Generalizable Person Re-identification via Self-Supervised Batch Norm Test-Time Adaption

Ke Han[1,2], **Chenyang Si**[1], **Yan Huang**[1,3*], **Liang Wang**[1,3,4,5], **Tieniu Tan**[1,3,4]
[1] Center for Research on Intelligent Perception and Computing, Institute of Automation, Chinese Academy of Sciences
[2] School of Future Technology, University of Chinese Academy of Sciences (UCAS)
[3] School of Artificial Intelligence, University of Chinese Academy of Sciences (UCAS)
[4] Center for Excellence in Brain Science and Intelligence Technology (CEBSIT)
[5] Chinese Academy of Sciences, Artificial Intelligence Research (CAS-AIR)
ke.han@cripac.ia.ac.cn, chenyang.si.mail@gmail.com, {yhuang, wangliang, tnt}@nlpr.ia.ac.cn

Abstract

In this paper, we investigate the generalization problem of person re-identification (re-id), whose major challenge is the distribution shift on an unseen domain. As an important tool of regularizing the distribution, batch normalization (BN) has been widely used in existing methods. However, they neglect that BN is severely biased to the training domain and inevitably suffers the performance drop if directly generalized without being updated. To tackle this issue, we propose Batch Norm Test-time Adaption (BNTA), a novel re-id framework that applies the self-supervised strategy to update BN parameters adaptively. Specifically, BNTA quickly explores the domain-aware information within unlabeled target data before inference, and accordingly modulates the feature distribution normalized by BN to adapt to the target domain. This is accomplished by two designed self-supervised auxiliary tasks, namely part positioning and part nearest neighbor matching, which help the model mine the domain-aware information with respect to the structure and identity of body parts, respectively. To demonstrate the effectiveness of our method, we conduct extensive experiments on three re-id datasets and confirm the superior performance to the state-of-the-art methods.

Introduction

Person re-identification (re-id) aims to match individuals with the same identity across cameras at different locations over a large disjoint space. A number of efforts have been made by the re-id community over the past years, making remarkable progress in scenarios where training and test data come from the same domain. In reality, however, if tested directly on a previously unseen domain, most existing methods suffer significant performance degradation due to the distribution shift in background (Song et al. 2018), resolution (Han et al. 2021), clothing styles (Huang et al. 2021a), etc. Improving the generalization ability is therefore greatly important for promoting the study of re-id.

Recently, generalizable re-id methods have drawn increasing attention, which can be roughly divided into three categories. The methods of the first category, based on meta-learning, mimic the train-test splits to improve the ability of dealing with the stimulated generalization situations (Bai et al. 2021; Choi et al. 2021). The second methods aim to learn domain-invariant re-id features by the memory mechanism (Song et al. 2019), hard example mining (Tamura and Murakami 2019) or adversarial learning (Chen et al. 2021). The third methods advance the common usage of batch normalization (BN), *e.g.*, combining it with instance normalization (IN) to offset domain-related information captured by BN (Jia, Ruan, and Hospedales 2019; Jin et al. 2020).

The third methods reveal that BN discourages the generalization ability of the model, due to learning the knowledge biased to the source domain. We further explicitly investigate the correlation between distributions of BN outputs and domains in Fig. 1. (1). The red and green lines indicate the empirical Gaussian-like distributions output by the same (freezing) BN layer on the source and target domain, respectively. They exhibit a great distribution shift, *e.g.*, the variance increases obviously from the red line to the green one, suggesting the BN outputs on the target domain are dispersed more widely. The reason lies in that BN parameters

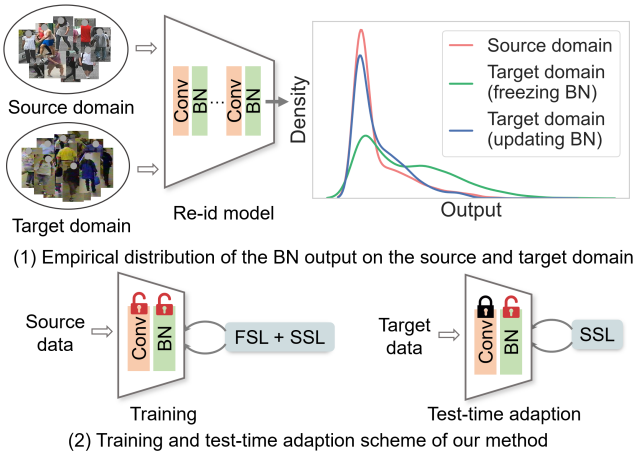

Figure 1: Illustration of our main idea. (1) The empirical distribution of the BN outputs shifts largely from the source to the target domain when BN layers are frozen, but the shift is mitigated when BN layers are updated. (2) Our model is jointly optimized via FSL and SSL heads during training. BN layers are updated via SSL, while other layers remain frozen during test-time adaption.

