation). These results imply that the proposed CSeFT module is a crucial part of the set-to-set matching model architecture. The second lower part of Table 4 shows the results of ablation study performed on the matching layer. Reducing mCS to a single CS and excluding ReLU from the CS functions reduced the accuracies of both models by 1.0 and 2.0%, respectively. It is interesting to observe that the ReLU was more important than the number of CS functions; this demonstrated the importance of nonlinearity in the matching layer. Furthermore, replacing our mCS with max pooling, average pooling, projection metric [25], covariance matrix [75, 4], set kernel [33], and cosine similarity metric [53] all resulted in significant accuracy degradation implying the effectiveness of our mCS functions (see supplementary materials for details). The lowermost part of Table 4 shows the results of ablation study performed on the feature extractor and matching layer. The extension of Set Transformer, which does not include the proposed CSeFT module and CS function, yielded significant accuracy degradation. These results show the validity of our architecture for heterogeneous set-to-set matching.

Table 4: Ablation study. Average accuracies (%) on Market-1501 Group are shown, where the non-noisy and six noise person patterns, presented in Table 2, are included.

Method	Accuracy
Training method ablation	
Cross Affinity (baseline)	**87.0**
Baseline with triplet loss	45.5
Feature extractor ablation	
Baseline with L=1	86.0
Baseline with h=1	85.9
w/o Enc	85.7
w/o CSeFT	82.8
Matching layer ablation	
Single CS	86.0
w/o ReLU in mCS	85.0
Max pooling	86.1
Average pooling	85.8
Projection metric	67.0
Covariance matrix	61.1
Set kernel	53.3
Cosine similarity metric	53.1
Feature & matching layer ablation	
Set Transformer	80.6

6   Conclusion

In this study, we investigated the heterogeneous set-to-set matching problem. We proposed a novel architecture comprising the (1) cross-set feature transformation (CSeFT) module and (2) cross-similarity (CS) function, in addition to a loss function and set-data augmentation for performing set-to-set matching.

We showed that our architecture preserves the two types of exchangeability for a pair of sets and also the items within them, thereby satisfying the requirements of set-to-set matching procedure.

We demonstrated that our models performed well compared with the state-of-the-art methods and baselines in the fashion set matching and group re-id experiments. Furthermore, we validated our proposed architecture through the ablation study. These results support the claim that the exchangeability and the feature representations extracted with interactions between the two sets improve the accuracy and robustness of the heterogeneous set-to-set matching.

References

1. Arandjelovic, O., Shakhnarovich, G., Fisher, J., Cipolla, R., Darrell, T.: Face recognition with image sets using manifold density divergence. In: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05). vol. 1, pp. 581–588. IEEE (2005)
2. Bai, Y., Ding, H., Bian, S., Chen, T., Sun, Y., Wang, W.: Simgnn: A neural network approach to fast graph similarity computation. In: Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining. pp. 384–392 (2019)
3. Bai, Y., Ding, H., Sun, Y., Wang, W.: Convolutional set matching for graph similarity. arXiv preprint arXiv:1810.10866 (2018)
4. Cai, Y., Takala, V., Pietikainen, M.: Matching groups of people by covariance descriptor. In: 2010 20th International Conference on Pattern Recognition. pp. 2744–2747. IEEE (2010)
5. Caspi, Y., Simakov, D., Irani, M.: Feature-based sequence-to-sequence matching. International Journal of Computer Vision **68**(1), 53–64 (2006)
6. Cevikalp, H., Triggs, B.: Face recognition based on image sets. In: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. pp. 2567–2573. IEEE (2010)
7. Cucurull, G., Taslakian, P., Vazquez, D.: Context-aware visual compatibility prediction. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 12617–12626 (2019)
8. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
9. Feng, J., Karaman, S., Chang, S.F.: Deep image set hashing. In: 2017 IEEE Winter Conference on Applications of Computer Vision (WACV). pp. 1241–1250. IEEE (2017)
10. Fey, M., Lenssen, J.E., Morris, C., Masci, J., Kriege, N.M.: Deep graph matching consensus. arXiv preprint arXiv:2001.09621 (2020)
11. Gao, Z., Wang, D., He, X., Zhang, H.: Group-pair convolutional neural networks for multi-view based 3d object retrieval. In: Thirty-Second AAAI Conference on Artificial Intelligence (2018)
12. Gionis, A., Indyk, P., Motwani, R., et al.: Similarity search in high dimensions via hashing. In: Vldb. vol. 99, pp. 518–529 (1999)
13. Glorot, X., Bordes, A., Bengio, Y.: Deep sparse rectifier neural networks. In: Proceedings of the fourteenth international conference on artificial intelligence and statistics. pp. 315–323 (2011)
14. Guo, M., Chou, E., Huang, D.A., Song, S., Yeung, S., Fei-Fei, L.: Neural graph matching networks for fewshot 3d action recognition. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 653–669 (2018)
15. Hamm, J., Lee, D.D.: Grassmann discriminant analysis: a unifying view on subspace-based learning. In: Proceedings of the 25th international conference on Machine learning. pp. 376–383 (2008)
16. Han, X., Wu, Z., Jiang, Y.G., Davis, L.S.: Learning fashion compatibility with bidirectional lstms. In: Proceedings of the 25th ACM international conference on Multimedia. pp. 1078–1086. ACM (2017)
17. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)

