Multi-pseudo Regularized Label for Generated Samples in Person Re-Identification

Yan Huang, Jinsong Xu, Qiang Wu, *Member, IEEE* Zhedong Zheng, Zhaoxiang Zhang, *Senior Member, IEEE* and Jian Zhang, *Senior Member, IEEE*

***Abstract*—Sufficient training data is normally required to train deeply learned models. However, the number of pedestrian images per ID in person re-identification (re-ID) datasets is usually limited, since manually annotations are required for multiple camera views. To produce more data for training deeply learned models, generative adversarial network (GAN) can be leveraged to generate samples for person re-ID. However, the samples generated by vanilla GAN usually do not have labels. So in this paper, we propose a virtual label called Multi-pseudo Regularized Label (MpRL) and assign it to the generated images. With MpRL, the generated samples will be used as supplementary of real training data to train a deep model in a semi-supervised learning fashion. Considering data bias between generated and real samples, MpRL utilizes different contributions from pre-defined training classes. The contribution-based virtual labels are automatically assigned to generated samples to reduce ambiguous prediction in training. Meanwhile, MpRL only relies on pre-defined training classes without using extra classes. Furthermore, to reduce over-fitting, a regularized manner is applied to MpRL to regularize the learning process.**

To verify the effectiveness of MpRL, two state-of-the-art convolutional neural networks (CNNs) are adopted in our experiments. Experiments demonstrate that by assigning MpRL to generated samples, we can further improve the person re-ID performance on three datasets *i.e.*, Market-1501, DukeMTMC-reID, and CUHK03. The proposed method obtains +6.29%, +6.30% and +5.58% improvements in rank-1 accuracy over a strong CNN baseline respectively, and outperforms the state-of-the-art methods.

***Index Terms*—person re-identification, generated samples, virtual label, semi-supervised learning.**

I. Introduction

In 2014s, generative adversarial network (GAN) was proposed to generate samples (images) with perceptual quality [18]. Since then, several improved approaches [39], [3] were presented to further improve the quality of generated samples by leveraging the ability of GAN. However, how to use these generated samples is still an open question. Meanwhile, person re-identification (re-ID) is a challenging task of recognizing a person at different camera views. It is a typical computer vision problem that requires sufficient training data to learn a discriminative model. In the past few years, deep learning has demonstrated its performance in person re-ID by producing several state-of-the-art methods [22], [38], [33], [55], [56], [58]. To this end, sufficient labeled training data is essential to train deeply learned models in a supervised learning fashion. Some datasets *i.e.*, Market-1501 [54], DukeMTMC-reID [59], CUHK03 [28], *etc.* have been proposed to remedy the limited training data problem. However, due to the expensive cost of data acquisition that needs to manually find corresponding labels for pedestrians who appear under different camera views, the number of images per ID in these datasets is still limited.

Using generated samples to remedy the lack of training data is a promising solution. Therefore, we attempt to use unlabeled samples generated by GANs to further improve the person re-ID performance. Rather than investigating how to enhance the quality of generated samples [39], [3], our research will focus on how to use GANs to promote the performance of classifiers. We follow the same pipeline in [59] that incorporates generated samples with real samples to train deep models in a semi-supervised learning fashion. Compared with previous attempts [40], [39] that perform semi-supervised learning in the discriminator of GANs, sufficient unlabeled generated samples will directly participate in training as supplementary of limited labeled real samples in our work.

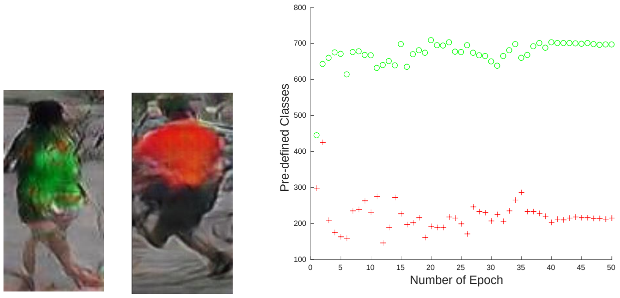
In 2017s, a related work was first proposed in [59] that introduced a method called label smooth regularization for outliers (LSRO). This method assigns virtual labels to generated samples with a uniform label distribution over all the pre-defined training classes. The uniform distribution considers contributions from all the pre-defined classes equally. More specifically, if the number of pre-defined class is K, the contribution of each class is equally divided into $1/K$. Under this design, LSRO shows two undesirable characteristics: 1) in the generated sample space, all the generated samples share the same virtual label, and 2) in the real sample space, the contributions from all pre-defined classes are equally considered for generated samples.

For the first characteristic, if generated samples with significant visual differences are fed into the network, ambiguous predictions may happen in training. Figure 1(a) and Figure 1(b) show two generated samples with red and green clothes, respectively. If we fit these two samples into the pre-defined classes (only using the maximum predicted probability) over 50 training epochs, distinguishable label distribution can be

Yan Huang, Jinsong Xu, Qiang Wu and Jian Zhang are with the Global Big Data Technologies Centre (GBDTC), School of Electrical and Data Engineering, University of Technology Sydney, Australia. (Email: Yan.Huang-3@student.uts.edu.au, JingSong.Xu@uts.edu.au, Qiang.Wu@uts.edu.au and Jian.Zhang@uts.edu.au)

Zhedong Zheng is with the Centre for Artificial Intelligence (CAI), School of Software, University of Technology Sydney, Australia. (Email: Zhedong.Zheng@student.uts.edu.au)

Zhaoxiang Zhang is with the Research Center for Brain-Inspired Intelligence, CAS Center for Excellence in Brain Science and Intelligence Technology, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China (e-mail: zhaoxiang.zhang@ia.ac.cn).



(a)

(b)

(c) Fitting generated samples (a) and (b) on pre-defined classes with 'o' and '+' marker, respectively.

Fig. 1. Pre-defined class label distribution (c) for generated samples (a) and (b). Only the maximum predicted probability of pre-defined classes is activated in each epoch (see (c)). Distinguishable label distributions can be observed between these two samples.

observed in Figure 1(c). Therefore, using the same virtual label over all the generated samples is improper. **We need to assign different label for every generated sample.** For the second characteristic, if a uniform label distribution is used, any generated sample should have the property of simulating all distributions of pre-defined classes equally. However, in regard of the training phase of a GAN, only a random batch of real samples are simulated in each iteration by using limited input points amongst a continuous noise distribution [18], [39]. Consequently, the data distribution between generated samples and real samples is biased. Under this bias, equally utilizing the contributions from all pre-defined classes in the real sample space is inappropriate when assigning virtual labels to generated samples. **We need to dynamically assign the label distribution over all pre-defined classes based on their contributions.**

Although LSRO has demonstrated its effectiveness in [59], the above problems still limit its efficiency. To this end, a Multi-pseudo Regularized Label (MpRL) is proposed as virtual label assigned to generated samples. We attempt to overcome the above problems for designing MpRL. The main contributions of the proposed MpRL can be summarized in three-fold:

- To reduce ambiguous predictions in training, we assign each generated sample with a contribution based virtual label by using MpRL according to different predicted probabilities from pre-defined training classes. Meanwhile, rather than using any extra class, MpRL is only based on pre-defined training classes.
- Due to the biased distribution between generated samples and real samples, a dictionary that records contributions amongst pre-defined classes is built in MpRL. For a generated sample, all predicted probabilities are iteratively retained to produce contributions in the dictionary in training. The dictionary will be dynamically updated to affect the affiliation of a generated sample on pre-defined classes. Furthermore, a regularized manner is used to regularize the learning process to reduce over-fitting.
- We adopt two different convolutional neural networks (CNNs) for person re-ID to evaluate the proposed MpRL. Comparing with the state-of-the-art method LSRO [59], a further improvement in performance is obtained. Moreover, we also verify the MpRL is robust to the generalization performance of CNNs by changing the batchsize in training [24].

This paper is organized as follows. We first review some related works in Section II. In Section III, we begin to revisit the state-of-the-art virtual label used on generated samples. Then the implementation details of the proposed MpRL are provided. A brief analysis is discussed to demonstrate why MpRL works better. In Section IV, we introduce how to use GAN to generate samples, and two different CNNs for person re-ID are presented to evaluate MpRL. The experiments are shown in Section V. Conclusion is in Section VI.

II. Related Work

In this section, we will review existing work related to semi-supervised learning and person re-ID.

