Beyond Triplet Loss: Meta Prototypical N-tuple Loss for Person Re-identification

Zhizheng Zhang, Cuiling Lan, *Member, IEEE,* Wenjun Zeng, *Fellow, IEEE,*
Zhibo Chen, *Senior Member, IEEE,* and Shih-Fu Chang, *Fellow, IEEE*

***Abstract*—Person Re-identification (ReID) aims at matching a person of interest across images. In convolutional neural network (CNN) based approaches, loss design plays a vital role in pulling closer features of the same identity and pushing far apart features of different identities. In recent years, triplet loss achieves superior performance and is predominant in ReID. However, triplet loss considers only three instances of two classes in per-query optimization (with an anchor sample as query) and it is actually equivalent to a two-class classification. There is a lack of loss design which enables the joint optimization of multiple instances (of multiple classes) within per-query optimization for person ReID. In this paper, we introduce a multi-class classification loss, *i.e.*, N-tuple loss, to jointly consider multiple (N) instances for per-query optimization. This in fact aligns better with the ReID test/inference process, which conducts the ranking/comparisons among multiple instances. Furthermore, for more efficient multi-class classification, we propose a new meta prototypical N-tuple loss. With the multi-class classification incorporated, our model achieves the state-of-the-art performance on the benchmark person ReID datasets.**

***Index Terms*—person re-identification, loss design, metric learning**

I. Introduction

PERSON re-identification (ReID) aims to identify the same persons across images captured at different times, or places, or from different cameras. It has drawn a lot of attention from both academia and industry. The objective of CNN-based person ReID methods is to minimize the feature discrepancies (distances) among the samples with the same identity while maximizing the feature discrepancies (distances) among the samples of different identities to encourage the separation of positive pairs and negative pairs.

Person ReID lies in between image classification [1], [2], where each identity is treated as one class in the training, and instance retrieval [3]. The identities are available during training while the identities of test images are previously "unseen" [4]. The person ReID test can be considered as a retrieval process. Given a query image, its distances (or similarities) to all the samples in the gallery set will be calculated and ranked to identify the matched images. Given a person sample as the anchor, triplet loss optimizes its distance to one sample of the same identity to be closer than its distance to another sample of a different identity. There are also many works [5], [6], [7], [8], that use the conventional classification loss for feature learning, casting person ReID as a classification problem over all identities in the training set (with each identity taken as a category). Recently, many state-of-the-art works combine both triplet loss and classification loss, which leads to a superior performance to those using only one of them [9], [10], [11], [12], [13].

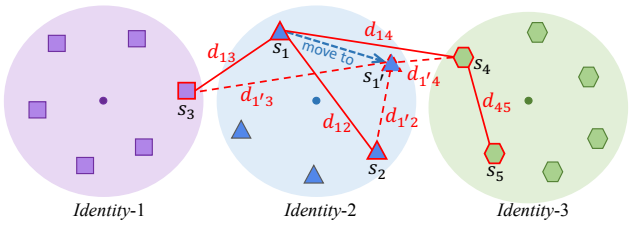


Fig. 1. A toy example of samples of three different identities identified by different shapes (rectangle, triangle or hexagon) in the embedding space. The solid circles denote the identity centers. The samples (s_1, s_2, s_3) constitute a triplet wherein s_1 is the anchor, s_2 is the positive sample (that has the same ID with the anchor), and s_3 is the negative sample (that has a different ID). Similarly, (s_4, s_5, s_1) constitute another triplet.

Fig. 1 illustrates the roles triplet loss and classification loss play in optimization and also reveals the limitation of the triplet loss. The conventional classification loss optimizes towards pulling samples closer to their corresponding class centers (*i.e.*, the coefficients of the fully connected layer of the classifier) and pushing them farther apart from other class centers. This encourages the separation of class centers, which enables a global-wise optimization but does not ensure the reasonableness of the relative order of different instances. As the examples shown in Fig. 1, although all the samples can be correctly classified with a conventional classifier (that is implemented by network parameters), the distance between sample s_1 and s_2 (of the same identity) is still larger than the distance between s_1 and s_3 (of different identities). This contradicts ReID inference which performs ranking among instances.

The triplet loss directly optimizes the relative order of three samples/instances from two identities. As illustrated in Fig. 1, a triplet consists of an anchor sample (*e.g.*, s_1), a positive sample (*e.g.*, s_2) that has the same identity as the anchor, and a negative sample with different identity (*e.g.*, s_3). The triplet loss aims to reduce the distance (d_{12}) of the positive pair and enlarge the distance of the negative pair (d_{13}) to

Zhizheng Zhang and Zhibo Chen are with the CAS Key Laboratory of Technology in Geo-Spatial Information Processing and Application System, University of Science and Technology of China, Hefei, Anhui, P.R. China. (E-mail: zhizheng@mail.ustc.edu.cn, chenzhibo@ustc.edu.cn)
Cuiling Lan and Wenjun Zeng are with Microsoft Research Asia, Beijing, P.R. China. (E-mail: culan, wezeng@microsoft.com)
Shih-Fu Chang is with Columbia University, New York, NY. (E-mail: sc250@columbia.edu)

make d_{13} larger than d_{12} in the embedding space. However, when we look into two triplets (s_1, s_2, s_3) and (s_4, s_5, s_1) respectively, their optimization directions may contradict each other and thus render inferior results. For example, sample s_1 may move to a new position $s_1^{'}$ as a result of optimizing for the first triplet. However, due to lack of efficient interaction with the second triplet, the optimized new position $s_1^{'}$ would make the second triplet worse. For the anchor s_4, the distance with the negative sample s_1 becomes smaller. Thus, considering multiple instances (*e.g.*, 5 samples) through several (*e.g.*, 2) independent triplets is hard to provide powerful constraints.

To address the above limitation of triplet loss for person ReID, we rethink the loss designs and propose to exploit N-tuple loss to jointly optimize multiple instances (from multiple classes) within a per-query optimization. This enables the joint comparison across multi-class instances which better matches the nature of ReID inference, *i.e.*, distance-based ranking across many images in the gallery set. We validate the effectiveness of the N-tuple loss on several benchmark datasets. Moreover, we propose a lifted variant of N-tuple loss, meta prototypical N-tuple (MPN-tuple) loss for more efficient multi-class classification. In the MPN-tuple loss, we employ a meta-predictor to learn the category-specific prototype from instances as the classifier for multi-class classification, which is optimized in an end-to-end manner.

We summarize our main contributions as follows:

- We introduce the N-tuple loss, under a unified loss formulation, to person ReID, which jointly optimizes over multiple instances from multiple classes for each query sample. This in fact aligns better with the ReID test/inference process but is under-explored in previous ReID works.
- We propose a variant of N-tuple loss, *i.e.*, meta prototypical N-tuple (MPN-tuple) loss, towards more efficient multi-class classification optimization for person ReID.
- As a minor contribution, we conduct a systematic empirical study by revisiting the design choices of both triplet loss and conventional classification loss as well as their combinations. We use the best practices to serve as our baseline.

While simple, our scheme achieves the state-of-the-art performance, outperforming those of previous loss designs by a large margin. We hope our scheme could serve as a new strong baseline which benefits the ReID community in the future.

II. Related Works

Person Re-identification. Many efforts have been made for representative feature learning from the perspectives of network or loss designs for person ReID.

Some approaches exploit multi-granularity feature representation to capture both global and local features for person ReID [14], [9], [5], [15], [16], [17]. Some others introduce attention mechanisms for discriminative feature learning, [18], [19], [11], [20], [21]. To tackle the challenges of diverse viewpoints and poses, many works exploit some auxiliary semantics (*e.g.*segmentation [21], human parsing [22], pose [16], [17], dense semantics [10], [23]) to address the misalignment problem in person ReID.

For loss designs in person ReID, the classification loss is widely used, with the total number of classes being the number of identities in the training set [5], [6], [7], [8]. In the early years, some works employ the contrastive loss [24], [25], [26] and verification loss [27], [28] to optimize the instances. By introducing relative distance order between the positive sample pair and the negative sample pair, triplet loss and its variants prevail [29], [30], [31], [32], [33], [34], [35], [36] for person ReID. Hermans *et al*. introduce a batch-level hard triplet mining, which selects the hardest positive and the hardest negative samples within a batch for optimization [32]. Chen *et al*. propose the quadruplet loss which is built based on triplet loss and additionally pushes away negative pairs from positive pairs w.r.t. different probe images [37]. *These losses do not jointly consider the relative order of multiple instances (e.g.>3) in per-query optimization.*

Most recent works combine conventional classification loss and triplet-based loss in optimization for a higher performance [9], [10], [11], [38], [12], [13]. In this work, we introduce N-tuple loss and further develop its improved version for person ReID, which jointly optimizes multiple instances of multiple identities for a given query. This aligns better with the person ReID test/inference and thus leads to a superior performance.

Metric Learning. The study on metric learning [39], [40] stemmed from the era before deep learning. It has been an indispensable part of deep learning in the form of loss design for many applications, such as person ReID [25], [32], [37], face recognition [41], few-shot learning [42], [43]. As one of the most commonly used pairwise losses, contrastive loss is investigated in [44]. CenterLoss [45] calculates class-specific center in the embedding space and explicitly optimizes the intra-class compactness by explicitly minimizing the Euclidean distances between samples and their corresponding class centers. However, it leaves the inter-class distances under-considered. Triplet-based losses setup an anchor and pull the distance of the positive pair to be smaller than the negative pair. To guarantee the effectiveness of selected triplets, batch hard mining [32] and soft margin [32], [12] are widely used. Triplet loss is also improved in its speed and robustness by generalizing from optimizing instance-to-instance distances to optimizing instance-to-centroid distances towards its upper bound [46]. N-tuple loss pushes $N-2$ negative samples and pulls the positive pair all at once [47]. Being conceptually interesting, however, such joint ordering of multiple classes is under-explored in ReID. In this paper, we raise this under-explored issue in current loss designs for ReID and highlight that multi-class classification has an important impact on its performance, hopefully enabling other researchers in the field to leverage the full potential of multi-class joint optimization.