18. He, R., Packer, C., McAuley, J.: Learning compatibility across categories for heterogeneous item recommendation. In: 2016 IEEE 16th International Conference on Data Mining (ICDM). pp. 937–942. IEEE (2016)

19. Hermans, A., Beyer, L., Leibe, B.: In defense of the triplet loss for person re-identification. CoRR **abs/1703.07737** (2017), http://arxiv.org/abs/1703.07737

20. Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H.: Mobilenets: Efficient convolutional neural networks for mobile vision applications. CoRR **abs/1704.04861** (2017), http://arxiv.org/abs/1704.04861

21. Hsiao, W.L., Grauman, K.: Creating capsule wardrobes from fashion images. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 7161–7170 (2018)

22. Hu, J., Shen, L., Sun, G.: Squeeze-and-excitation networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 7132–7141 (2018)

23. Hu, Y., Mian, A.S., Owens, R.: Sparse approximated nearest points for image set classification. In: CVPR 2011. pp. 121–128. IEEE (2011)

24. Huang, Z., Wang, Z., Hung, T., Satoh, S., Lin, C.: Group re-identification via transferred representation and adaptive fusion. In: 2019 IEEE Fifth International Conference on Multimedia Big Data (BigMM). pp. 128–132 (Sep 2019). https://doi.org/10.1109/BigMM.2019.00-34

25. Huang, Z., Wu, J., Van Gool, L.: Building deep networks on grassmann manifolds. In: Thirty-Second AAAI Conference on Artificial Intelligence (2018)

26. Huang, Z., Wang, Z., Hu, W., Lin, C.W., Satoh, S.: Dot-gnn: Domain-transferred graph neural network for group re-identification. In: Proceedings of the 27th ACM International Conference on Multimedia. pp. 1888–1896 (2019)

27. Huang, Z., Wang, Z., Satoh, S., Lin, C.W.: Group re-identification via transferred single and couple representation learning. arXiv preprint arXiv:1905.04854 (2019)

28. Ilse, M., Tomczak, J.M., Welling, M.: Attention-based deep multiple instance learning. arXiv preprint arXiv:1802.04712 (2018)

29. Iwata, T., Kanagawa, M., Hirao, T., Fukumizu, K.: Unsupervised group matching with application to cross-lingual topic matching without alignment information. Data mining and knowledge discovery **31**(2), 350–370 (2017)

30. Iwata, T., Lloyd, J.R., Ghahramani, Z.: Unsupervised many-to-many object matching for relational data. IEEE transactions on pattern analysis and machine intelligence **38**(3), 607–617 (2015)

31. Jain, S., Wallace, B.C.: Attention is not explanation (2019)

32. Jiang, L., Zhou, Z., Leung, T., Li, L., Fei-Fei, L.: Mentornet: Regularizing very deep neural networks on corrupted labels. CoRR **abs/1712.05055** (2017), http://arxiv.org/abs/1712.05055