A. *Semi-supervised Learning*

Semi-supervised learning is halfway between supervised and unsupervised learning, which uses both the labeled data and unlabeled data to perform the learning task. It has been well investigated and dozens of methods have been proposed in the literature. In image segmentation, a small number of strongly annotated images and a large number of weakly annotated images are incorporated to perform semi-supervised learning [20], [11]. For person identification in TV series, Bauml *et al.* [5] take labeled data and unlabeled data into account, and constrain them in a joint formulation. To tackle multi-label image classification, Luo *et al.* [35] make use of unlabeled samples in semi-supervised learning to boost performance. In text classification, a region embedding is learned from unlabeled data to produce additional input in CNN [23].

Since obtaining training labels is expensive, previous semi-supervised works focus on how to utilize sufficient unlabeled data with labeled data to boost the performance. However, if the number of samples is scarce or hard to obtain, traditional semi-supervised methods may useless. In this paper, instead of using unlabeled data from the real sample space, we directly use existing data to generate unlabeled samples by GAN. Further, we would like to show that these generated samples can help improve discriminative model learning by adding our virtual label MpRL.

Besides, several methods have attempted to assign virtual labels to the unlabeled data in semi-supervised learning. In [37], [41], a new class in the discriminator is taken as a virtual label (all-in-one) to all the unlabeled samples produced by the generator of the GAN. The all-in-one method simply regards all generated samples as an extra class. Let K represents the number of pre-defined training classes in the real sample space, then $K+1$ is assigned to each generated sample. Since these samples are generated according to the distribution of real samples, they tend to belong to the pre-defined classes rather than a new class. To this end, the one-hot pseudo label [26] can assign a virtual label to them without using extra class. The one-hot pseudo label utilizes the maximum predicted probability of an unlabeled sample in the pre-defined training classes as its virtual label. In training, this virtual label is dynamically assigned to the unlabeled sample, so that the same sample may receive a different label each time when it is fed into the network. Using the one-hot pseudo label, after several training epochs, one generated sample will be fitted to a specific pre-defined class that may lead to over-fitting. To address this problem, Zheng *et al.* [59] introduce a virtual label LSRO that uses a uniform label distribution on generated samples to regularize the network training for person re-ID.

In this paper, the all-in-one [37], [41], one-hot pseudo [26], and LSRO [59] will be used as contrast experiments. Amongst them, LSRO is the state-of-the-art virtual label for person re-ID. In particular, we call the pseudo [26] as one-hot pseudo in this paper since only one pre-defined class with the maximum predicted probability is activated in each training iteration.

B. Person Re-isdentification

The person re-ID is selected to evaluate our MpRL based on two reasons. Firstly, in the past five years, there has been a tremendous increase of computer vision research on solving the person re-ID problem, which has drawn growing interest from academic research to practical applications [17]. Secondly, compared with other computer vision task, acquiring labeled real samples is expensive for person re-ID. This problem inspires us to leverage generated samples by GAN to remedy the limited training data. For person re-ID, two branches, including traditional methods and deep learning methods, have demonstrated their performance.

In traditional methods, the task of person re-ID can be divided into two modules: feature extraction and metric learning. In feature extraction, Liao *et al.* [31] propose local maximal occurrence feature to against viewpoint changes and handle illumination variations. Chen *et al.* [8] introduce a mirror representation to alleviate the view-specific feature distortion problem. Zheng *et al.* [54] present a bag-of-words descriptor that describes each person by visual word histogram. In metric learning, Zheng *et al.* [57] use a relative distance comparison method to minimize the probability of a negative pair that have a larger distance than a positive pair. Liao *et al.* [32] propose a logistic metric learning via an asymmetric sample weighting strategy. Li *et al.* [30] employ a locally-adaptive decision function that integrates traditional metric learning with a local decision rule. Yu *et al.* [50] learn an asymmetric metric that projects each view in unsupervised learning fashion.

Unlike the above traditional methods that are manually designed to handle the person re-ID task explicitly. Deep learning discovers more implicit information in matching persons has been widely used in producing many state-of-the-art results.

In deep learning methods, to distinguish person appearance at the right spatial locations and scales, Qian *et al.* [38] propose a multi-scale deep learning model to learn discriminative features for matching persons. Lin *et al.* [33] introduce a consistent-aware deep learning approach that seeks the globally optimal matching for the whole camera network. Also, deep features over the full body and body parts are captured from local context knowledge by stacking multi-scale convolutions in [27]. Moreover, two-stream network [16], [58], triplet loss network [12], [10] and quadruplet network [7] have been designed to address person re-ID in the past few years.

In [55], [56], Zheng *et al.* use an identification (Identif) CNN for person re-ID. This Identif network takes person re-ID as a multi-classification task, and a CNN embedding is learned to discriminate different identities in training. Beyond that, Zheng *et al.* [58] propose a Two-stream deep network. A verification function that separates two input samples belong to the same or different identities is considered in the Two-stream network to further improve the performance of the Identif network. In testing, the above two networks extract CNN embeddings in the last convolutional layer to compare the similarity between two inputs using squared Euclidean distance. The two networks produce state-of-the-art results for person re-ID, and both of them are utilized in [59] to investigate the improvement by adding generated samples with LSRO virtual label in training.

In this work, we adopt the Identif network [55], [56] and the Two-stream network [58] to verify the effectiveness of the proposed MpRL. Compared with the previous related work, our MpRL can further improve the performance of the two networks.

Boosting. In previous works, some methods also have been proposed as a procedure to further boost person re-ID performance. Like the boosting process, we attempt to improve the performance of two different CNNs by using MpRL. Huang *et al.* [21] formulate person re-ID as a tree matching problem, and a complete bipartite graph matching is presented to refine the final matching result at the top layer of the tree. To study person re-ID with manifold-based affinity learning, Bai *et al.* [4] introduce a manifold-preserving algorithm plunges into existing re-ID algorithms for enhancing the accuracy. Reranking that exploits the relationships amongst initial ranking list in person re-ID have been studied to improve performance [60], [15], [14]. Finally, human feedback in-the-loop is required that provides an instant improvement to re-ID ranking on-the-fly [2], [48], [34].

Unlike the above attempts, in this work, we propose to use generated samples to boost person re-ID performance on off-the-shelf CNNs by incorporating with MpRL. Although our main contribution is not to produce state-of-the-art person re-ID results. We also try to boost the Two-stream network [58] to achieve the state-of-the-art performance by using MpRL. Moreover, amongst the above mentioned boosting methods, we also leverage the state-of-the-art re-ranking method [60] to further improve the final performance.

III. The Proposed Multi-pseudo Regularized Label

In this section, we first revisit the state-of-the-art virtual label LSRO [59] for person re-ID. Inspired by this method, MpRL is proposed to solve its limitations and produces better performance. In the end, a brief analysis is given to demonstrate why MpRL works better.

A. LSRO for Person Re-ID Revisit

LSRO assumes the generated samples do not belong to any pre-defined class and uses the uniform label distribution on each of them to address over-fitting [59]. LSRO is inspired by label smoothing regularization (LSR) [44] which assigns small values to the no-ground truth classes instead of zero. Through this design, the network encourages not to be too confident towards the ground truth. Formally, for a generated image, its class label distribution $q_{LSRO}(k)$ defined as follows:

$$q_{LSRO}(k) = \frac{1}{K}, \tag{1}$$

where K is the number of pre-defined training classes in the real sample space, $k \in \{1, 2, ..., K-1, K\}$ represents the k-th pre-defined class. In training, the loss of LSRO on a generated sample is defined as follows:

$$l_{LSRO} = -\frac{1}{K}\sum_{k=1}^{K} log(p(X_k)), \tag{2}$$

where X_k represents the output for k-th pre-defined class, $p(X_k) \in (0, 1)$ is the softmax predicted probability of X_k belonging to the pre-defined class k, defined as follows:

$$p(X_k) = \frac{e^{X_k - X_{max}}}{\sum_{j=1}^{K} e^{X_j - X_{max}}}, \tag{3}$$

in which X_{max} represents the maximum output value in vector X.