III. MPN-tuple Loss for Person ReID

To better understand the prevalent loss designs in person ReID, we first revisit the roles different loss designs (*i.e.*, classification loss and triplet loss) play in optimizing person ReID models. To remedy the limitation of adopting triplet loss and enable more effective feature learning, we introduce N-tuple loss to this field which aligns better with the test process of person ReID. This facilitates the joint comparisons with multiple instances per query sample rather than only with two

samples (in triplet loss). Moreover, we propose a new variant of N-tuple loss, named meta prototypical N-tuple (MPN-tuple) loss, to further promote the metric learning.

A. Revisiting Loss Designs Under a Unified View

We revisit different loss designs under a unified formulation. We show that the triplet loss can be treated as optimizing a two-class classification where the query/anchor sample is compared with two reference samples (a positive sample and a negative sample), and the probability of being classified as the positive class is maximized in the optimization.

A Unified Loss Formulation. To facilitate the understanding, we formulate the loss designs (for classifying the anchor as the j^{th} class) from a unified classification view as:

$$\begin{aligned}\mathcal{L}^i_{unified} &= -\log\frac{\exp(\frac{1}{\tau}\mathcal{S}(\boldsymbol{x}_a,\boldsymbol{c}_i))}{\sum_{k=1}^{C}\exp(\frac{1}{\tau}\mathcal{S}(\boldsymbol{x}_a,\boldsymbol{c}_k))} \\ &= \log\left(1+\frac{\sum_{k\neq j}^{C}\exp(\frac{1}{\tau}\mathcal{S}(\boldsymbol{x}_a,\boldsymbol{c}_k))}{\exp(\frac{1}{\tau}\mathcal{S}(\boldsymbol{x}_a,\boldsymbol{c}_j))}\right),\end{aligned} \tag{1}$$

where $\mathcal{S}(\cdot,\cdot)$ denotes the similarity between two feature vectors/nodes, $\boldsymbol{x}_a \in \mathbb{R}^d$ denotes the feature vector (of d dimensions) of an anchor (query) sample a to be classified/matched, and $\boldsymbol{c}_i \in \mathbb{R}^d$ denotes the weight vector corresponding to the class of $\boldsymbol{x}_a$ in the classifier. τ denotes a temperature parameter. We can write the weight matrix as $\mathbf{W}_b = [\boldsymbol{c}_1, \boldsymbol{c}_2, \cdots, \boldsymbol{c}_C] \in \mathbb{R}^{d\times C}$. Generally, $\boldsymbol{c}_k$ with $k = 1, 2, \cdots, C$ can be considered as the *reference nodes* for classification/matching where each node acts as the feature representation of the class center of a category. C denotes the number of the reference classes or instances. The more similar between $\boldsymbol{x}_a$ and a reference $\boldsymbol{c}_j$, the higher of the probability that they belong to the same class. *Minimizing this loss is to maximize the probability of correct classification.*

The Softmax function in Eq.(1) plays the role of normalization and enabling the interaction between the query sample and the reference nodes. For the query sample $\boldsymbol{x}_a$, it optimizes the similarities/distances between the query and the reference nodes, enabling the joint comparisons among these pairs.

This unified formulation can be instantiated to different loss designs. Here, we discuss the widely used conventional classification loss and triplet loss in person ReID in detail.

Conventional Classification Loss. In the unified formulation, when C is the total number of classes/identities in the training set, and the weight matrix $\mathbf{W}_b = [\boldsymbol{c}_1, \boldsymbol{c}_2, \cdots, \boldsymbol{c}_C] \in \mathbb{R}^{d\times C}$ is composed of the learned weights of the Fully-Connected (FC) layer of a classifier and $\mathcal{S}(\cdot,\cdot)$ is calculated by the inner product, the loss for the sample $\boldsymbol{x}_a$ becomes the conventional classification loss as:

$$\mathcal{L}_{cls} = -\log\frac{\exp(\frac{1}{\tau}\boldsymbol{c}_i^{\mathrm{T}}\boldsymbol{x}_a)}{\sum_{k=1}^{C}\exp(\frac{1}{\tau}\boldsymbol{c}_k^{\mathrm{T}}\boldsymbol{x}_a))}, \tag{2}$$

which denotes the negative logarithm of the probability of classifying the sample $\boldsymbol{x}_a$ into the i^{th} class. C is the total number of classes/identities in the training set. Each reference node $\boldsymbol{c}_k$ is a learned weight vector (*i.e.*, network parameters), that plays the role of "class center". The classification probability for the sample $\boldsymbol{x}_a$ is obtained by comparing the similarities/distances between $\boldsymbol{x}_a$ and all the "class centers".

The conventional classification loss as formulated in Eq.(2) is widely used in person ReID. It optimizes the similarities between the query instance (to be classified) and all "class centers", which plays a role of globally optimizing the "class centers". The classification loss thus inclines to improve class discrimination but lacks the adaptability in the instance level. As illustrated by the toy examples in Fig. 1, even when all samples are already correctly classified based on the similarity to the class centers, it is still questionable to determine the relative order of different instances for a given query based on their distances. For example, for the query sample s_1, its distance to the positive sample s_2 is even larger than that to the negative sample s_3. *Conventional classification loss lacks the capability of optimizing the relative order of instance pairs.*

Triplet Loss. Triplet loss explicitly optimizes the distances/similarities of a positive instance pair and a negative instance pair for a given query. It remedies the above discussed limitation of the conventional classification loss. The vanilla version of triplet loss encourages the similarity between the anchor/query and a positive sample to be larger than the similarity between this anchor and a negative sample by a hard margin m as below:

$$\mathcal{L}_{triplet} = [m + \mathcal{S}(\boldsymbol{x}_a, \boldsymbol{x}^-) - \mathcal{S}(\boldsymbol{x}_a, \boldsymbol{x}^+)]_+, \tag{3}$$

where $\boldsymbol{x}_a$, $\boldsymbol{x}^+$, and $\boldsymbol{x}^-$ denote an anchor sample, a positive sample (that has the same identity as $\boldsymbol{x}_a$), and a negative sample (that has a different identity to $\boldsymbol{x}_a$), respectively. $[\cdot]_+ = max(\cdot, 0)$. The soft margin variant of triplet loss has been demonstrated to be more effective for person ReID [32], [12] than its other variants. It replaces the hinge function $[m + \cdot]_+$ by softplus function $\log(1 + \exp(\cdot))$ which decays exponentially instead of having a hard cut off. It is defined as:

$$\mathcal{L}_{triplet} = \log(1 + \exp(\mathcal{S}(\boldsymbol{x}_a, \boldsymbol{x}^-) - \mathcal{S}(\boldsymbol{x}_a, \boldsymbol{x}^+)), \tag{4}$$

which is equivalent to:

$$\begin{aligned}\mathcal{L}_{triplet} &= \log(1 + \exp(\mathcal{S}(\boldsymbol{x}_a, \boldsymbol{x}^-)/\exp(\mathcal{S}(\boldsymbol{x}_a, \boldsymbol{x}^+)) \\ &= -\log\frac{\exp(\mathcal{S}(\boldsymbol{x}_a, \boldsymbol{x}^+))}{\exp(\mathcal{S}(\boldsymbol{x}_a, \boldsymbol{x}^+)) + \exp(\mathcal{S}(\boldsymbol{x}_a, \boldsymbol{x}^-))}.\end{aligned} \tag{5}$$

Comparing Eq.(1) and Eq.(5), triplet loss with soft margin is actually an instantiation of the unified classification loss (see Eq.(1)), by setting C=2 and taking the positive sample $\boldsymbol{x}^+$ and the negative sample $\boldsymbol{x}^-$ as reference nodes. The loss maximizes the probability of the sample $\boldsymbol{x}_a$ to be classified into the class corresponding to the positive sample $\boldsymbol{x}^+$, by encouraging a higher similarity with the positive sample and a lower similarity with the negative sample. We thus view the soft-margin triplet loss as a classification loss over two categories/identities (*i.e.*, a two-class classification loss). It optimizes instance-to-instance distances with two instances as the reference nodes. A temperature factor $1/\tau$ is multiplied over the similarity function.

For triplet loss, only three instances of two classes are jointly optimized at once in a per-query optimization. In a

min-batch, multiple triplets are usually sampled to calculate the batch-level triplet losses. One may argue that when we look at multiple triplets (*e.g.*, two), multiple instances are also "jointly" optimized. Actually, this viewpoint is questionable. For example, in the case of two triplets, there is a lack of valid interaction between them. *The optimization directions of the two triplets may contradict each other as illustrated in Fig. 1 (analyzed in Section I), leading to inferior optimization.*

B. N-tuple Loss

We propose to jointly optimize multiple instances in a per query optimization to address the limitation of triplet loss. Particularly, we increase the number of instances jointly considered in per-query optimization, enabling the comparisons of a query sample with more samples. This better aligns with ReID inference, where a query sample is to be compared with the samples in the gallery set. Although simple and intuitive in concept, this is overlooked in the person ReID literature.