33. Kim, J., McCourt, M., You, T., Kim, S., Choi, S.: Practical bayesian optimization over sets (2019)

34. Kim, T.K., Kittler, J., Cipolla, R.: Discriminative learning and recognition of image set classes using canonical correlations. IEEE Transactions on Pattern Analysis and Machine Intelligence **29**(6), 1005–1018 (2007)

35. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems. pp. 1097–1105 (2012)

36. Le, D.T., Lauw, H.W., Fang, Y.: Correlation-sensitive next-basket recommendation. In: Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19. pp. 2808–2814. International Joint Conferences on Ar-

tificial Intelligence Organization (7 2019). https://doi.org/10.24963/ijcai.2019/389, https://doi.org/10.24963/ijcai.2019/389

37. Lee, J., Lee, Y., Kim, J., Kosiorek, A., Choi, S., Teh, Y.W.: Set transformer: A framework for attention-based permutation-invariant neural networks. In: International Conference on Machine Learning. pp. 3744–3753 (2019)

38. Li, Y., Gu, C., Dullien, T., Vinyals, O., Kohli, P.: Graph matching networks for learning the similarity of graph structured objects. arXiv preprint arXiv:1904.12787 (2019)

39. Li, Y., Cao, L., Zhu, J., Luo, J.: Mining fashion outfit composition using an end-to-end deep learning approach on set data. CoRR **abs/1608.03016** (2016), http://arxiv.org/abs/1608.03016

40. Lin, W., Li, Y., Xiao, H., See, J., Zou, J., Xiong, H., Wang, J., Mei, T.: Group reidentification with multigrained matching and integration. IEEE transactions on cybernetics (2019)

41. Lisanti, G., Martinel, N., Bimbo, A.D., Foresti, G.L.: Group re-identification via unsupervised transfer of sparse features encoding. CoRR **abs/1707.09173** (2017), http://arxiv.org/abs/1707.09173

42. Liu, D., Liang, C., Zhang, Z., Qi, L., Lovell, B.C.: Exploring inter-instance relationships within the query set for robust image set matching. Sensors **19**(22), 5051 (2019)

43. Liu, X., Guo, Z., Li, S., Kong, L., Jia, P., You, J., Kumar, B.: Permutation-invariant feature restructuring for correlation-aware image set-based recognition. arXiv preprint arXiv:1908.01174 (2019)

44. Liu, Y., Yan, J., Ouyang, W.: Quality aware network for set to set recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5790–5799 (2017)

45. Lu, J., Wang, G., Deng, W., Moulin, P., Zhou, J.: Multi-manifold deep metric learning for image set classification. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1137–1145 (2015)

46. Maas, A.L., Hannun, A.Y., Ng, A.Y.: Rectifier nonlinearities improve neural network acoustic models. In: Proc. icml. vol. 30, p. 3 (2013)

47. Maron, H., Ben-Hamu, H., Shamir, N., Lipman, Y.: Invariant and equivariant graph networks. arXiv preprint arXiv:1812.09902 (2018)

48. Mishchuk, A., Mishkin, D., Radenovic, F., Matas, J.: Working hard to know your neighbor's margins: Local descriptor learning loss. In: Advances in Neural Information Processing Systems. pp. 4826–4837 (2017)

49. Moon, H., Phillips, P.J.: Computational and performance aspects of pca-based face-recognition algorithms. Perception **30**(3), 303–321 (2001)

50. Mudgal, S., Li, H., Rekatsinas, T., Doan, A., Park, Y., Krishnan, G., Deep, R., Arcaute, E., Raghavendra, V.: Deep learning for entity matching: A design space exploration. In: Proceedings of the 2018 International Conference on Management of Data. pp. 19–34 (2018)

51. Nakamura, T., Goto, R.: Outfit generation and style extraction via bidirectional LSTM and autoencoder. CoRR **abs/1807.03133** (2018), http://arxiv.org/abs/1807.03133