- In Eq.2, the forward loss is as follows:

$$\begin{aligned} l_{LSRO} &= -\frac{1}{K}\sum_{k=1}^{K} log(\frac{e^{X_k - X_{max}}}{\sum_{j=1}^{K} e^{X_j - X_{max}}}) \\ &= -\frac{1}{K}\sum_{k=1}^{K}(X_k - X_{max}) + log(\sum_{j=1}^{K} e^{X_j - X_{max}}). \end{aligned} \tag{4}$$

- While, the backward gradient is as follows:

$$l^{'}_{LSRO} = -\frac{1}{K} + \frac{e^{X_k - X_{max}}}{\sum_{j=1}^{K} e^{X_j - X_{max}}}. \tag{5}$$

B. Multi-pseudo Regularized Label

Like LSRO, we use MpRL to assign virtual labels to each generated sample when it is fed into the network, and all the pre-defined classes are retained on MpRL. However, unlike LSRO, we do not set the the virtual label as a uniform distribution over all pre-defined classes (*i.e.*, $1/K$). The contributions from all pre-defined classes are considered different in MpRL. In this way, a dictionary α is built to record the contributions. Compared with the LSRO (see Eq.1), for a generated sample, its label distribution is defined as follows:

$$q_{MpRL}(k) = \frac{\alpha_k}{K}, \tag{6}$$

where α_k is the contribution from k-th pre-defined class in the dictionary α. The reason why different contributions are considered in MpRL will be introduced in Section III-C3. MpRL does not belong to any pre-defined training classes, but is constituted by different contributions on each of them. We define α_k according to the predicted probability from the current trained network:

$$\alpha_k = \Phi(p(X_k), sort_{min\to max}(p(X))), \tag{7}$$

where $p(X_k)$ is defined in Eq.3, $sort_{min\to max}(\cdot)$ sorts all the predicted probability of $p(X)$ from the minimum to maximum value into a sorted list, $\Phi(\cdot)$ returns the index of $p(X_k)$ to α_k in this list. Finally the corresponding relationships between pre-defined classes and its contributions are obtained.

We call our method 'multi-pseudo' label because compared with the one-hot pseudo label that only the maximum predicted probability is activated, all the predicted probabilities are used in MpRL. To address over-fitting (after several training iterations some contributions from pre-defined classes may become more significant and others may become insignificant.), a regularized manner is used in MpRL to make sure the gap between two contiguous contributions is not too big. In MpRL, the gap is $1/K$ by combining Eq.6 and Eq.7. Moreover, we keep the nature of LSRO that retains a multiple distributions over all pre-defined classes, even though some of them may not or just have an insignificant contribution on a generated sample.

Combining the generated and real samples together in training, according to Eq.3, Eq.6 and Eq.7, we define the cross-entropy loss of MpRL as follows:

$$l_{MpRL} = -(1-y)log(p(X_c)) - y\cdot\sigma\sum_{k=1}^{K}\frac{\alpha_k}{K}\cdot log(p(X_k)), \tag{8}$$

where c represents the truth label of a real sample, $\frac{\alpha_k}{K}$ is defined in Eq.7, σ is a normalization factor. In Eq.8, if we sum up contributions over K per-defined classes ($\sum_{k=1}^{K}\frac{\alpha_k}{K}$), the total contribution equals to $\frac{(1+K)\cdot K}{2}$. Therefore, to normalize contributions over K pre-defined classes, σ is set to $\frac{2}{1+K}$.

For a real sample $y = 0$, Eq.8 is equivalent to softmax loss. For a generated sample $y = 1$, only the MpRL is considered. Overall, the network has two types of losses: one for real samples and the other for generated samples.

- In Eq.8, the forward loss is as follows:

For a real sample, $y = 0$:

$$\begin{aligned} l_{MpRL} &= -log(\frac{e^{X_c - X_{max}}}{\sum_{j=1}^{K} e^{X_j - X_{max}}}) \\ &= -X_c + X_{max} + log(\sum_{j=1}^{K} e^{X_j - X_{max}}). \end{aligned} \tag{9}$$

For a generated sample, $y = 1$:

$$\begin{aligned} l_{MpRL} = -\sigma\sum_{k=1}^{K}\alpha_k \cdot log(\frac{e^{X_k - X_{max}}}{\sum_{j=1}^{K} e^{X_j - X_{max}}}) = \\ -\sigma(\sum_{k=1}^{K}\alpha_k(X_k - X_{max}) - \sum_{k=1}^{K}\alpha_k log(\sum_{j=1}^{K}(e^{X_j - X_{max}}))). \end{aligned} \tag{10}$$

- While, the backward gradient is as follows:

For a real sample, $y = 0$:

$$l^{'}_{MpRL} = -1 + \frac{e^{X_c - X_{max}}}{\sum_{j=1}^{K} e^{X_j - X_{max}}}. \tag{11}$$

For a generated sample, $y = 1$:

$$l'_{MpRL} = (-\sigma \cdot \alpha_k + \frac{e^{X_k - X_{max}}}{\sum_{j=1}^{K}(e^{X_j - X_{max}})}). \qquad (12)$$

Implementations. To demonstrate the effectiveness of MpRL, three different implementations are introduced. Descriptions are as follows:

- **Static MpRL (sMpRL).** The sMpRL is assigned to each generated sample before starting training the network. We use a pre-trained Identif network (see Section IV-B) to assign MpRL. This implementation is similar to the LSRO except that we consider different contributions on pre-defined classes instead of regarding them equally.
- **Dynamic MpRL-I (dMpRL-I): Dynamically Update MpRL from the scratch.** During training, MpRLs are dynamically assigned to each generated sample. Therefore, the same generated sample may receive a different MpRL each time when it is fed into the network. This dynamic progress is started from the start point of training. So, each generated sample will receive a random MpRL in the first training iteration.
- **Dynamic MpRL-II (dMpRL-II): Dynamically Update MpRL from the intermediate point.** We try to assign MpRLs after 20 epochs when the CNN becomes relatively stable. Also, the global weight of the loss for generated samples set to 0.1 and 1 to the real samples. That is, in Eq.8 $y = 0$, and until after 20 epochs it is set to 1. Therefore, σ becomes to $\frac{0.2}{1+K}$ according to the different weight between real and generated samples. The detailed training phase of the proposed implementation is shown in Algorithm 1.

Algorithm 1: The training phase for dMpRL-II: dynamically update MpRL from the intermediate point.

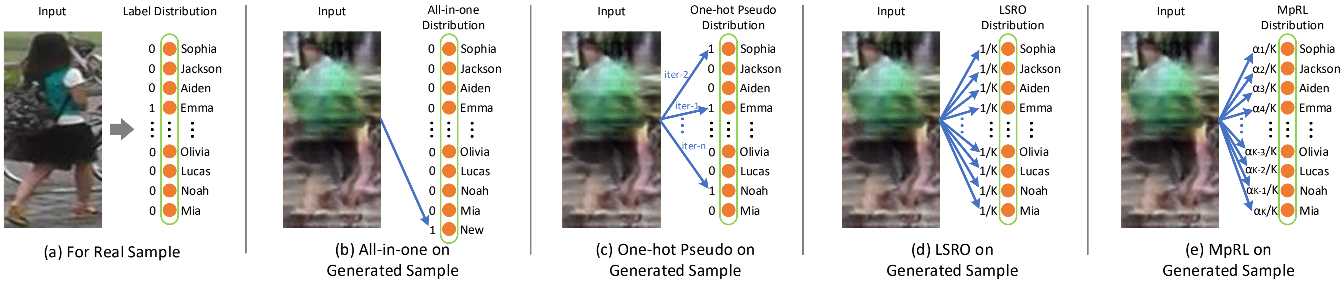
```
Input: Real sample set: $R$;
       Generated sample set: $G$;
       Merged sample set: $D = R \cup G$;
       Loss for real sample set: $l1$;
       Loss for generated sample set: $l2$.
for number of training epochs do
    Shuffle $D$ ;
    for number of training iterations in each epoch do
        Set $l1 = 0$, $l2 = 0$;
        Sample minibatch from $D \to D'$;
        Select real samples $R'$ from $D'$;
        Set $y = 0$ in Eq.8;
        Calculate loss $l1$ for $R'$;
        if number of epochs $\geq$ 20 then
            Select generated samples $G'$ from $D'$;
            Assign MpRL to $G'$ using Eq.6 and Eq.7;
            Set $y = 1$ in Eq.8;
            Calculate loss $l2$ for $G'$;
        Calculate the final loss = $l1$ + $l2 \times 0.1$ ;
        Backward propagation;
        Update parameters;
final;
```

C. Why Multi-pseudo Regularized Label Works Better?

We use the all-in-one [37], [41], one-hot pseudo [26] and LSRO [59] as our contrast experiments. Figure 2(b), Figure 2(c) and Figure 2(d) respectively illustrate the label distribution of them. Given a generated sample, a new label that do not belong to any pre-defined class is assigned to it by using the all-in-one (see Figure 2(b)). With the one-hot pseudo, a generated sample only receives the maximum predicted probability as a label in each training iteration (see Figure 2(c)). A uniform label distribution $1/K$ is assigned to a generated sample by utilizing LSRO (see Figure 2(d)). The label distribution of MpRL is illustrated in Figure 2(e), and $\alpha = \{\alpha_k | k \in \{1, 2, ...K-1, K\}\}$ represents the dictionary that used to record the different contributions over all the pre-defined classes on a generated sample. In this section the differences between MpRL and the other three virtual labels are illustrated. Three aspects are discussed: using the one-hot vs. multiple label distribution, using the same vs. different virtual labels on generated samples and using the same vs. different contributions from pre-defined classes. Three reasons are given to support MpRL, while corresponding numerical evidence will be provided in experiments (see Section V).