We introduce N-tuple loss to enable the joint optimization of multiple instances (from more than two identities). N-tuple loss [47] allows the interaction of an anchor/query sample with multiple samples from multiple (more than two) different classes. Given an anchor sample $\boldsymbol{x}_a$, N-tuple loss is defined as:

$$\begin{aligned}\mathcal{L}_{Ntuple} = -\log\frac{\exp(\frac{1}{\tau}\mathcal{S}_a^+)}{\exp(\frac{1}{\tau}\mathcal{S}_a^+ + \sum_{k=1}^{N-2}\exp(\frac{1}{\tau}\mathcal{S}_{a,k}^-)},\\ \mathcal{S}_a^+ = \mathcal{S}(\boldsymbol{x}_a, \boldsymbol{x}^+), \qquad \mathcal{S}_{a,k}^- = \mathcal{S}(\boldsymbol{x}_a, \boldsymbol{x}_k^-),\end{aligned} \tag{6}$$

where the N of a N-tuple refers to the number of elements in this tuple including one anchor sample $\boldsymbol{x}_a$, one positive sample and N-2 negative nodes. $\boldsymbol{x}_k^-$, $k = 1, \cdots, N-2$ corresponds to $N-2$ negative samples of $N-2$ different classes, and $\boldsymbol{x}^+$ corresponds to a positive sample (same identity as the anchor/query sample). Compared to the soft-margin triplet loss in Eq.(5), the difference in Eq.(6) is that multiple negative samples instead of one are used. When $N = 3$, it degenerates to the soft-margin triplet loss. N-tuple loss actually corresponds to a multi-class classification and optimizes instance-to-instance distances with $N-1$ instances as reference nodes and one instance as query. *The joint optimization of multi-instances promotes the relative order of distances among many instance pairs which matches ReID inference well.*

C. Proposed Meta Prototypical N-tuple Loss

Based on the multi-class classification loss, *i.e.*, N-tuple loss, we further propose a *Meta Prototypical N-tuple* loss (abbreviated as MPN-tuple loss) for effective optimization in person ReID. In N-tuple loss, feature vectors of instances themselves are taken as the reference nodes (comprising a classifier) to perform multi-class classification for multiple instances. Here, we propose to learn better reference nodes from instance features via a trainable meta-learner. First, we incorporate a trainable meta-learner as a predictor to predict the category-specific reference node from each instance in a mini-batch. Second, when the number of classes jointly taken into account increases in per-query optimization, the total number of tuples increases exponentially and becomes intractable quickly. We average the reference nodes of the same identity within a mini-batch to be the final reference node in the N-tuple optimization. The advantages lie in two aspects. 1) The feature representation of one instance might be affected by the noise of this instance. In contrast, the averaged result across multiple instances is more robust to be a reference node. 2) This reduces the number of N-tuples to make it trackable especially when N is large.

Particularly, instead of directly using the features of sampled instances as the reference nodes (as in N-tuple loss), we propose to employ a meta learner by using a mapping subnet $\phi(\cdot)$ for obtaining the reference nodes based on the refined instance features. We define $\phi(\boldsymbol{x}_k)$ as

$$\phi(\boldsymbol{x}_k) = W_2(\mathrm{BN}(W_1(\boldsymbol{x}_k))), \tag{7}$$

where $\phi(\cdot)$ is implemented by two Fully-Connected (FC) layers with a Batch Normalization (BN) layer, $W_1 \in \mathbb{R}^{\frac{d}{s}\times d}$, $W_2 \in \mathbb{R}^{d\times\frac{d}{s}}$, wherein s is an integer which controls the dimension reduction ratio and we experimentally set it to 8. The dimension reduction here is inspired by the squeeze and excitation operations in [48], which reduces the number of parameters to make the optimization easier. Here, we define the similarity function $\mathcal{S}$ as the cosine similarity of the two input vectors as

$$\mathcal{S}(\boldsymbol{x}_a, \phi(\boldsymbol{x}_k)) = (\phi(\boldsymbol{x}_k)^{\mathrm{T}}\boldsymbol{x}_a)/(\|\phi(\boldsymbol{x}_k)^{\mathrm{T}}\| \cdot \|\boldsymbol{x}_a\|). \tag{8}$$

For an anchor $\boldsymbol{x}_a$, we define the Meta N-tuple loss as:

$$\begin{aligned}\mathcal{L}_{MNtuple} = -\log\frac{\exp(\frac{1}{\tau}\mathcal{S}_a^{\phi(+)})}{\exp(\frac{1}{\tau}\mathcal{S}_a^{\phi(+)}) + \sum_{k=1}^{N-2}\exp(\frac{1}{\tau}\mathcal{S}_{a,k}^{\phi(-)})},\\ \mathcal{S}_a^{\phi(+)} = \mathcal{S}(\boldsymbol{x}_a, \phi(\boldsymbol{x}^+)), \quad \mathcal{S}_{a,k}^{\phi(-)} = \mathcal{S}(\boldsymbol{x}_a, \phi(\boldsymbol{x}_k^-)).\end{aligned} \tag{9}$$

Essentially, the introduction of $\phi(\cdot)$ enables more effective metric learning, where $\phi(\cdot)$ learns to map extracted features to the refined representations in a trainable way. The optimized feature representations as the reference nodes can then further guide the optimization of features (before the mapping).

For K samples ($\boldsymbol{x}_{c,j}$, with $j = 1, \cdots, K$) of the class c, we build its prototype reference node by averaging their mapped features as $\widehat{\phi_c} = \frac{1}{K}\sum_{j=1}^{K}\phi(\boldsymbol{x}_{c,j})$. For an anchor sample $\boldsymbol{x}_a$, we define the Meta Prototypical N-tuple loss as:

$$\begin{aligned}\mathcal{L}_{MPNtuple} = -\log\frac{\exp(\frac{1}{\tau}\mathcal{S}_a^{\widehat{\phi^+}})}{\exp(\frac{1}{\tau}\mathcal{S}_a^{\widehat{\phi^+}}) + \sum_{k=1}^{N-2}\exp(\frac{1}{\tau}\mathcal{S}_{a,k}^{\widehat{\phi^-}})},\\ \mathcal{S}_a^{\widehat{\phi^+}} = \mathcal{S}(\boldsymbol{x}_a, \widehat{\phi^+}), \qquad \mathcal{S}_{a,k}^{\widehat{\phi^-}} = \mathcal{S}(\boldsymbol{x}_a, \widehat{\phi_k^-}).\end{aligned} \tag{10}$$

where $\widehat{\phi^+}$ denotes the prototype reference node obtained from the positive samples (corresponding to the same class as the anchor sample) while $\widehat{\phi_k^-}$ denotes that for the k^{th} negative class. We will demonstrate the effectiveness of the proposed MPN-tuple loss for person ReID in the experiment section.

As described above, the prototype reference nodes in our proposed method are obtained through averaging the refined feature representations inferred by the meta-learner. Here, the meta-learner can be formulated as a trainable function $\phi(\cdot)$ as mentioned before, which aims to predict the class centers

adaptively. Thus, the class-specific prototypes in Eq.(10) are updated dynamically as the feature learning so that they are robust to data noises and also compact as shown in the histogram plotting of the following experiment part. We note that the class centers are also adopted in CenterLoss [45] to regularize each sample to approach the corresponding updated center of the same class for minimizing the intra-class distances. However, it leaves the distances of negative pairs under-considered. In contrast, our proposed MPN-tuple loss enables a joint multi-class instance optimization to optimize the similarities of both positive pairs and negative pairs simultaneously for better metric learning. The testing scenario of person ReID is to match a query sample with the samples of the same identity in the gallery set and avoid it to be matched to different identities. Thus, the MPN-tuple loss aligns better with the optimization objective of this task.

IV. Experiments

A. Datasets and Evaluation Metrics

Datasets. We evaluate our methods using three widely-used person ReID datasets: *i.e.*, CUHK03 [27], Market1501 [49], DukeMTMC-reID [50] and the large-scale MSMT17 [51].

CUHK03 [27] consists of 1,467 pedestrians. This dataset provides both manually labeled bounding boxes from 14,096 images and DPM-detected bounding boxes from 14,097 images. We adopt the new training/testing protocol following [52], [53], [54]. In this protocol, 767 identities are used for training and the remaining for testing. We only show the evaluation results for the labeled setting (L) while the detected setting (D) presents a similar trend.

Market-1501 [49] contains 12,936 images of 751 identities for training and 19,281 images of 750 identities for testing, which are captured by 6 cameras.

DukeMTMC-reID [50] has 36,411 images, where 702 identities are used for training and 702 identities for testing. They are captured by 8 cameras.

MSMT17 [51] contains 126,441 images, where 1,041 identities and 3,060 identities are used for training and testing respectively. They are captured by 15 cameras.

Evaluation Metrics. We follow the common practices to use Rank-1 (R1) and mean average precision (mAP) for evaluating person ReID models.

B. Implementation Details

We perform the empirical study of combining triplet loss and conventional classification loss and evaluate our proposed MPN-tuple loss on the regular supervised person ReID. Besides, we further verify the robustness of MPN-tuple loss on visible-infrared and cloth-changing person ReID. Here, we introduce the configurations of the main body of our experiments in this section, leaving the detailed introduction of experiment configurations for visible-infrared and cloth-changing person ReID in the corresponding sub-sections.

Network Settings. We follow the common practices in ReID [35], [10], [13] and take ResNet-50 [55] to build our baseline network for effectiveness validation. Similar to [56], [10], we remove the last spatial down-sampling operation in the conv5_x block of ResNet-50. Unless otherwise stated, similar to [57], we add Instance Normalization to the first three blocks (conv2_x-conv4_x) to enhance model's generalization ability [58], which is found effective in improving the performance because the identities during testing are unseen (different from the training identities). In detail, following IBN-a in [57], we replace half of the channels of Batch Normalization (BN) by Instance Normalization (IN) for the first BN layer of the residual blocks within $conv2\_x$, $conv3\_x$ and $conv4\_x$, and leave other BN layers in other positions intact.

On top of the spatially pooled feature (2048 dimensions) of ResNet-50, a Batch Normalization (BN) layer is added to obtain the ReID feature vector $\boldsymbol{x} \in \mathbb{R}^{1024}$ and a followed Fully Connected (FC) layer is employed as the classifier for adding the conventional classification loss. In our studies, the multi-class classification losses are added on the ReID feature vector $\boldsymbol{x}$ by default. For this loss, similar to [59], the temperature parameter τ is a learnable parameter. In our implementation, for the purpose of numerical stability, we learn a weight parameter s where $s = 1/\tau$ instead of τ. Note that we do not employ re-ranking [52] in all our experiments.