52. Newell, A., Yang, K., Deng, J.: Stacked hourglass networks for human pose estimation. CoRR **abs/1603.06937** (2016), http://arxiv.org/abs/1603.06937

53. Nguyen, H.V., Bai, L.: Cosine similarity metric learning for face verification. In: Asian conference on computer vision. pp. 709–720. Springer (2010)

54. Parkhi, O.M., Vedaldi, A., Zisserman, A., et al.: Deep face recognition. In: bmvc. vol. 1, p. 6 (2015)

55. Qi, C.R., Su, H., Mo, K., Guibas, L.J.: Pointnet: Deep learning on point sets for 3d classification and segmentation. CoRR **abs/1612.00593** (2016), http://arxiv.org/abs/1612.00593
56. Rendle, S., Freudenthaler, C., Schmidt-Thieme, L.: Factorizing personalized markov chains for next-basket recommendation. In: Proceedings of the 19th international conference on World wide web. pp. 811–820. ACM (2010)
57. Ristani, E., Solera, F., Zou, R.S., Cucchiara, R., Tomasi, C.: Performance measures and a data set for multi-target, multi-camera tracking (2016)
58. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al.: Imagenet large scale visual recognition challenge. International journal of computer vision **115**(3), 211–252 (2015)
59. Saito, Y., Hong, P.K., Niihara, T., Miyamoto, H., Fukumizu, K.: Data-driven taxonomy matching of asteroid and meteorite. Meteoritics & Planetary Science **55**(1), 193–206 (2020). https://doi.org/10.1111/maps.13428, https://onlinelibrary.wiley.com/doi/abs/10.1111/maps.13428
60. Sannai, A., Takai, Y., Cordonnier, M.: Universal approximations of permutation invariant/equivariant functions by deep neural networks. arXiv preprint arXiv:1903.01939 (2019)
61. Sarwar, B.M., Karypis, G., Konstan, J.A., Riedl, J., et al.: Item-based collaborative filtering recommendation algorithms. Www **1**, 285–295 (2001)
62. Segol, N., Lipman, Y.: On universal equivariant set networks. arXiv preprint arXiv:1910.02421 (2019)
63. Shakhnarovich, G., Fisher, J.W., Darrell, T.: Face recognition from long-term observations. In: European Conference on Computer Vision. pp. 851–865. Springer (2002)
64. Si, J., Zhang, H., Li, C.G., Kuen, J., Kong, X., Kot, A.C., Wang, G.: Dual attention matching network for context-aware feature sequence based person re-identification (2018)
65. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
66. Sogi, N., Nakayama, T., Fukui, K.: A method based on convex cone model for image-set classification with cnn features. In: 2018 International Joint Conference on Neural Networks (IJCNN). pp. 1–8. IEEE (2018)
67. Sohn, K.: Improved deep metric learning with multi-class n-pair loss objective. In: Lee, D.D., Sugiyama, M., Luxburg, U.V., Guyon, I., Garnett, R. (eds.) Advances in Neural Information Processing Systems 29, pp. 1857–1865. Curran Associates, Inc. (2016), http://papers.nips.cc/paper/6200-improved-deep-metric-learning-with-multi-class-n-pair-loss-objective.pdf
68. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2818–2826 (2016)
69. Vasileva, M.I., Plummer, B.A., Dusad, K., Rajpal, S., Kumar, R., Forsyth, D.A.: Learning type-aware embeddings for fashion compatibility. CoRR **abs/1803.09196** (2018), http://arxiv.org/abs/1803.09196
70. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. CoRR **abs/1706.03762** (2017), http://arxiv.org/abs/1706.03762
71. Vincent, P., Bengio, Y.: K-local hyperplane and convex distance nearest neighbor algorithms. In: Advances in neural information processing systems. pp. 985–992 (2002)