1) Using The One-hot vs. Multiple Label Distribution: The all-in-one and one-hot pseudo are two standard one-hot labels that assign a virtual label to each generated sample from outside (a new class) and inside (picked up from one pre-defined class) pre-defined classes, respectively. Compared with the multiple label distribution that retains all the pre-defined classes information, the one-hot distribution may suffer from a disadvantage in reducing over-fitting. Compared with the multiple label distribution, the one-hot label distribution may lack the power of regularization in training, which is important to address over-fitting. In the one-hot distribution, the network may mislead to learn a discriminative feature on an infrequent sample or class. While using multiple distributions, the network will discourage to be tuned towards one particular class and thus reduces the chances of over-fitting [44], [59]. Therefore, we design our MpRL in a multiple label distribution. In Section V-F, corresponding experiments are given to demonstrate the superiority of the multiple label distribution over the one-hot label distribution.

2) Using The Same vs. Different Virtual Labels on Generated Samples: Two strategies can be considered to assign virtual labels to generated samples: 1) using the same virtual label over all the generated samples, or 2) assigning different virtual labels to generated samples. Both all-in-one and LSRO follow the first strategy, while one-hot pseudo and MpRL go with the second one. Compared with the second strategy, assigning each generated with the same label could lead to ambiguous predictions in training. In Figure 3, four generated samples with significant visual differences (in red, yellow, white and green clothes) are given to find their top ten nearest representations which represent different pre-defined classes in the real sample space. The four groups of representations



(a) For Real Sample

(b) All-in-one on Generated Sample

(c) One-hot Pseudo on Generated Sample

(d) LSRO on Generated Sample

(e) MpRL on Generated Sample

Fig. 2. The label distribution of real and generated samples. The ground truth label is assigned to the real sample (a). For a generated sample, all-in-one (b) assigns a new label to it. One-hot pseudo (c) uses only one pre-defined class with maximum predicted probability to assign label in each training iteration. LSRO (d) uses a uniform label distribution on it, while MpRL (e) considers the different contributions over all pre-defined classes by using a dictionary $\alpha = \{\alpha_k | k \in \{1, 2, ...K-1, K\}\}$.

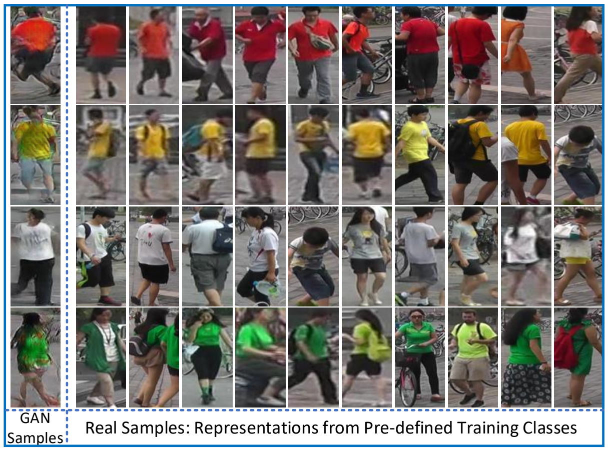

GAN Samples
Real Samples: Representations from Pre-defined Training Classes

Fig. 3. Examples of generated samples and their corresponding representations in the real sample space. The left side shows four generated samples with significant visual differences (in red, yellow, white and green clothes). For each generated sample, the right side gives ten nearest representations which represent each pre-defined class in the real sample space. Clear visual differences are shown amongst the four groups.

visually show clear differences. Considering that, if we still train a network by adding the four generated samples with the same virtual label, the network will mislead in identifying them. Therefore, we follow the second strategy that assigns each generated sample with a contribution based virtual label according to different predicted probabilities in MpRL. Corresponding experiments can be found in Section V-F and V-G to show that by assigning different labels, the proposed MpRL can produce better results.

3) Using The Same vs. Different Contributions From Pre-defined Classes: LSRO assumes that the contribution from each pre-defined class should be equal. Under this design, a generated sample should have the property to equally simulate the distribution of all the pre-defined classes. However, this is impractical when considering the generation process of GAN (details can be found in [18], [39]). This is mainly for two reasons. Firstly, in the training phase of the GAN, a minibatch of random noise is fed into a generator to simulate another minibatch of the real sample in each training iteration. This process indicates that the simulation ability of each input noise point is limited in a small scope, specifically, within a minibatch of the real sample. Secondly, normally the input random noise obeys a continuous distribution, *e.g.*, Gaussian distribution, while the distribution of real samples is discrete. Consequently, complete mapping does not exist between the two types of distributions. Due to the above two reasons, bias exists between distributions of generated and real samples. Therefore, a generated sample does not have the ability to simulate the distribution of all real samples. Let alone it considers the same contribution from each pre-defined class.

TABLE I
COMPARISON OF PROPERTIES AMONGST VIRTUAL LABELS INCLUDING: ALL-IN-ONE, ONE-HOT PSEUDO, LSRO AND THE PROPOSED MPRL.

	Label Distribution	Label Assigning	Contributions on Pre-defined Classes
All-in-one [37], [41]	One-hot	Same	–
Pseudo [26]	One-hot	Different	–
LSRO [59]	Multiple	Same	Same
MpRL (ours)	Multiple	Different	Different

To address the problem of LSRO, MpRL uses different contributions from pre-defined classes (see Section III-B). A dictionary α is used to record the contributions in training (see Eq.6). In our experiment, we observe that MpRL can outperform the state-of-the-art label LSRO on three person re-ID datasets (see Section V-F). Meanwhile, we also verify that the proposed MpRL can always further improve the performance of CNNs with different batchsize in training [24], while the LSRO may fail (see Section V-G).

Through the above analyses, Table I summaries the properties between MpRL and other labels. Our MpRL takes the advantages of all the properties and produce better performance than others. The numerical evidence which shows the superiority of MpRL will be presented in Section V.

IV. Network Overview

This section will present the structure of the GAN and CNNs used in this paper in details.

A. GAN for Generated Samples

GAN simultaneously trains two models: a generator that simulates the distribution of real samples, and a discriminator that estimates the probability that a sample comes from the real sample set rather than generator [18]. To generate samples, we use the DCGAN in [39] and follow the same settings in [59].

For the generator, a 100-dim random noise is fed into a linear function to produce a tensor with size $4 \times 4 \times 16$. Then, five deconvolutional functions with a kernel size 5×5 and a stride of 2 are used to enlarge the tensor. A rectified linear unit and batch normalization are used after each deconvolution. Also, one deconvolutional layer with a kernel size 5×5 and a stride of 1 are added to fine-tune the result followed with a tanh activation function. Finally, $128 \times 128 \times 3$ sized samples can be generated. The input of the discriminator includes the generated and real samples. Five convolutional layers are used to classify whether the generated sample is fake with a kernel size 5×5 and a stride 2. In the end, a fully-connected layer is added to perform the binary classification.

B. CNN for Evaluation

We adopt two CNNs to evaluate the proposed MpRL. These two networks have been used to evaluate the performance of the all-in-one, one-hot pseudo and LSRO labels in [59]. The first is an Identif network [55], [56] that takes person re-ID as a multi-classification task according to the number of pre-defined classes in the real sample space. We use the Identif network as a baseline when only the real samples are used. Furthermore, to compare the promotion ability amongst different virtual labels, generated samples are incorporated with real samples as the input of the Identif network. The second one is a Two-stream network [58] that combines the Identif network with a verification function to train the network. Given two input samples, the verification function could separate them into two classes that belong to the same or different identities. We use this Two-stream network to achieve better results by adding the generated samples in training. In our experiment, both Identif and Two-stream networks use the pre-trained resnet-50 [19] as a basic component. We change the last fully-connected layer to have K neurons to predict K classes, where K is the number of pre-defined classes. Since we do not need to add extra classes on generated samples by using the proposed MpRL, the last fully-connected layer remains K neurons in training.

Figure 4(a) and Figure 4(b) respectively show the Identif and Two-stream networks. MpRLs are assigned to generated samples when they are fed into the network. The loss between the predicted probability (the output of the final fully-connected layers with K neurons) and the MpRL is calculated by Eq.8. In the Two-stream network, squared Euclidean distance is used as a similarity measure between two outputs of the K neurons, and weights are shared between the two resnet-50 components. In particular, since generated samples are unlabeled data that do not belong to any classes, only real samples participate in the verification function.