Training. We use the commonly used data augmentation strategies of random cropping [60], horizontal flipping, and random erasing [61]. The input resolution is set to 384×128. Each batch includes $B = P\times K$ images. P and K denote the number of different persons (identities) and the number of different images per person, respectively. We perform the experiments with $P = 16$, $K = 4$ using one GPU card. In a batch, the total number of triplets is $T = C_P^N \cdot K^N N(K-1) = 11520$. Even for batch hard mining triplet, the similarities for all the sample pairs in a batch need to be calculated for the selection of the hard triples so that it actually has similar training complexity as using all the triplets. But, as the number of classes increases in N-tuple loss, the total number of tuples increases exponentially which quickly becomes intractable. Therefore sampling the tuples is desirable to limit the complexity.

We initialize the ResNet-50 [55] backbone network with ImageNet [62] pre-trained weights and train the ReID network for 600 epochs in total. An epoch means that all identities in the entire training dataset are traversed. We adopt Adam optimizer with the momentum of 0.9 and the weight decay of 5×10^{-4}. During the training, we first warm up with a linear growth learning rate from 8×10^{-6} to 8×10^{-4} for 20 epochs and the learning rate is decayed by a factor of 0.5 for every 60 epochs. Unless otherwise specified, we train the entire network in an end-to-end manner.

Testing. In the testing, for a given person image, we take the feature vector that is obtained by adopting global average pooling in spatial on the feature map extracted by the ResNet-50 backbone as the final representation of this sample. We calculate the cosine similarities over different person images and perform ranking/retrieval based on such calculated similarities.

C. Empirical Study of Triplet Loss and Conventional Classification Loss for Person ReID

The joint use of the triplet loss and the conventional classification loss achieves superior performance and is predominant

TABLE I
A CASE STUDY FOR COMBINING TRIPLET LOSS (TRI.) AND CONVENTIONAL CLASSIFICATION LOSS (CLS.). *Distance* DENOTES THE SIMILARITY METRIC USED FOR TRIPLET LOSS. *HardMining* DENOTES WHETHER BATCH HARD MINING IS USED IN TRIPLET LOSS.

Loss	Distance	HardMining	CUHK03(L)		Market1501		DukeMTMC		MSMT17	
			R1	mAP	R1	mAP	R1	mAP	R1	mAP
Cls.	-	-	67.2	63.6	94.1	83.5	85.6	73.8	72.7	46.8
Tri.	Euclidean	yes	79.4	75.0	94.0	84.8	86.9	74.6	73.8	49.8
Tri.	Euclidean	no	63.7	60.2	87.6	74.0	75.9	59.0	55.0	33.0
Tri.	Cosine	yes	63.3	57.5	83.1	65.3	81.6	66.4	33.2	25.6
Tri.	Cosine	no	43.1	40.6	68.8	50.2	65.0	46.2	25.2	12.5
Tri.+Cls.	Euclidean	yes	79.6	75.8	94.9	86.6	87.3	76.7	78.8	56.0
Tri.+Cls.	Euclidean	no	74.3	70.6	**95.0**	86.9	87.3	77.1	77.6	53.9
Tri.+Cls.	Cosine	yes	80.7	76.4	94.6	86.9	88.8	77.9	78.6	54.5
Tri.+Cls.	Cosine	no	**81.8**	**78.2**	94.7	**87.3**	**88.7**	**78.3**	**79.8**	**56.2**

in ReID. For triplet loss, there is a lack of comprehensive study on its design choices and corresponding effectiveness. In this section, we study the good practice of the triplet loss designs when combining it with the classification loss, including the similarity functions (*i.e.*, Euclidean, Cosine similarity), sampling mechanisms (batch hard mining or not). Note that for triplet loss, we use soft-margin triplet (see (4)) which has been demonstrated to be better than triplet with hard margin [32] (we have the similar observations).

Table I shows the results. We have the following observations. **1)** The joint use of triplet loss (Tri.) and classification loss (Cls.) achieves much better performance than using only one, even brings 3.2% improvement in mAP on CUHK03(L). Without classification loss, triplet loss alone using cosine similarity suffers from difficulty in optimization and is easy to be trapped to local optimal. Thus, they are complementary in learning discriminative features. **2)** When Tri. and Cls. are jointly used, employing cosine similarity in general significantly outperforms employing (the negative of) Euclidean distance. Note that in the ReID inference, normalization on each sample is generally performed for the matching to exclude the interference of the amplitudes (energies) of sample features. Cosine similarity inherently evaluates the correlation of two features with energy normalized and it aligns better with ReID inference. **3)** When Tri. and Cls. are jointly used with cosine similarity, batch hard mining (which selects the hard positive and hard negative samples to form a triple) is inferior to the scheme without hard mining. That may be because the soft-margin enables the optimization of moderate hard triples and easy triples while hard mining could ignore those triplets. This phenomena is not observed for the Euclidean distance setting.

Hereafter, we refer to the scheme (last row in Table I) with the best combination of design choices as *Baseline*.

D. Effectiveness of Multi-class Classification Loss and Our MPN-tuple Loss

We validate the effectiveness of the vanilla N-tuple loss and our proposed Meta Prototypical N-tuple (MPN-tuple) loss for person ReID. The conventional classification loss is always used hereafter in considering its complementary role. Table II shows the results. *Baseline* refers to our obtained strong baseline (*i.e.*, the best one in Table. I). *P2S Tri.* refers to point-to-set triplet loss wherein the prototypes (the average results over all instances of the same class) are taken as the reference nodes. This can reduce the number of triplets for calculating multi-class classification losses within a batch to $B = 64$. We observe that *P2S Tri. + Cls.* is slightly inferior to *Baseline* but the number of formed triplets within a batch is smaller.

Influence of the Number of Classes in N-tuple Loss. We investigate the influence of the number of jointly considered classes (denoted by # Cls) for a given query by increasing the N in N-tuple loss. With the increase of N, the number of possible tuples $C_P^N \cdot K^N N(K-1)$ increases exponentially such that it quickly becomes intractable. We randomly sample M tuples to calculate the losses. For fair comparison among schemes with different number of classes, we set $M = T$, where T is the number of total triplets in a batch. We denote these schemes as *N-tuple+Cls.*.

In Table II, we observe that as the number of classes in *N-tuple* loss increases, the person ReID performance in general increases. When the number of classes increases from 2 to 16, the mAP accuracy is improved by 1.4%, 0.7%, 1.1%, and 2.4% on CUHK03, Market1501, DukeMTMC, and MSMT17, respectively. Note that since the *Baseline* is already very strong with superior performance, our gains on top of it can be considered as significant. The gradients of two different triplets may contradict each other as discussed in Section I. Although P identities are involved in calculating the triplet loss at the batch level, it still does not allow valid interaction among different triplets. In N-tuple loss, as N increases, more instances are jointly considered so that the corresponding instances are optimized towards the correct ranking with respect to the given query. Note that the performance of *N-tuple + Cls.* with $N = 2$ is lower than our *Baseline (Tri.+Cls.)* because the random sampling in N-tuple loss cannot assure a complete traversal over all triplets even though the number of samplings is the same as the number of all triplets.

Effectiveness of Our Proposed MPN-tuple Loss. To reduce the number of possible tuples in N-tuple loss, we calculate the prototype of multiple instances of the same identity and take the calculated prototype as reference node instead of using sample instance. In this way, the number of possible tuples is reduced from $C_P^N \cdot K^N N(K-1)$ to $B \cdot C_P^N$. We denote such scheme as *PN-tuple + Cls.*. Taking the extreme $N = 16$ case as an example, the number of all possible tuples is reduced to 64 and we use the 64 tuples to calculate the losses. *PN-tuple + Cls.* achieves similar performance as *N-tuple + Cls.* when $N = 16$ but avoids the experience of too many tuples

TABLE II
PERFORMANCE (%) COMPARISONS FOR TRIPLET LOSS, MULTI-CLASS CLASSIFICATION (N-TUPLE LOSS), AND OUR PROPOSED MPN-TUPLE LOSS. *Baseline* REFERS TO OUR STRONG BASELINE (THE BEST ONE IN TABLE I). *P2S Tri.+Cls.* REFERS TO THE USE OF POINT-TO-SET TRIPLET LOSS AND CLASSIFICATION LOSS. *#Cls* DENOTES THE NUMBER OF CLASSES.

Loss	#Cls	CUHK03(L)		Market1501		DukeMTMC		MSMT17	
		R1	mAP	R1	mAP	R1	mAP	R1	mAP
Baseline(Tri.+Cls.)	2	81.8	78.2	94.7	87.3	88.9	78.3	79.8	56.2
P2S Tri.+Cls.	2	81.1	77.8	94.8	86.7	88.5	78.1	80.0	55.6
N-tuple+Cls.	2	81.4	77.7	94.4	87.0	88.9	78.1	79.2	55.7
N-tuple+Cls.	4	82.1	78.4	94.5	87.2	88.8	78.6	80.0	57.2
N-tuple+Cls.	8	82.1	78.9	94.5	87.5	89.0	79.0	80.3	57.8
N-tuple+Cls.	16	82.2	79.1	94.7	87.7	89.4	79.2	80.2	58.1
PN-tuple+Cls.	16	82.9	79.6	94.8	87.5	89.7	78.8	80.9	58.2
MPN-tuple+Cls.	16	**84.4**	**80.3**	**95.3**	**88.7**	**89.5**	**79.7**	**82.2**	**60.1**

TABLE III
PERFORMANCE (%) COMPARISONS WITH THE STATE OF THE ART METHODS. BOLD NUMBERS DENOTE THE BEST PERFORMANCE AND THE NUMBERS WITH UNDERLINES DENOTE THE SECOND BEST ONES.