72. Vinyals, O., Bengio, S., Kudlur, M.: Order matters: Sequence to sequence for sets. arXiv preprint arXiv:1511.06391 (2015)
73. Wagstaff, E., Fuchs, F.B., Engelcke, M., Posner, I., Osborne, M.: On the limitations of representing functions on sets. arXiv preprint arXiv:1901.09006 (2019)
74. Wang, J., Song, Y., Leung, T., Rosenberg, C., Wang, J., Philbin, J., Chen, B., Wu, Y.: Learning fine-grained image similarity with deep ranking. CoRR **abs/1404.4661** (2014), http://arxiv.org/abs/1404.4661
75. Wang, R., Guo, H., Davis, L.S., Dai, Q.: Covariance discriminative learning: A natural and efficient approach to image set classification. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition. pp. 2496–2503. IEEE (2012)
76. Wang, R., Shan, S., Chen, X., Gao, W.: Manifold-manifold distance with application to face recognition based on image set. In: 2008 IEEE Conference on Computer Vision and Pattern Recognition. pp. 1–8. IEEE (2008)
77. Xiao, H., Lin, W., Sheng, B., Lu, K., Yan, J., Wang, J., Ding, E., Zhang, Y., Xiong, H.: Group re-identification: Leveraging and integrating multi-grain information. In: Proceedings of the 26th ACM International Conference on Multimedia. pp. 192–200. MM '18, ACM, New York, NY, USA (2018). https://doi.org/10.1145/3240508.3240539, http://doi.acm.org/10.1145/3240508.3240539
78. Xie, W., Shen, L., Zisserman, A.: Comparator networks. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 782–797 (2018)
79. Yamaguchi, O., Fukui, K., Maeda, K.i.: Face recognition using temporal image sequence. In: Proceedings Third IEEE International Conference on Automatic Face and Gesture Recognition. pp. 318–323. IEEE (1998)
80. Yang, M., Zhu, P., Van Gool, L., Zhang, L.: Face recognition based on regularized nearest points between image sets. In: 2013 10th IEEE international conference and workshops on automatic face and gesture recognition (FG). pp. 1–7. IEEE (2013)
81. Yang, Z., Yang, D., Dyer, C., He, X., Smola, A., Hovy, E.: Hierarchical attention networks for document classification. In: Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies. pp. 1480–1489 (2016)
82. Yarotsky, D.: Universal approximations of invariant maps by neural networks. arXiv preprint arXiv:1804.10306 (2018)
83. Yoshida, T., Takeuchi, I., Karasuyama, M.: Learning interpretable metric between graphs: Convex formulation and computation with graph mining. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. p. 10261036. KDD 19, Association for Computing Machinery, New York, NY, USA (2019). https://doi.org/10.1145/3292500.3330845, https://doi.org/10.1145/3292500.3330845
84. Yu, R., Dou, Z., Bai, S., Zhang, Z., Xu, Y., Bai, X.: Hard-aware point-to-set deep metric for person re-identification. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 188–204 (2018)
85. Zaheer, M., Kottur, S., Ravanbakhsh, S., Póczos, B., Salakhutdinov, R., Smola, A.J.: Deep sets. CoRR **abs/1703.06114** (2017), http://arxiv.org/abs/1703.06114
86. Zanfir, A., Sminchisescu, C.: Deep learning of graph matching. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2684–2693 (2018)
87. Zheng, L., Shen, L., Tian, L., Wang, S., Wang, J., Tian, Q.: Scalable person re-identification: A benchmark. In: Proceedings of the IEEE International Conference on Computer Vision (2015)

88. Zheng, W.S., Gong, S., Xiang, T.: Associating groups of people. In: BMVC. vol. 2 (2009)
89. Zhong, Z., Zheng, L., Kang, G., Li, S., Yang, Y.: Random erasing data augmentation. arXiv preprint arXiv:1708.04896 (2017)
90. Zhou, S., Wang, J., Wang, J., Gong, Y., Zheng, N.: Point to set similarity based deep feature learning for person re-identification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3741–3750 (2017)
91. Zhu, F., Chu, Q., Yu, N.: Consistent matching based on boosted salience channels for group re-identification. In: 2016 IEEE International Conference on Image Processing (ICIP). pp. 4279–4283. IEEE (2016)
92. Zhu, P., Zhang, L., Zuo, W., Zhang, D.: From point to set: Extend the learning of distance metrics. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 2664–2671 (2013)