V. Experiments

In our experiment, we mainly evaluate the proposed MpRL using the Market-1501 [54] dataset. We also report our results on the DukeMTMC-reID [59] and the CUHK03 [28] datasets.

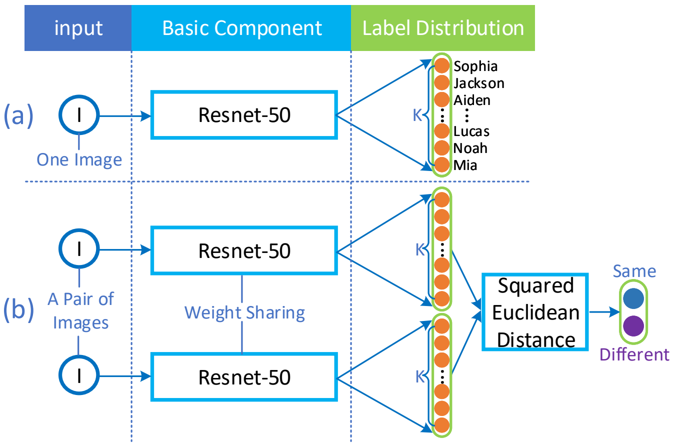


Fig. 4. (a) is the Identif network presented in [55], [56], (b) is the Two-stream network introduced in [58]. Both networks use resnet-50 as a basic component of CNN.

A. Person re-ID Datasets

Market-1501 is collected from six cameras in Tsinghua University. It contains 12,936 training samples and 19,732 testing images. The number of identities are 751 and 750 in the training and testing sets, respectively. An average of 17.2 training identity sample is in the set. All the samples are detected by the deformable part model (DPM) [13]. Both single and multiple query settings are used.

DukeMTMC-reID is collected from eight cameras. The original dataset is used for cross-camera multi-target pedestrian tracking. In this work, we use the re-ID version benchmark [59] to evaluate our method. It contains 1,404 identities in which 702 identities for training and the remaining 702 identities for testing. The total training samples are 16,522. In the testing set, one query sample for each identity is picked up in each camera and put the remaining samples in the gallery. There are 2,228 query samples and 17,661 gallery samples for the other 702 testing identities.

CUHK03 is captured by two cameras on the CUHK campus. It contains 14,097 images of 1,467 identities. There is an average of 9.6 training identity images in this set. CUHK03 contains two image settings: one is annotated by hand-drawn bounding boxes, and the other is produced by the DPM [13]. We use the detected and single shot setting. The result is reported by the averaged value after training/testing 20 times.

B. Experimental Setup

The GAN model. Tensorflow [1] and DCGAN package are used to train the GAN model using the real samples in the training set. All the images are resized to 128×128 and randomly flipped before training. The adam stochastic optimization [25] is used with parameters $\beta1 = 0.5, \beta2 = 0.99$. The training stops after 30 epochs. During testing, a 100-dim random vector ranged in [-1, 1] with Gaussian noise distribution is fed into the GAN to generate a person image. Finally, all the generated samples are resized to 256×256 and will be used in training CNNs with the proposed MpRL.

Figure 5 illustrates the generated and real samples on these three datasets. Although the generated samples can be easily

TABLE II
Performance of the Identif and Two-stream networks. Only the real samples are used. Rank-1 accuracy and mAP are listed.

Dataset	CNN	mAP	rank-1
Market-1501	Identif [55], [56]	52.68%	74.08%
	Two-stream [58]	64.09%	81.83%
DukeMTMC-reID	Identif [55], [56]	42.20%	61.94%
	Two-stream [58]	51.04%	72.62%
CUHK03	Identif [55], [56]	68.36%	63.10%
	Two-stream [58]	85.20%	81.88%

recognized as fake by a human, they remain effective in promoting the performance by adding the MpRL as virtual labels in our experiment.

CNNs. The Matconvnet [46] package is used to implement the Identif network and the Two-stream network. During training, the resnet-50 [19] is used as a basic component for both networks. All the images are resized to 256×256 before being randomly cropped into 224×224 with random horizontal flipping. A dropout layer is inserted before the final convolutional layer of resnet-50. The dropout rate is set to 0.75 for Market-1501 and DukeMTMC-reID, and 0.5 for CUHK03. We modify the fully-connected layer of resnet-50 to have 751, 702 and 1,367 neurons for Market-1501, DukeMTMC-reID and CUHK03, respectively. For the verification function in the Two-stream network, a dropout layer with a rate 0.9 is adopted after the similarity measure. Stochastic gradient descent is used on both networks with momentum 0.9. The learning rate is set to 0.1 and decay to 0.01 after 40 epochs, and we stop training after the 50th and 60th epochs on the Identif network and Two-stream network, respectively. The batchsize is 64 to the Identif network and 32 to the Two-stream network. During testing, for both networks, a 2,048-dim CNN embedding in the last convolutional layer of resnet-50 is extracted. The similarity between two images is calculated by a squared Euclidean distance before ranking.

C. The CNN Performance

Using the experimental setup in Section V-B, we train the Identif and Two-stream networks on Market-1501, DukeMTMC-reID and CUHK03, respectively. Table II shows the experimental results using the real samples only. With the Identif (Two-stream) network, we obtain the rank-1 accuracy 74.08% (81.83%), 61.92% (72.62%) and 63.10% (81.88%) on Market-1501, DukeMTMC-reID and CUHK03, respectively. The result shown in Table II is a baseline and our goal is to promote the performance of these two networks by using MpRL with generated samples in training.

D. Generated Samples Improve the Performance of The Identif Network

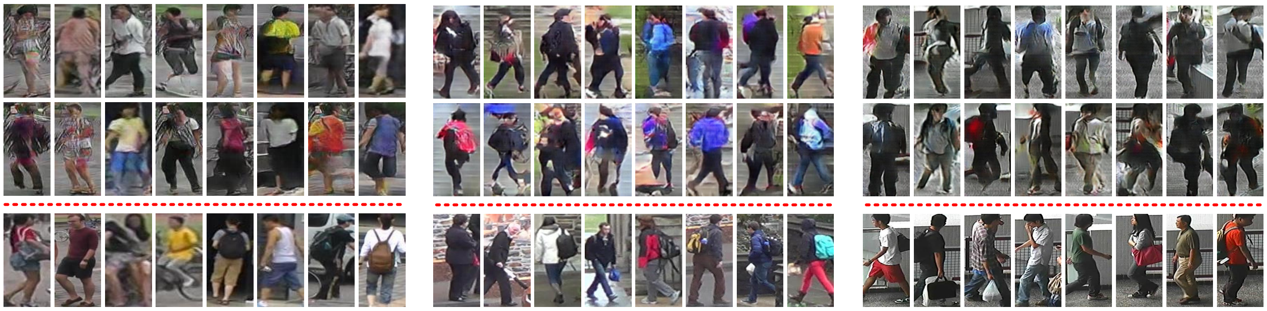
TABLE III
Comparison between LSRO and dMpRL-II on three datasets. Identif network is used by adding 24,000 generated samples. We show the promotion result with underline and **bold font** by using LSRO and MpRL, respectively.

Dataset	Method	mAP	rank-1
Market-1501	baseline	52.68%	74.08%
	LSRO [59]	56.33%	78.21%
	Promotion	+3.65%	+4.14%
	dMpRL-II	58.59%	80.37%
	Promotion	**+5.91%**	**+6.29%**
DukeMTMC-reID	baseline	42.20%	61.94%
	LSRO [59]	46.66%	66.92%
	Promotion	+4.46%	+4.98%
	dMpRL-II	48.58%	68.24%
	Promotion	**+6.38%**	**+6.30%**
CUHK03	baseline	68.36%	63.10%
	LSRO [59]	71.60%	66.30%
	Promotion	+3.24%	+3.20%
	dMpRL-II	73.48%	68.68%
	Promotion	**+5.12%**	**+5.58%**

We first give the result of the Identif network to evaluate our MpRL. Since the performance of the Two-stream network is higher, it will be used to compare with some state-of-the-art methods with the MpRL in Section V-H. Table III shows that when we add 24,000 GAN images in training the Identif network, our dMpRL-II significantly improves the re-ID performance on the strong baseline on Market-1501. The improvement is +5.91% (from 52.68% to 58.59%) and +6.29% (from 74.08% to 80.37%) in mAP and rank-1 accuracy, respectively. On the DukeMTMC-reID dataset, +6.38% (from 42.20% to 48.58%) and +6.3% (from 61.94% to 68.24%) improvements are obtained in mAP and rank-1 accuracy, respectively. We also use the CUHK03 dataset to evaluate our MpRL. The improvement is 5.12% (from 68.36% to 73.48%) and 5.58% (from 63.10% to 68.68%) in mAP and rank-1 accuracy, respectively. The above results indicate that by using the proposed MpRL on the generated samples, it can effectively yield improvements over the baseline performance.