Model	CUHK03(L)		Market1501		DukeMTMC		MSMT17	
	R1	mAP	R1	mAP	R1	mAP	R1	mAP
HAP2S [34]	-	-	84.6	69.4	76.0	60.6	-	-
CE-FAT [36]	-	-	91.4	76.4	80.8	63.1	69.4	39.2
IDE [2]	43.8	38.9	85.3	68.5	73.2	52.8		
IDO-Cls [8]	62.8	56.7	93.9	80.5				
Gp-reid [35]	-	-	92.2	81.2	85.2	72.8	-	-
Bag of Tricks [13]	-	-	94.5	85.9	86.4	76.4	-	-
IANet [63]	-	-	94.4	83.1	87.1	73.4	75.5	46.8
HCTL [64]	-	-	93.8	81.8	83.3	68.2	74.3	43.6
PCB+RPP [56]	63.7	57.5	93.8	81.6	83.3	69.2	68.2	40.4
MGN [9]	68.0	67.4	95.7	86.9	88.7	<u>78.4</u>	-	
DSA-reID [10]	78.9	75.2	95.7	87.6	86.2	74.3	-	
SAN [23]	80.1	76.4	**96.1**	<u>88.0</u>	87.9	75.5	79.2	55.7
MHN-6(PCB) (Chen et al. 2019)	77.2	72.4	95.1	85.0	<u>89.1</u>	77.2	-	-
BAT-net [38]	78.6	76.1	95.1	84.7	87.7	77.3	79.5	56.8
OSNet [65]	-	-	94.8	84.9	88.6	73.5	78.7	52.9
Mancs [19]	69.0	63.9	93.1	82.3	84.9	71.8	-	-
JDGL [66]	-	-	94.8	86.0	86.6	74.8	77.2	52.3
RGA-SC [11]	80.4	76.5	<u>95.8</u>	88.1	86.1	74.9	<u>81.3</u>	<u>56.3</u>
GASM [67]	-	-	95.3	84.7	88.3	74.4	79.5	52.5
FIDI [68]	75.0	73.2	94.5	86.8	88.1	77.5	-	-
Baseline(Cls.+Tri.)	<u>81.8</u>	<u>78.2</u>	94.7	87.3	88.7	78.3	79.8	56.2
Ours(Cls.+MPN-tuple)	**84.4**	**80.3**	95.3	**88.7**	**89.5**	**79.7**	**82.2**	**60.1**

in calculating the loss value.

MPN-tuple+Cls. denotes our scheme where a meta learner (a mapping subnet) is introduced for inferring reference nodes. This enables a more general similarity metric for effective feature learning. We can see that *MPN-tuple+Cls.* with $N = 16$ achieves significant improvement over *PN-tuple+Cls.*, *i.e.*, 0.7%, 1.2%, 0.9%, and 1.9% gain in mAP accuracy on four benchmark datasets, respectively. Thanks to the introduction of prototypes and the meta-learner, *MPN-tuple+Cls.* achieves the best performance. Note that at the initial stage when the network has not been trained well, it is challenging for the meta learner to learn a good representation. Thus, we use three-stage training. In the first stage, we train the network with classification loss and PN-tuple loss for 360 epoches. In the second stage (361-480 epoches), we fix the network and only train the meta-learner with MPN-tuple loss and the FC layer corresponding to the classification loss. In the third stage (480-600 epoches), we jointly finetune the entire network.

Note that it's desirable to involve as many classes as possible but impractical in the case when the number of classes is limited within a batch. We fix the batch size for fair comparison and convincing ablation study, because the learning for person ReID is sensitive to the batch size used. Thus, the number of classes is also limited. We set it to 16.

E. Comparison with the State-of-the-Arts

Table III shows the comparison results with the state-of-the-art approaches. We group these approaches into two groups. The first group aims at designing strong baseline networks, including loss deigns and training tricks. In [13], bag of tricks are collected and evaluated for person ReID and a strong baseline built based on ResNet-50 is provided. The other group of approaches focuses on special network designs for person ReID. Some approaches [56], [9], [20] ensemble the local region feature representations, while others introduce attention designs to focus on discriminative features [19], [38], [11]. Besides, there are some approaches exploiting auxiliary semantics (*e.g.*, dense semantics [10], [23]) to address the misalignment (caused by the diverse viewpoints and poses).

Our study belongs to the first group. Thanks to the re-investigation on triplet loss design choices, we provide a strong baseline sdheme *Baseline*, which jointly uses the soft-margin triplet loss (with cosine similarity, without hard mining) and

TABLE IV
PERFORMANCE (%) COMPARISONS FOR TRIPLET LOSS, MULTI-CLASS CLASSIFICATION (N-TUPLET LOSS), AND OUR PROPOSED MPN-TUPLE LOSS ON TOP OF THE *Plain Baseline*. THE SYMBOL "✓" REPRESENTS "USING IBN" WHILE "✗" REPRESENTS "ALL NOT USING IBN".

Model & Loss	# Class	IBN	Resolution	CUHK03(L)		Market1501		DukeMTMC		MSMT17	
				Rank-1	mAP	Rank-1	mAP	Rank-1	mAP	Rank-1	mAP
(1) Baseline (Tri. + Cls.)	2	✗	256 × 128	69.7	66.5	93.6	83.4	86.0	75.5	71.9	46.2
(2) N-tuplet + Cls.	2	✗	256 × 128	69.3	66.1	94.2	84.2	85.9	75.5	71.5	46.3
(3) N-tuplet + Cls.	4	✗	256 × 128	71.9	68.6	94.5	85.6	87.2	76.8	72.9	48.2
(4) N-tuplet + Cls.	8	✗	256 × 128	74.9	70.3	94.7	86.1	87.3	76.9	73.7	49.3
(5) N-tuplet + Cls.	16	✗	256 × 128	75.4	71.0	94.7	86.2	86.8	77.3	74.3	49.9
(6) MPN-tuple + Cls.	16	✗	256 × 128	77.7	73.4	94.6	86.4	87.3	77.5	77.6	52.4

TABLE V
AN ABLATION STUDY ON THE IMPACT OF INSTANCE NORMALIZATION IN BACKBONE NETWORK, AND THE INPUT IMAGE RESOLUTION. *Baseline* REFERS TO OUR BASELINE MODELS WHICH USE TRIPLET LOSS (DENOTED BY "TRI.") AND CONVENTIONAL CLASSIFICATION LOSS (DENOTED BY "CLS.") AS SUPERVISION (SEE TABLE I). "✓" REPRESENTS "USING IBN" WHILE "✗" REPRESENTS "ALL NOT USING IBN".

Model & Loss	# Class	IBN	Resolution	CUHK03(L)		Market1501		DukeMTMC		MSMT17	
				Rank-1	mAP	Rank-1	mAP	Rank-1	mAP	Rank-1	mAP
(1) Baseline (Tri. + Cls.)	2	✗	256 × 128	69.7	66.5	93.6	83.4	86.0	75.5	71.9	46.2
(2) Baseline (Tri. + Cls.)	2	✗	384 × 128	76.4	73.1	94.4	85.2	87.8	77.5	76.1	52.2
(3) Baseline (Tri. + Cls.)	2	✓	384 × 128	81.8	78.2	94.7	87.3	88.9	78.3	79.8	56.2
(4) Ours (MPN-tuple + Cls.)	16	✓	384 × 128	**84.4**	**80.3**	**95.3**	**88.7**	**89.5**	**79.7**	**82.2**	**60.1**

classification loss (see Table I about the ablation study). We can see that our *Baseline* achieves high performance, being superior or competitive to the state-of-the-art approaches. We denote our final scheme with the proposed MPN-tuple loss as *MPN-tuple + Cls*. It achieves the best mAP accuracy on all these datasets and outperforms the *Baseline* by a large margin, achieving 2.1%, 1.4%, 1.4% and 3.9% gain in mAP accuracy on the four datasets, respectively. Our design is simple yet effective. There is no increase in the computational complexity during the inference. We hope our scheme could serve as a strong baseline for the ReID community and inspire more novel designs on loss functions.

F. Effectiveness of our Proposed Loss Designs on the Plain Baseline

We have validated the effectiveness of N-tuple loss and our proposed Meta Prototypical N-tuplet (MPN-tuple) loss on top of our strong baseline scheme *Baseline* in Section IV-D. Here, we further study their effectiveness on top of the *Plain Baseline* (*i.e*., the model (1) in Table IV) which does not use IN and uses the low resolution setting. Table IV shows the results.

For model (2) to (5) in Table IV, we can see that both the mAP and Rank-1 accuracy are consistently and significantly improved as the number of classes (denoted by "# Class") increases for the N-tuple loss. When the number of classes increases from 2 to 16, the mAP accuracy is improved by 4.9%, 2.0%, 1.8% and 3.6% on CUHK03, Market1501, DukeMTMC, and MSMT17 respectively, which is more significant than that on the strong baseline (see Table V uses IBN and a higher resolution). This demonstrates that the increase of the number of jointly considered classes in per-query optimization is very helpful for ReID, since such comparisons among more classes are more consistent with the ReID test/inference which is actually a retrieval/comparison process.

The model with the incorporation of our proposed MPN-tuple loss (*i.e*., model (6)) significantly outperforms the *Plain Baseline*, *i.e*., model (1), by 6.9%, 3.0%, 2.0% and 6.2% on CUHK03, Market1501, DukeMTMC, and MSMT17, respectively. Our MPN-tuple loss enables stronger metric learning and is superior to N-tuplet loss.

G. Influence of Input Resolution and IBN on Baseline

We have conducted an empirical study of the effects of loss designs for person ReID baseline models (see Section IV-D). For those settings in Section IV-D, we use ResNet-50 backbone with Instance Normalization (IN) inserted (*i.e*., IBN) [57] and the input image resolution is 384×128. Note that 384×128 and 256×128 are two commonly used input resolutions in person ReID literatures. Here, we study the influence of Instance Normalization (IN) and the input resolutions. Table V shows the results.

In Table V, we refer to model (1) which does not use IBN and uses the low resolution setting as *Plain Baseline* while referring model (3) that uses IBN and a higher resolution as *Strong Baseline*. Comparing *Plain Baseline* to *Strong Baseline*, we find both factors bring significant improvements, wherein IBN enhances the generalization performance for the "unseen" testing identities, and higher resolutions can preserve more details of the input images. When compared with the *Strong Baseline* (*i.e*.model (3) in Table V), our proposed MPN-tuple loss brings 2.1%, 1.4%, 1.4%, and 3.9% improvements in mAP on CUHK03, Market1501, DukeMTMC, and MSMT17, respectively. This demonstrates the effectiveness of our MPN-tuple loss. Note that the gain of MPN-tuple loss on *Strong Baseline* (see Table V) is smaller than that on *Plain Baseline* (see Table IV), which is because the stronger the baseline, the harder it is to obtain additional gain based on this baseline.

H. Analysis of Features Learned from Our MPN-tuple Loss

In our proposed MPN-tuple loss, we adopt a trainable meta-learner to predict the category-specific reference node as a

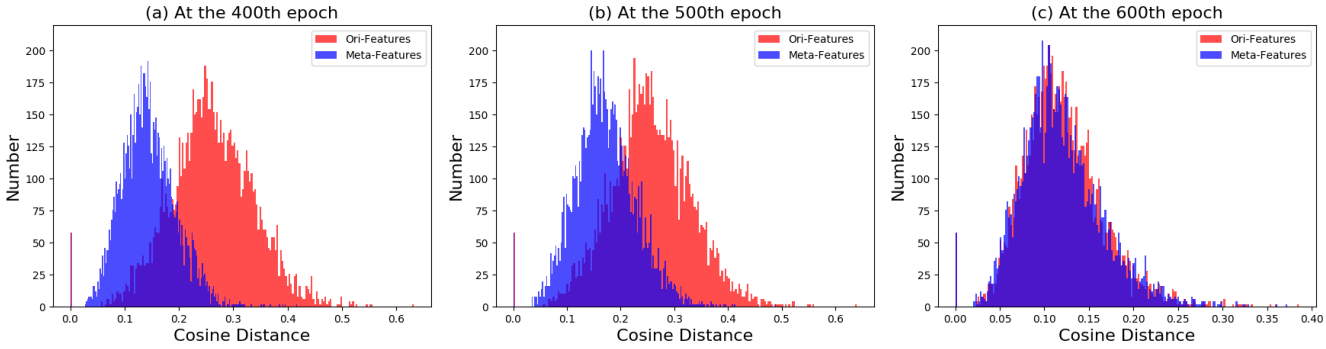


Fig. 2. Histograms of distances of positive sample pairs (*i.e.*, the pairs of the same identity). Red: within-class distances between *original features* of two samples (*i.e.*, the features before the mapping via the meta learner (a subnet)). Blue: within-class distances between *meta-features* of two samples (*i.e.*, the output features of the meta learner).

refined representation from each instance in a mini-batch. The meta-learner is end-to-end trained. Minimizing the loss on the "refined" feature vectors (after meta-learner) will further derive the optimization of the input features of the meta-learner through end-to-end training. This is prone to enforce closer distances between the anchor and positive samples in the feature space and farther distances between the anchor and negative samples to ease the optimization. To demonstrate this, we visualize the the histogram of distances of positive sample pairs as the training process goes in Fig. 2.

Based on our model which is trained with *MPN-tuple+Cls.* losses on the training set of Market1501, we visualize the distribution of the feature distances of positive (*i.e.*, the same identity) sample pairs at the 400^{th} epoch, the 500^{th} epoch, and the 600^{th} epoch in Fig. 2 (a), (b), and (c), respectively. We refer to the features before the subnet mapping (meta learner) as *original features* $\boldsymbol{x}$ (namely *Ori-Features*), and the features after the subnet mapping as *meta-features* $\phi(\boldsymbol{x})$ (namely *Meta-Features*). For the red/blue histogram marked by *Ori-Features*/*Meta-Features*, we obtain the feature distance of a sample pair (sample A, sample B) by calculating the Cosine distance (*i.e.*, One minus the Cosine similarity) between the sample A and sample B by using *original features*/*meta-features*.

We can observe that at the early training epochs (*e.g.*, the 400^{th} epoch shown in Fig. 2(a)) after the proposed MPN-tuple loss is employed, the distances of positive sample pairs calculated using meta-features are smaller than those calculated using original features, indicating the within-class similarity based on the meta-features is higher than that based on original features. The learned meta-features thus can be viewed as predictions for the identity representations from the original features by exploiting the patterns/correlations among different feature dimensions. Within the proposed MPN-tuple loss, we calculate the Cosine similarity between the original feature of one sample $\boldsymbol{x_a}$ and the meta feature $\phi(\boldsymbol{x_b})$ of another sample $\boldsymbol{x_b}$, which plays a equivalent role of learning a more general similarity metric between $\boldsymbol{x_a}$ and $\boldsymbol{x_b}$ [69]. As the training goes on, optimized by the MPN-tuple loss, the network becomes better under this learned similarity metric and makes original features approach the corresponding predicted identity representations such that the within-class similarity are further enhanced during training. We observe that the distances of positive pairs using *original features* become smaller and smaller. In the final, the discrepancy of these two distributions (red histogram and blue histogram) are quite small after 600 epochs (see Fig. 2(b) and (c)). Thus, during the inference, we can get rid of that mapping subnet and directly use original features as the ReID feature $\boldsymbol{x}$ for matching (the difference is marginal: $<$0.4%).

I. Effectiveness and Robustness Evaluation on Visible-Infrared Person ReID

Experimental Configurations. For the visible-infrared person ReID, we evaluate the effectiveness of our proposed method by performing experimental comparisons on the commonly used datasets SYSU-MM01 [70] and RegDB [71]. In each mini-batch, we randomly sample 64 images with 16 identities where there are 4 RGB images and 4 thermal images for each identities. Empirically, we adopt SGD optimizer to train the entire model for 70 epochs. The initial learning rate is 0.1 and the learning rate decay is performed after the 40^{th} and the 50^{th} epoch. Other configurations are kept the same as the description in the Section IV-B.

We adopt two different evaluation modes on RegDB [71] and SYSU-MM01 [70], respectively. For RegDB, "Visible to Thermal" refers to that visible images are taken as the query set while thermal images are taken as the gallery set, and so on. For SYSU-MM01, "All Search" mode means that all images are used for testing. And under the "Indoor Search" mode, only indoor images from 1st, 2nd, 3rd, 6th cameras are used. The table VI shows the ablation study results on visible-infrared person ReID. Our proposed MPN-tuple loss outperforms the triplet loss with hard mining (namely "TriHard.") by 2.3%/6.1% and 2.1%/6.6% for the "Visible to Thermal" and "Thermal to Visible" evaluation settings on the RegDB dataset and 4.7%/4.7% for 6.1%/7.0% on the SYSU-MM01 dataset in Rank-1/mAP, respectively. This experimental result shows that our proposed MPN-tuple loss has significant

TABLE VI
PERFORMANCE (%) COMPARISONS ON VISIBLE-INFRARED PERSON REID. WE ADOPT DIFFERENT TEST MODES ON REGDB, *i.e.*, "VISIBLE TO THERMAL" AND "THERMAL TO VISIBLE", WHERE "VISIBLE TO THERMAL" REFERS TO THAT VISIBLE IMAGES ARE TAKEN AS THE QUERY SET WHILE THERMAL IMAGES ARE TAKEN AS THE GALLERY SET, AND SO ON. THERE ARE ALSO TWO DIFFERENT MODES FOR THE SINGLE-SHOT EVALUATION ON SYSU-MM01, *i.e.*, "ALL SEARCH" MODE AND "SINGLE SEARCH" MODE. FOR "ALL SEARCH" MODE, ALL IMAGES ARE USED. AND FOR THE "INDOOR SEARCH" MODE, ONLY INDOOR IMAGES FROM 1ST, 2ND, 3RD, 6TH CAMERAS ARE USED.

Loss	RegDB				SYSU-MM01			
	Visible to Thermal		Thermal to Visible		All Search		Indoor Search	
	Rank-1	mAP	Rank-1	mAP	Rank-1	mAP	Rank-1	mAP
Cls. + Tri.	52.7	48.0	58.9	55.5	36.2	37.7	49.7	56.3
Cls. + TriHard.	68.3	62.9	67.7	62.1	43.2	42.4	51.1	58.1
Cls. + MPN-tuple	70.6	69.0	69.8	68.7	47.9	47.1	57.2	65.1

TABLE VII
PERFORMANCE (%) COMPARISONS ON CLOTH-CHANGING PERSON REID. ON LTCC DATASET, "STANDARD" REFERS TO THE EVALUATION SETTING WHERE THE IMAGES OF THE SAME IDENTITY AND CAMERA VIEW ARE DISCARDED DURING TESTING. AND "CLOTH-CHANGING" DENOTES THAT THE IMAGES WITH THE SAME IDENTITY, CAMERA VIEW AND CLOTHES ARE DISCARDED DURING TESTING. ON PRCC DATASET, "CROSS-CLOTHES" REFERS TO THE PERSON MATCHING WITH CLOTH CHANGES WHEREAS "SAME-CLOTHES" TESTING HAS NO CLOTH CHANGES.