E. Comparison with Different Implementations of MpRL

To demonstrate the effectiveness of the proposed MpRL, three implementations are used in our experiments (see Section III-B). We conduct this experiment using the Identif network. Table IV gives the comparison amongst the three implementations of MpRL on Market-1501. We observe by dynamically updating the contributions in training, dMpRL-I (+4.74% and +4.87% promotion in mAP and rank-1 accuracy, respectively) and dMpRL-II (+5.91% and +6.29% promotion in mAP and rank-1 accuracy, respectively) achieve better improvements compared with the sMpRL (+3.08% and +4.77% promotion in mAP and rank-1 accuracy, respectively). This is because each generated sample could receive a preferable MpRL along with the performance of the CNN getting better in training. Also, compared with dMpRL-I that updates the contribution from scratch, dMpRL-II achieves the best improvement by using MpRL when the network becomes relatively stable after 20 training epochs.



(a) Market-1501.

(b) DukeMTMC-reID.

(c) CUHK03.

Fig. 5. Examples of generated and real person images. (a), (b) and (c) show the generated person images (first two rows) and real person images (the third row) on Market-1501, DukeMTMC-reID and CUHK03, respectively.

TABLE IV

COMPARISON OF ALL-IN-ONE, ONE-HOT PSEUDO, LSRO AND MPRLS UNDER DIFFERENT NUMBERS OF GENERATED SAMPLES ON MARKET-1501 BY USING THE IDENTIF NETWORK. THE BEST PROMOTION OF DIFFERENT METHODS IS HIGHLIGHTED IN **BOLD**. RANK-1 ACCURACY AND MAP ARE SHOWN.

#GAN Img	All-in-one [37], [41]		One-hot Pseudo [26]		LSRO [59]		sMpRL		dMpRL-I		dMpRL-II	
	mAP	rank-1	mAP	rank-1	mAP	rank-1	mAP	rank-1	mAP	rank-1	mAP	rank-1
0 (base)	52.68%	74.08%	52.68%	74.08%	52.68%	74.08%	52.68%	74.08%	52.68%	74.08%	52.68%	74.08%
12000	55.68%	76.96%	55.69%	76.52%	55.22%	77.17%	55.27%	77.73%	55.84%	77.88%	58.14%	79.22%
18000	55.59%	**77.40%**	56.04%	77.95%	55.28%	76.96%	55.05%	77.73%	56.21%	78.36%	58.31%	79.81%
24000	56.07%	77.21%	**56.90%**	77.62%	**56.33%**	**78.21%**	55.59%	**78.85%**	56.10%	77.79%	**58.59%**	**80.37%**
30000	**56.19%**	77.17%	56.54%	**77.95%**	55.40%	77.46%	**55.76%**	77.82%	57.15%	78.65%	57.69%	79.16%
36000	55.24%	75.92%	56.38%	77.42%	55.82%	77.91%	55.45%	78.32%	**57.42%**	**78.95%**	57.61%	79.90%
promotion	3.51%	3.32%	4.22%	3.87%	3.65%	4.13%	3.08%	4.77%	4.74%	4.87%	**5.91%**	**6.29%**

F. Comparison with Existing Virtual Labels

To evaluate MpRL, we compare it with other three competitive virtual labels: all-in-one, one-hot pseudo and LSRO. Amongst them, LSRO [59] is the state-of-the-art method used on generated samples for person re-ID. Table IV provides the comparison results on Market-1501 using the Identif network. We add a different number of generated samples in training to show the promotion. We observe by adding 30,000 and 18,000 generated samples, the all-in-one achieves the best improvements in mAP (+3.51%) and rank-1 accuracy (+3.32%), respectively. The one-hot pseudo achieves +4.22% and +3.87% improvements in mAP and rank-1 accuracy when 24,000 and 30,000 generated samples are respectively added. Compared with the two methods, LSRO obtains a better rank-1 accuracy improvement (+4.13%) when adding 24,000 generated samples. However, the improvement of mAP (+3.65%) is slightly less than the one-hot pseudo. In this experiment, we use the same generated samples over all the methods, the promotion results are on par with that reported in [59]. Compared with the above methods, although the improvement of mAP (+3.08%) under the sMpRL is less than them, we obtain better rank-1 accuracy improvements under all the implementations of MpRL (+4.77%, +4.87%, and +6.29%). The mAP improvements of dMpRL-I and dMpRL-II are also better than others (+4.74% and +5.91%). By adding 24,000 generated samples, dMpRL-II improves the mAP and rank-1 accuracy of the Identif network from 52.68% and 74.08% to 58.59% and 80.37%, respectively. Compared with the previous state-of-the-art method LSRO, our promotion outperforms it to a certain degree (mAP: +3.65%→ +5.91%, rank-1 accuracy: +4.13%→ +6.29%).

In Table IV, it is clear to see that the multiple label distribution (LSRO and MpRL) can always outperform the one-hot label distribution (all-in-one and one-hot pseudo) in the rank-1 accuracy. The reason can be found in Section III-C1. Besides, we also find that compared with the way using the same label, assigning different labels to generated samples can produce better results in both of the multiple (MpRL vs. LSRO) and one-hot (one-hot pseudo vs. all-in-one) label distribution. The reason can be found in Section III-C2.

Furthermore, using the Identif network, Table III shows comparison results between dMpRL-II and LSRO on three datasets by adding 24,000 generated samples. This result indicates that by using different contributions from pre-defined classes, dMpRL-II can always outperform previous state-of-the-art virtual label LSRO in promoting the performance of the Identif network. The reason can be found in Section III-C3.

G. Evaluation on Different Batchsizes

In [24], Keskar *et al.* analysis the effect of different batchsizes in training CNNs. Using different batchsizes, the learned CNN will produce different generalization performance, which could affect the performance of testing. We try to evaluate MpRL through changing the batchsize in the two different CNNs. In this way, we want to observe the effectiveness of MpRL when the generalization performance of networks is changed.

TABLE V
RESULTS ON MARKET-1501 BY USING THE IDENTIF NETWORK WITH DIFFERENT BATCHSIZES (BS). RANK-1 ACCURACY AND MAP ARE LISTED.

BS	mAP	r1	BS	mAP	r1
64	52.68%	74.08%	28	57.71%	78.98%
60	53.08%	74.05%	24	58.64%	79.87%
56	53.74%	75.62%	20	58.73%	79.48%
52	54.06%	76.16%	*16*	*60.33%*	*81.89%*
48	54.35%	75.53%	12	60.16%	81.71%
44	55.11%	76.99%	8	60.43%	81.53%
40	54.95%	76.10%	4	55.29%	78.65%
36	56.68%	77.91%	16 (LSRO [59])	58.83%	80.23%
32	56.69%	77.76%	**16 (dMpRL-II)**	**63.34%**	**83.52%**

TABLE VI
RESULTS ON MARKET-1501 BY USING THE TWO-STREAM NETWORK WITH DIFFERENT BATCHSIZES (BS). RANK-1 ACCURACY AND MAP ARE LISTED.

	Only Real samples		LSRO [59]		dMpRL-II	
BS	mAP	rank-1	mAP	rank-1	mAP	rank-1
48	64.12%	82.13%	66.17%	83.82%	**67.70%**	**84.89%**
44	63.03%	80.55%	66.04%	83.19%	**67.98%**	**85.30%**
40	64.72%	82.75%	65.50%	83.88%	**67.91%**	**85.21%**
36	65.67%	83.31%	66.46%	83.94%	**68.63%**	**85.33%**
32	64.09%	81.83%	65.24%	83.82%	**67.53%**	**85.75%**
28	64.48%	83.52%	64.96%	*83.14%*	**66.15%**	**85.24%**
24	65.94%	83.58%	65.58%	83.55%	**66.05%**	**84.48%**
20	64.11%	82.74%	63.99%	83.31%	**65.52%**	**84.71%**
16	61.54%	81.47%	57.39%	79.27%	**67.43%**	**85.45%**
12	64.62%	82.42%	52.75%	75.36%	**64.63%**	**84.12%**
8	61.91%	81.98%	48.34%	72.09%	**64.96%**	**83.43%**
4	54.56%	76.22%	49.65%	72.63%	**58.27%**	**80.40%**

In this experiment, we modify the batchsize from 4 to 64 (stepsize is 4) in the Identif network, and 4 to 48 (stepsize is 4) in the Two-stream network to train Market-1501. The dMpRL-II is compared with the baseline (without generated samples) and LSRO. Table V shows the result by using the Identif network. Except $batchsize = 4$, it is clear to see an increasing tendency in performance when the batchsize becoming smaller. By just changing the batchsize in the Identif network, the gap between the best and the worst mAP can achieve 7.75% (from 52.68% $batchsize = 64$ to 60.43% $batchsize = 8$). Meanwhile, the gap between the best and worst rank-1 accuracy is 7.84% (from 74.05% $batchsize = 60$ to 81.89% $batchsize = 16$). The Identif network produces the best rank-1 performance when $batchsize = 16$. We add 24,000 generated samples and train the Identif network with $batchsize = 16$ by using LSRO and dMpRL-II. When the generalization performance of network increases, the performance of LSRO even worse than just using the real samples (mAP from 60.33% to 58.83% and rank-1 accuracy from 81.89% to 80.23%). However, dMpRL-II still obtains a further improvement under this strong baseline. The mAP and rank-1 accuracy are increased by +3.01% (from 60.33% to 63.34%) and +1.63% (from 81.89% to 83.52%), respectively. In this experiment, compared with previous state-of-the-art method LSRO, dMpRL-II demonstrates its strong regularization power by using different contributions from pre-defined classes. The reason can be found in Section III-C3.

Table VI shows the performance using different batchsizes on the Two-stream network. If only using the real samples to train the baseline, we obtain the best mAP (65.94%) and rank-1 accuracy (83.58%) when $batchsize = 24$. We also add 24,000 generated samples to train the Two-stream network by using LSRO and MpRL. When $batchsize = 44$ LSRO obtains the best improvement: +3.01% (from 63.03% to 66.04%) and +2.64% (from 80.55% to 83.19%) in mAP and rank-1 accuracy, respectively. However, when the $batchsize <= 28$ LSRO cannot improve the baseline with generated samples. Compared with LSRO, MpRL obtains better improvements: +4.95% (from 63.03% to 67.98%) and +4.75% (from 80.55% to 85.30%) in mAP and rank-1 accuracy, respectively when $batchsize = 44$. Beyond that, MpRL can always improve the performance of the Two-stream network under all the batchsize settings. This is mainly because MpRL assigns contribution based (different) virtual labels to all the generated samples. When the verification function added into the Two-stream network, MpRL could reduce ambiguous predictions in training. The reason can be found in Section III-C2.

H. Comparison with The State-of-the-art Methods

Although the main contribution in this paper focuses on using the generated samples to promote the performance of CNNs, but not on producing a state-of-the-art result, we still compare our result with recent published state-of-the-art methods. Table VII lists the comparison results. We can observe when $batchsize = 16$ on Market-1501, the Identif network is already competitive compared with many state-of-the-art methods except JLML [29], PDC [42] and Two-stream+LSRO [59] by training with MpRL (mAP: 63.34% and rank-1: 83.52%). Besides, it is clear to see that although the performance of the original Two-stream network is competitive, it still be inferior to many methods such as Resnet+OIM [49], SSM [4], JLML [29], SVDNet [43], PDC [42]. However, by incorporating with the proposed dMpRL-II, the Two-stream network obtains the state of the art compared with all the recently published methods on Market-1501, DukeMTMC-reID and CUHK03. To achieve better performance, after obtaining the rank list by sorting the similarity of gallery images to the query, a state-of-the-art re-ranking method [60] is adopted to further boost our performance. The combination of the dMpRL-II and reranking on the Two-stream network achieves the best results on the three datasets. In addition, we find that the rank-1 accuracy of the DPFL method [9] proposed in the ICCV17 workshop is slightly higher than our result on Market-1501 (88.90% in single query and 92.30% in multiple query). However, DPFL uses a ensemble deep model with multiple granularity inputs (from coarser to finer) for each image. Our Two-stream network just utilizes a single model and also outperforms the DPFL on CUHK03 by a large margin on mAP even without reranking (mAP: 87.53% (our) vs. 78.10% (DPFL), rank-1: 85.42% (our) vs. 82.00% (DPFL)).

TABLE VII

Comparison of our results with the published state-of-the-art methods. The best and the second best results are shown in **bold** and underline, respectively. Rank-1 accuracy and mAP are listed. BS means the batchsize.

Method		Market-1501				DukeMTMC-reID		CUHK03	
		Single Query		Multiple Query		Single Query		Single Query (detected)	
		mAP	rank-1	mAP	rank-1	mAP	rank-1	mAP	rank-1
Gate-reID	ECCV16 [45]	39.55%	65.88%	48.45%	76.04%	–	–	58.84%	68.10%
SI-CI	CVPR16 [47]	–	–	–	–	–	–	–	52.17%
GOG+XQDA	CVPR16 [36]	–	–	–	–	–	–	–	65.50%
SCSP	CVPR16 [6]	26.35%	51.90%	–	–	–	–	–	–
DNS	CVPR16 [51]	35.68%	61.02%	46.03%	71.56%	–	–	–	54.70%
Resnet+OIM	CVPR17 [49]	–	82.10%	–	–	–	68.10%	–	–
Latent Parts	CVPR17 [27]	57.53%	80.31%	66.70%	86.79%	–	–	–	67.99%
P2S	CVPR17 [61]	44.27%	70.72%	55.73%	85.78%	–	–	–	–
IDE(R)+KISSME+reranking	CVPR17 [60]	63.63%	77.11%	–	–	–	–	–	–
Consistent-Aware	CVPR17 [33]	55.60%	80.90%	–	–	–	–	–	–
Spindle	CVPR17 [52]	–	76.90%	–	–	–	–	–	–
SSM	CVPR17 [4]	68.80%	82.21%	76.18%	88.18%	–	–	–	72.70%
JLML	IJCAI17 [29]	65.50%	85.10%	74.50%	89.70%	–	–	–	80.60%
SVDNet	ICCV17 [43]	62.10%	82.30%	–	–	56.80%	76.70%	84.80%	81.80%
PDC	ICCV17 [42]	63.41%	84.14%	–	–	–	–	–	78.29%
Part Aligned	ICCV17 [53]	63.40%	81.00%	–	–	–	–	–	81.60%
Two-stream+LSRO	ICCV17 [59]	66.07%	83.97%	76.10%	88.42%	47.13%	67.68%	87.40%	84.60%
Identif [55], [56] (BS=64)		52.68%	74.08%	–	–	42.20%	61.94%	68.36%	63.10%
Identif (BS=64)+dMpRL-II (our)		58.59%	80.37%	–	–	48.58%	68.24%	73.48%	68.68%
Identif [55], [56] (BS=16)		60.33%	81.89%	–	–	–	–	–	–
Identif (BS=16)+dMpRL-II (our)		63.34%	83.52%	–	–	–	–	–	–
Two-stream [58] (BS=32)		64.09%	81.83%	73.65%	86.82%	51.40%	72.62%	85.20%	81.88%
Two-stream (BS=32)+dMpRL-II (our)		67.53%	85.75%	77.85%	89.88%	58.56%	76.81%	87.53%	85.42%
Two-stream (BS=32)+dMpRL-II (our)+reranking		**81.18%**	**87.96%**	**86.53%**	**90.97%**	**74.54%**	**81.28%**	**90.16%**	**88.00%**

VI. Conclusion

In this paper, we propose a new virtual label MpRL for generated samples by GAN. To train a CNN, MpRL is used as virtual labels assigned to generated samples, and these generated samples are incorporated for semi-supervised learning. Two CNNs are adopted to show the effectiveness of MpRL. Experiments demonstrate that generated samples can effectively improve the performance of the two CNNs trained with MpRL. Compared with the previous state-of-the-art method LSRO [59], MpRL always obtains better improvements. In the future, considering the capability of GAN, we will continue to investigate virtual labels used on generated samples for semi-supervised learning and apply the proposed method to other fields.

References

[1] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, et al. Tensorflow: A system for large-scale machine learning. In *OSDI*, volume 16, pages 265–283, 2016.

[2] S. Ali, O. Javed, N. Haering, and T. Kanade. Interactive retrieval of targets for wide area surveillance. In *in Proc. ACM Int. Conf. Multimedia. (ACMMM)*, pages 895–898, 2010.

[3] M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein gan. *arXiv preprint arXiv:1701.07875.*, 2017.

[4] S. Bai, X. Bai, and Q. Tian. Scalable person re-identification on supervised smoothed manifold. In *in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2017.

[5] M. Bauml, M. Tapaswi, and R. Stiefelhagen. Semi-supervised learning with constraints for person identification in multimedia data. In *in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 3602–3609, 2013.