Loss	LTCC				PRCC					
	Standard		Cloth-changing		Cross-Clothes			Same-Clothes		
	Rank-1	mAP	Rank-1	mAP	Rank-1	Rank-10	Rank-20	Rank-1	Rank-10	Rank-20
Cls. + Tri.	58.5	27.3	39.6	11.3	26.2	35.9	40.0	80.5	87.4	90.0
Cls. + TriHard.	55.4	23.7	49.5	14.0	20.8	29.6	33.0	84.7	89.8	91.8
Cls. + MPN-tuple	59.8	29.8	53.0	21.6	35.2	42.1	45.0	88.2	93.1	94.8

improvements compared to the triplet loss, indicating that MPN-tuple loss is also effective and robust for visible-infrared person ReID task.

J. Effectiveness and Robustness Evaluation on Cloth-changing Person ReID

For the cloth-changing person ReID, we evaluate the effectiveness of our proposed method by performing experimental comparisons on two most commonly used datasets, *i.e.*, PRCC [72] and LTCC [73]. In each mini-batch, we randomly sample 16 identities where each identity include 4 randomly sampled images, resulting in a batch size of 64. We adopt Adam optimizer to train the entire network for 70 epochs. During the training, we first warm up with a linear growth learning rate from 4×10^{-6} to 4×10^{-4} for 10 epochs and the learning rate is decayed by a factor of 0.5 for every 20 epochs. Other configurations are kept the same as the description in the Section IV-B.

The table VII shows performance comparisons for effectiveness evaluation. On the dataset LTCC, our proposed MPN-tuple loss achieves the Rank-1/mAP improvements of 4.4%/6.1% and 3.5%/7.6% in the "Standard" and "Cloth-changing" evaluation scenarios respectively compared to the triplet loss with hard mining (namely "TriHard."). On the dateset PRCC, MPN-tuple loss outperforms "TriHard." by 14.4% and 3.5% in the Rank-1 accuracy under the "Cross-clothes" and "Same-Clothes" evaluation scenarios respectively. This experimental result verify the robustness of our proposed MPN-tuple loss on cloth-changing person ReID.

V. CONCLUSIONS

In this paper, we reformulate prevalent loss designs (triplet loss and classification loss) under a unified form and analyze their inherent limitations for person ReID. The triplet loss can be viewed as a two-class classification. There is a lack of valid interaction between different triplets and their optimization directions may contradict each other. The classification loss optimizes the similarity/distance between instances and parameter-based category centers, which enables stable global scope optimization but does not align well with the retrieval-based person ReID inference. Furthermore, we point out that N-tuple loss can provide more consistent optimization between training and testing but is under-explored for ReID task. Moreover, we introduce MPN-tuple loss which uses a meta learner to learn better references nodes (*i.e.*, better classifier) for more effective metric learning. The scheme powered by our proposed MPN-tuple loss achieves the state-of-the-art performance. Besides, we further verify the effectiveness and robustness of our proposed MPN-tuple loss on visible-infrared and cloth-changing person ReID. And we believe it has the potentials to be applied to other vision tasks. We hope that in the future the ReID community will build on top of our strong baseline to investigate more novel loss designs.

REFERENCES

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *NeurIPS*, 2012, pp. 1097–1105.
[2] L. Zheng, H. Zhang, S. Sun, M. Chandraker, Y. Yang, and Q. Tian, "Person re-identification in the wild," in *CVPR*, 2017, pp. 1367–1376.
[3] L. Zheng, Y. Yang, and Q. Tian, "Sift meets cnn: A decade survey of instance retrieval," *TPAMI*, vol. 40, no. 5, pp. 1224–1244, 2017.
[4] L. Zheng, Y. Yang, and A. G. Hauptmann, "Person re-identification: Past, present and future," *arXiv preprint arXiv:1610.02984*, 2016.
[5] W. Li, X. Zhu, and S. Gong, "Person re-identification by deep joint learning of multi-loss classification," *arXiv preprint arXiv:1705.04724*, 2017.
[6] Y. Sun, L. Zheng, Y. Yang, Q. Tian, and S. Wang, "Beyond part models: Person retrieval with refined part pooling (and a strong convolutional baseline)," in *ECCV*, 2018, pp. 480–496.
[7] H. Yao, S. Zhang, R. Hong, Y. Zhang, C. Xu, and Q. Tian, "Deep representation learning with part loss for person re-identification," *TIP*, vol. 28, no. 6, pp. 2860–2871, 2019.
[8] Y. Zhai, X. Guo, Y. Lu, and H. Li, "In defense of the classification loss for person re-identification," in *CVPR Workshops*, 2019, pp. 0–0.

[9] G. Wang, Y. Yuan, X. Chen, J. Li, and X. Zhou, "Learning discriminative features with multiple granularities for person re-identification," in *ACM Multimedia*, 2018, pp. 274–282.

[10] Z. Zhang, C. Lan, W. Zeng, and Z. Chen, "Densely semantically aligned person re-identification," in *CVPR*, 2019, pp. 667–676.

[11] Z. Zhang, C. Lan, W. Zeng, X. Jin, and Z. Chen, "Relation-aware global attention for person re-identification," in *CVPR*, 2020.

[12] H. Lawen, A. Ben-Cohen, M. Protter, I. Friedman, and L. Zelnik-Manor, "Attention network robustification for person reid," *arXiv preprint arXiv:1910.07038*, 2019.

[13] H. Luo, W. Jiang, Y. Gu, F. Liu, X. Liao, S. Lai, and J. Gu, "A strong baseline and batch normalization neck for deep person re-identification," *TMM*, 2019.

[14] X. Bai, M. Yang, T. Huang, Z. Dou, R. Yu, and Y. Xu, "Deep-person: Learning discriminative deep features for person re-identification," *Pattern Recognition*, 2020.

[15] L. Wei, S. Zhang, H. Yao, W. Gao, and Q. Tian, "Glad: global-local-alignment descriptor for pedestrian retrieval," in *ACM Multimedia*, 2017, pp. 420–428.

[16] C. Su, J. Li, S. Zhang, J. Xing, W. Gao, and Q. Tian, "Pose-driven deep convolutional model for person re-identification," in *ICCV*, 2017.

[17] H. Zhao, M. Tian, S. Sun, J. Shao, J. Yan, S. Yi, X. Wang, and X. Tang, "Spindle net: Person re-identification with human body region guided feature decomposition and fusion," in *CVPR*, 2017.

[18] W. Li, X. Zhu, and S. Gong, "Harmonious attention network for person re-identification," in *CVPR*, 2018, pp. 2285–2294.

[19] C. Wang, Q. Zhang, C. Huang, W. Liu, and X. Wang, "Mancs: A multi-task attentional network with curriculum sampling for person re-identification," in *ECCV*, 2018, pp. 365–381.

[20] B. Chen, W. Deng, and J. Hu, "Mixed high-order attention network for person re-identification," in *ICCV*, 2019, pp. 371–381.

[21] C. Song, Y. Huang, W. Ouyang, and L. Wang, "Mask-guided contrastive attention model for person re-identification," in *CVPR*, 2018, pp. 1179–1188.

[22] M. M. Kalayeh, E. Basaran, M. Gökmen, M. E. Kamasak, and M. Shah, "Human semantic parsing for person re-identification," in *CVPR*, 2018, pp. 1062–1071.

[23] X. Jin, C. Lan, W. Zeng, G. Wei, and Z. Chen, "Semantics-aligned representation learning for person re-identification," in *AAAI*, 2020.

[24] R. R. Varior, M. Haloi, and G. Wang, "Gated siamese convolutional neural network architecture for human re-identification," in *ECCV*, 2016, pp. 791–808.

[25] R. R. Varior, B. Shuai, J. Lu, D. Xu, and G. Wang, "A siamese long short-term memory architecture for human re-identification," in *ECCV*, 2016, pp. 135–153.

[26] F. Wang, W. Zuo, L. Lin, D. Zhang, and L. Zhang, "Joint learning of single-image and cross-image representations for person re-identification," in *CVPR*, 2016, pp. 1288–1296.

[27] W. Li, R. Zhao, T. Xiao, and X. Wang, "Deepreid: Deep filter pairing neural network for person re-identification," in *CVPR*, 2014, pp. 152–159.

[28] E. Ahmed, M. Jones, and T. K. Marks, "An improved deep learning architecture for person re-identification," in *CVPR*, 2015, pp. 3908–3916.

[29] S. Khamis, C.-H. Kuo, V. K. Singh, V. D. Shet, and L. S. Davis, "Joint learning for attribute-consistent person re-identification," in *ECCV*, 2014, pp. 134–146.

[30] S. Paisitkriangkrai, C. Shen, and A. Van Den Hengel, "Learning to rank in person re-identification with metric ensembles," in *CVPR*, 2015, pp. 1846–1855.

[31] D. Cheng, Y. Gong, S. Zhou, J. Wang, and N. Zheng, "Person re-identification by multi-channel parts-based cnn with improved triplet loss function," in *CVPR*, 2016, pp. 1335–1344.

[32] A. Hermans, L. Beyer, and B. Leibe, "In defense of the triplet loss for person re-identification," *arXiv preprint arXiv:1703.07737*, 2017.

[33] S. Zhou, J. Wang, J. Wang, Y. Gong, and N. Zheng, "Point to set similarity based deep feature learning for person re-identification," in *CVPR*, 2017, pp. 3741–3750.

[34] R. Yu, Z. Dou, S. Bai, Z. Zhang, Y. Xu, and X. Bai, "Hard-aware point-to-set deep metric for person re-identification," in *ECCV*, 2018, pp. 188–204.

[35] J. Almazan, B. Gajic, N. Murray, and D. Larlus, "Re-id done right: towards good practices for person re-identification," *arXiv preprint arXiv:1801.05339*, 2018.

[36] Y. Yuan, W. Chen, Y. Yang, and Z. Wang, "In defense of the triplet loss again: Learning robust person re-identification with fast approximated triplet loss and label distillation," *arXiv preprint arXiv:1912.07863*, 2019.

[37] W. Chen, X. Chen, J. Zhang, and K. Huang, "Beyond triplet loss: a deep quadruplet network for person re-identification," in *CVPR*, 2017, pp. 403–412.