[6] D. Chen, Z. Yuan, B. Chen, and N. Zheng. Similarity learning with spatial constraints for person re-identification. In *in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 1268–1277, 2016.

[7] W. Chen, X. Chen, J. Zhang, and K. Huang. Beyond triplet loss: a deep quadruplet network for person re-identification. In *in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2017.

[8] Y. Chen, W. Zheng, and J. Lai. Mirror representation for modeling view-specific transform in person re-identification. In *in Proc. Int. Joint. Conf. Artifi. Intelli. (IJCAI)*, 2015.

[9] Y. Chen, X. Zhu, and S. Gong. Person re-identification by deep learning multi-scale representations. In *in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, pages 2590–2600, 2017.

[10] 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 *in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 1335–1344, 2016.

[11] J. Dai, K. He, and J. Sun. Boxsup: Exploiting bounding boxes to supervise convolutional networks for semantic segmentation. In *in Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, pages 1635–1643, 2015.

[12] S. Ding, L. Lin, G. Wang, and H. Chao. Deep feature learning with relative distance comparison for person re-identification. *Pattern Recognition*, 48(10):2993–3003, 2015.

[13] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32(9):1627–1645, 2010.

[14] J. García, N. Martinel, A. Gardel, I. Bravo, G. L. Foresti, and C. Micheloni. Discriminant context information analysis for post-ranking person re-identification. *IEEE Trans. Image Process.*, 26(4):1650–1665, 2017.

[15] J. Garcia, N. Martinel, C. Micheloni, and A. Gardel. Person re-identification ranking optimisation by discriminant context information analysis. In *in Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, pages 1305–1313, 2015.

[16] M. Geng, Y. Wang, T. Xiang, and Y. Tian. Deep transfer learning for person re-identification. *arXiv preprint arXiv:1611.05244.*, 2016.

[17] S. Gong and T. Xiang. Person re-identification. In *Visual Analysis of Behaviour*, pages 301–313. 2011.

[18] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *in Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, pages 2672–2680, 2014.

[19] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 770–778, 2016.

[20] S. Hong, H. Noh, and B. Han. Decoupled deep neural network for semi-supervised semantic segmentation. In *in Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, pages 1495–1503, 2015.

[21] Y. Huang, H. Sheng, and Z. Xiong. Person re-identification based on hierarchical bipartite graph matching. In *in Proc. IEEE Int, Conf. Image. Process. (ICIP)*, pages 4255–4259, 2016.

[22] Y. Huang, H. Sheng, Y. Zheng, and Z. Xiong. Deepdiff: Learning deep difference features on human body parts for person re-identification. *Neurocomputing*, 241:191–203, 2017.

[23] R. Johnson and T. Zhang. Semi-supervised convolutional neural networks for text categorization via region embedding. In *in Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, pages 919–927, 2015.

[24] N. S. Keskar, D. Mudigere, J. Nocedal, M. Smelyanskiy, and P. T. P. Tang. On large-batch training for deep learning: Generalization gap and sharp minima. In *in Proc. Int. Conf. Learn. Represent. (ICLR)*, 2017.

[25] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980.*, 2014.

[26] D.-H. Lee. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on Challenges in Representation Learning, ICML*, volume 3, page 2, 2013.

[27] D. Li, X. Chen, Z. Zhang, and K. Huang. Learning deep context-aware features over body and latent parts for person re-identification. In *in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 384–393, 2017.

[28] W. Li, R. Zhao, T. Xiao, and X. Wang. Deepreid: Deep filter pairing neural network for person re-identification. In *in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 152–159, 2014.

[29] W. Li, X. Zhu, and S. Gong. Person re-identification by deep joint learning of multi-loss classification. In *in Proc. Int. Joint. Conf. Artifi. Intelli. (IJCAI)*, pages 2194–2200, 2017.

[30] Z. Li, S. Chang, F. Liang, T. S. Huang, L. Cao, and J. R. Smith. Learning locally-adaptive decision functions for person verification. In *in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 3610–3617, 2013.

[31] S. Liao, Y. Hu, X. Zhu, and S. Z. Li. Person re-identification by local maximal occurrence representation and metric learning. In *in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 2197–2206, 2015.

[32] S. Liao and S. Z. Li. Efficient psd constrained asymmetric metric learning for person re-identification. In *in Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, pages 3685–3693, 2015.

[33] J. Lin, L. Ren, J. Lu, J. Feng, and J. Zhou. Consistent-aware deep learning for person re-identification in a camera network. In *in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 5771–5780, 2017.

[34] C. Liu, C. Change Loy, S. Gong, and G. Wang. Pop: Person re-identification post-rank optimisation. In *in Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, pages 441–448, 2013.

[35] Y. Luo, D. Tao, B. Geng, C. Xu, and S. J. Maybank. Manifold regularized multitask learning for semi-supervised multilabel image classification. *IEEE Trans. Image Process.*, 22(2):523–536, 2013.

[36] T. Matsukawa, T. Okabe, E. Suzuki, and Y. Sato. Hierarchical gaussian descriptor for person re-identification. In *in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 1363–1372, 2016.

[37] A. Odena. Semi-supervised learning with generative adversarial networks. *arXiv preprint arXiv:1606.01583*, 2016.

[38] X. Qian, Y. Fu, Y.-G. Jiang, T. Xiang, and X. Xue. Multi-scale deep learning architectures for person re-identification. In *in Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, 2017.

[39] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In *in Proc. Int. Conf. Learn. Represent. (ICLR)*, 2016.

[40] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen. Improved techniques for training gans. In *in Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, pages 2234–2242, 2016.

[41] T. Salimans, I. J. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen. Improved techniques for training gans. In *in Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, pages 2226–2234, 2016.

[42] C. Su, J. Li, S. Zhang, J. Xing, W. Gao, and Q. Tian. Pose-driven deep convolutional model for person re-identification. In *in Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, 2017.

[43] Y. Sun, L. Zheng, W. Deng, and S. Wang. Svdnet for pedestrian retrieval. In *in Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, 2017.

[44] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. In *in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 2818–2826, 2016.

[45] R. R. Varior, M. Haloi, and G. Wang. Gated siamese convolutional neural network architecture for human re-identification. In *in Proc. Eur. Conf. Comput. Vis. (ECCV)*, pages 791–808, 2016.

[46] A. Vedaldi and K. Lenc. Matconvnet: Convolutional neural networks for matlab. In *in Proc. ACM Int. Conf. Multimedia. (ACMMM)*, pages 689–692, 2015.

[47] 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 *in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 1288–1296, 2016.

[48] H. Wang, S. Gong, X. Zhu, and T. Xiang. Human-in-the-loop person re-identification. In *in Proc. Eur. Conf. Comput. Vis. (ECCV)*, pages 405–422, 2016.

[49] T. Xiao, S. Li, B. Wang, L. Lin, and X. Wang. Joint detection and identification feature learning for person search. In *in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 3376–3385, 2017.

[50] H. Yu, A. Wu, and W. Zheng. Cross-view asymmetric metric learning for unsupervised person re-identification. In *in Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, 2017.

[51] L. Zhang, T. Xiang, and S. Gong. Learning a discriminative null space for person re-identification. In *in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 1239–1248, 2016.

[52] 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 *in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 1077–1085, 2017.

[53] L. Zhao, X. Li, J. Wang, and Y. Zhuang. Deeply-learned part-aligned representations for person re-identification. In *in Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, 2017.

[54] L. Zheng, L. Shen, L. Tian, S. Wang, J. Wang, and Q. Tian. Scalable person re-identification: A benchmark. In *in Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, pages 1116–1124, 2015.

[55] L. Zheng, Y. Yang, and A. G. Hauptmann. Person re-identification: Past, present and future. *arXiv preprint arXiv:1610.02984.*, 2016.

[56] L. Zheng, H. Zhang, S. Sun, M. Chandraker, and Q. Tian. Person re-identification in the wild. In *in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2016.

[57] W.-S. Zheng, S. Gong, and T. Xiang. Person re-identification by probabilistic relative distance comparison. In *in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 649–656, 2011.

[58] Z. Zheng, L. Zheng, and Y. Yang. A discriminatively learned cnn embedding for person re-identification. *ACM Transaction on Multimedia Computing Communications and Applications*, 2016.

[59] Z. Zheng, L. Zheng, and Y. Yang. Unlabeled samples generated by GAN improve the person re-identification baseline in vitro. In *in Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, 2017.

[60] Z. Zhong, L. Zheng, D. Cao, and S. Li. Re-ranking person re-identification with k-reciprocal encoding. In *in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2017.

[61] S. Zhou, J. Wang, J. Wang, Y. Gong, and N. Zheng. Point to set similarity based deep feature learning for person re-identification. In *in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, volume 6, 2017.