[38] P. Fang, J. Zhou, S. K. Roy, L. Petersson, and M. Harandi, "Bilinear attention networks for person retrieval," in *ICCV*, 2019, pp. 8030–8039.

[39] S. Roweis, G. Hinton, and R. Salakhutdinov, "Neighbourhood component analysis," *NeurIPS*, vol. 17, pp. 513–520, 2004.

[40] K. Q. Weinberger, J. Blitzer, and L. K. Saul, "Distance metric learning for large margin nearest neighbor classification," in *NeurIPS*, 2006, pp. 1473–1480.

[41] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *CVPR*, 2015, pp. 815–823.

[42] O. Vinyals, C. Blundell, T. Lillicrap, D. Wierstra *et al.*, "Matching networks for one shot learning," in *NeurIPS*, 2016, pp. 3630–3638.

[43] J. Snell, K. Swersky, and R. Zemel, "Prototypical networks for few-shot learning," in *NeurIPS*, 2017, pp. 4077–4087.

[44] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality reduction by learning an invariant mapping," in *CVPR*, vol. 2, 2006, pp. 1735–1742.

[45] Y. Wen, K. Zhang, Z. Li, and Y. Qiao, "A discriminative feature learning approach for deep face recognition," in *ECCV*. Springer, 2016, pp. 499–515.

[46] T.-T. Do, T. Tran, I. Reid, V. Kumar, T. Hoang, and G. Carneiro, "A theoretically sound upper bound on the triplet loss for improving the efficiency of deep distance metric learning," in *CVPR*, 2019, pp. 10 404–10 413.

[47] K. Sohn, "Improved deep metric learning with multi-class n-pair loss objective," in *NeurIPS*, 2016, pp. 1857–1865.

[48] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *CVPR*, 2018, pp. 7132–7141.

[49] L. Zheng, L. Shen, L. Tian, S. Wang, J. Wang, and Q. Tian, "Scalable person re-identification: A benchmark," in *ICCV*, 2015.

[50] Z. Zheng, L. Zheng, and Y. Yang, "Unlabeled samples generated by gan improve the person re-identification baseline in vitro," *arXiv preprint arXiv:1701.07717*, 2017.

[51] L. Wei, S. Zhang, W. Gao, and Q. Tian, "Person transfer GAN to bridge domain gap for person re-identification," in *CVPR*, 2018.

[52] Z. Zhong, L. Zheng, D. Cao, and S. Li, "Re-ranking person re-identification with k-reciprocal encoding," in *CVPR*, 2017.

[53] Z. Zheng, L. Zheng, and Y. Yang, "Pedestrian alignment network for large-scale person re-identification," *TCSVT*, 2018.

[54] L. He, Z. Sun, Y. Zhu, and Y. Wang, "Recognizing partial biometric patterns," *arXiv preprint arXiv:1810.07399*, 2018.

[55] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016.

[56] Y. Sun, L. Zheng, Y. Yang, Q. Tian, and S. Wang, "Beyond part models: Person retrieval with refined part pooling," 2018.

[57] X. Pan, P. Luo, J. Shi, and X. Tang, "Two at once: Enhancing learning and generalization capacities via ibn-net," in *ECCV*, 2018.

[58] J. Jia, Q. Ruan, and T. M. Hospedales, "Frustratingly easy person re-identification: Generalizing person re-id in practice," *BMVC*, 2019.

[59] F. Wang, X. Xiang, J. Cheng, and A. L. Yuille, "Normface: L2 hypersphere embedding for face verification," in *ACM Multimedia*, 2017, pp. 1041–1049.

[60] Y. Wang, L. Wang, Y. You, X. Zou, V. Chen, S. Li, G. Huang, B. Hariharan, and K. Q. Weinberger, "Resource aware person re-identification across multiple resolutions," in *CVPR*, 2018.

[61] Z. Zhong, L. Zheng, G. Kang, S. Li, and Y. Yang, "Random erasing data augmentation," in *AAAI*, vol. 34, no. 07, 2020, pp. 13 001–13 008.

[62] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *CVPR*, 2009.

[63] R. Hou, B. Ma, H. Chang, X. Gu, S. Shan, and X. Chen, "Interaction-and-aggregation network for person re-identification," in *CVPR*, 2019, pp. 9317–9326.

[64] C. Zhao, X. Lv, Z. Zhang, W. Zuo, J. Wu, and D. Miao, "Deep fusion feature representation learning with hard mining center-triplet loss for person re-identification," *TMM*, 2020.

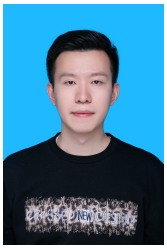
[65] K. Zhou, Y. Yang, A. Cavallaro, and T. Xiang, "Omni-scale feature learning for person re-identification," in *ICCV*, 2019, pp. 3702–3712.

[66] Z. Zheng, X. Yang, Z. Yu, L. Zheng, Y. Yang, and J. Kautz, "Joint discriminative and generative learning for person re-identification," in *CVPR*, 2019, pp. 2138–2147.

[67] L. He and W. Liu, "Guided saliency feature learning for person re-identification in crowded scenes," 2020.

[68] C. Yan, G. Pang, X. Bai, J. Zhou, and L. Gu, "Beyond triplet loss: Person re-identification with fine-grained difference-aware pairwise loss," *TMM*, 2021.

[69] S. Qiao, C. Liu, W. Shen, and A. L. Yuille, "Few-shot image recognition by predicting parameters from activations," in *CVPR*, 2018, pp. 7229–7238.
[70] A. Wu, W.-S. Zheng, H.-X. Yu, S. Gong, and J. Lai, "Rgb-infrared cross-modality person re-identification," in *ICCV*, 2017, pp. 5380–5389.
[71] D. T. Nguyen, H. G. Hong, K. W. Kim, and K. R. Park, "Person recognition system based on a combination of body images from visible light and thermal cameras," *Sensors*, vol. 17, no. 3, p. 605, 2017.
[72] Q. Yang, A. Wu, and W.-S. Zheng, "Person re-identification by contour sketch under moderate clothing change," *TPAMI*, 2019.
[73] X. Qian, W. Wang, L. Zhang, F. Zhu, Y. Fu, T. Xiang, Y.-G. Jiang, and X. Xue, "Long-term cloth-changing person re-identification," in *WACV*, 2020.

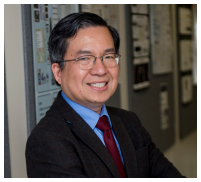


Zhizheng Zhang received the B.S. degree from University of Electronic Science and Technology of China in 2016. He received the Ph.D. degree from University of Science and Technology of China in 2021. He joined Microsoft Research in June 2021 and is now a researcher at Microsoft Research Asia. His research interests include person re-identification, image/video compression, few-shot learning and domain generalization/adaptation.

Cuiling Lan received the B.S. degree in electrical engineering and the Ph.D. degree in intelligent information processing from Xidian University, Xi'an, China, in 2008 and 2014, respectively. She joined Microsoft Research Asia, Beijing, China, in 2014. Her current research interests include computer vision problems related to pose estimation, action recognition, person/ vehicle re-identification, domain generalization/adaptation.

Wenjun Zeng (M'97-SM'03-F'12) is a Sr. Principal Research Manager and a member of the senior leadership team at Microsoft Research Asia. He has been leading the video analytics research empowering the Microsoft Cognitive Services, Azure Media Analytics Services, Office, Dynamics, and Windows Machine Learning since 2014. He was with Univ. of Missouri from 2003 to 2016, most recently as a Full Professor. Prior to that, he had worked for PacketVideo Corp., Sharp Labs of America, Bell Labs, and Panasonic Technology. Wenjun has contributed significantly to the development of international standards (ISO MPEG, JPEG2000, and OMA). He received his B.E., M.S., and Ph.D. degrees from Tsinghua Univ., the Univ. of Notre Dame, and Princeton Univ., respectively. His current research interests include mobile-cloud media computing, computer vision, and multimedia communications and security. He is on the Editorial Board of International Journal of Computer Vision. He was an Associate Editor-in-Chief of IEEE Multimedia Magazine, and was an AE of IEEE Trans. on Circuits & Systems for Video Technology (TCSVT), IEEE Trans. on Info. Forensics & Security, and IEEE Trans. on Multimedia (TMM). He was on the Steering Committee of IEEE Trans. on Mobile Computing and IEEE TMM. He served as the Steering Committee Chair of IEEE ICME in 2010 and 2011, and has served as the General Chair or TPC Chair for several IEEE conferences (e.g., ICME'2018, ICIP'2017). He was the recipient of several best paper awards. He is a Fellow of the IEEE.

Zhibo Chen (M'01-SM'11) received the B. Sc., and Ph.D. degree from Department of Electrical Engineering Tsinghua University in 1998 and 2003, respectively. He is now a professor at University of Science and Technology of China. His research interests include image and video compression, visual quality of experience assessment, immersive media computing and intelligent media computing. He has more than 150 publications and more than 50 granted EU and US patent applications. He is IEEE senior member, Secretary/Chair-Elect of IEEE Visual Signal Processing and Communications Committee, and member of IEEE Multimedia System and Applications Committee. He was TPC chair of IEEE PCS 2019 and organization committee member of ICIP 2017 and ICME 2013, served as TPC member in IEEE ISCAS and IEEE VCIP.



Shih-Fu Chang 's research is focused on computer vision, machine learning, and multimodal knowledge extraction. He received the IEEE Signal Processing Society Technical Achievement Award, ACM SIGMM Technical Achievement Award, the Honorary Doctorate from the University of Amsterdam, and the IEEE Kiyo Tomiyasu Award. He has been Interim Dean of Columbia Engineering (since 2021), an Amazon Scholar, a Fellow of AAAS, ACM, and IEEE, and an elected member of Academia Sinica.