*Corresponding author.

are severely biased to the training data when regularizing the distribution, but the target data comes from a quite different distribution. This leads to two consequences. 1) The input distribution to the following layers (*e.g.*, convolutional layers) is accordingly deviated from that on the source domain, which affects adversely their accuracy of handling information. 2) The shift has even been accumulated for the top layers, thereby weakening the discriminativeness of the output features. However, this issue is not tackled well by existing methods, subject to their applying the trained BN layers to an unseen distribution directly without any updating.

Target samples, despite without identity labels, carry underlying prior information about the target distribution. It can be exploited to correct the training bias stored in BN layers for adapting to the target domain before inference, thus mitigating the distribution shift and enhancing the generalization performance. However, how to quickly explore the domain-aware information from unlabeled target data is highly challenging and rarely investigated in re-id. Self-supervised learning (SSL) has recently been proven very effective in unsupervised learning in the classification task, such as MoCo (He et al. 2020) and BYOL (Grill et al. 2020). Experimentally, they do not suit to be directly applied to re-id because of the task gap, making it urgent to design re-id oriented SSL tasks for unsupervised learning.

In this paper, we propose a novel Batch Norm Test-time Adaption (BNTA) re-id framework to update BN layers via self-supervision. Specifically, BNTA fast explores the domain-aware information within unlabeled target samples, and accordingly updates BN parameters (including statistics and affine parameters) to modulate the normalized feature distribution on the target domain. Inspired by the previous works verifying both body structure and identity information are important to re-id (Quan et al. 2019), we design two SSL auxiliary tasks for re-id named part positioning and part nearest neighbor matching. They help the model mine the target distribution involved with the structure and identity cues of body parts, by predicting the position and exploiting the similarity between the nearest neighbors for body parts, respectively. Furthermore, based on the two SSL tasks, we present a training and test-time adaption scheme. As illustrated in Fig 1. (2), our model is trained jointly via FSL (fully-supervised learning) and SSL on the labeled source data, while SSL further enables updating BN layers during test-time adaption to absorb the target distribution.

As a result, as shown in Fig 1. (1), the outputs of the updated BN layer on the target domain (blue line) have a similar distribution to that on the source domain (red line), which ensures that the following layers receive a stable input distribution and effectively enhances the generalization performance. Extensive experiments on VIPeR, GRID and iLIDS datasets demonstrate the effectiveness of our method, and confirm the advantage over the state-of-the-art methods. Besides, the proposed test-time adaption is fast and easy to implement, which only takes a few seconds with hundreds of gallery samples without additional target data collection. The generalization performance is even further boosted as the number of target samples increases, showing the potential of our model in the real-world scenarios where plenty of unlabeled target images are generally available.

We summarize the contributions of this paper as follows.

- To alleviate the distribution shift when transferring BN layers to an unseen domain, we propose a BNTA re-id framework for fast updating test-time BN parameters.
- Two simple yet effective SSL auxiliary tasks are designed to explore the structure and identity information of body parts from the unlabeled target data for BNTA.
- Extensive experiments demonstrate the state-of-the-art performance of our model on three re-id datasets, and also promote the understanding about how and why updating BN parameters improves the re-id generalization.

Related Work

Cross-Domain Person Re-Identification. Cross-domain re-id addresses the re-id performance drop across domains, assuming that data from the labeled source domain and unlabeled target domain are both utilizable during training. Mainstream methods include transferring the image style from the source to target domain (Deng et al. 2018; Wei et al. 2018), clustering and generating pseudo labels (Huang et al. 2018; Ji et al. 2020), and learning domain-aligned features (Lin et al. 2018; Huang et al. 2019, 2021b; Niu, Huang, and Wang 2019). However, they require large-scale target data collection (generally at least thousands of target images) and plenty of training iterations, which is highly inflexible and time-consuming when deploying a re-id system to a new domain. In contrast, our method does not need target data during training, and only takes a few seconds with hundreds of gallery images for the test-time adaption.

Generalizable Person Re-Identification. Unlike cross-domain re-id, generalizable re-id aims to improve the generalization performance on unseen domains, which supposes the target data is invisible during training. The main literature can roughly fall into three categories, *i.e.*, meta-learning, domain-invariant learning and BN based methods. The representative works are introduced as follows, respectively. **1)** Inspired by meta-learning, MetaBIN (Choi et al. 2021) stimulates and learns to handle the unsuccessful generalization situations, whereas DMG-Net (Bai et al. 2021) is a dual-meta network to exploit the meta-learning more fully in both the training procedure and metric space learning. **2)** To learn domain-invariant representations, DIMN (Song et al. 2019) maps an image to its identity classifier with a novel memory bank module; AugMing (Tamura and Murakami 2019) advances the strategy of data augmentation and selects the hard examples for increasing the utilization of data; DDAN (Chen et al. 2021) selectively aligns the distribution of multiple source domains via domain-wise adversarial learning and identity-wise similarity enhancement. **3)** Some methods, realizing the limitation of the common BN usage on the generalization ability, additionally combine IN to reduce the domain bias captured by BN with instance-related information (Jia, Ruan, and Hospedales 2019), while restituting the identity-relevant information (Jin et al. 2020).

All above methods can only be trained on the training data, and applied to a new domain without being updated. Different from them, our model has a domain-adaptive

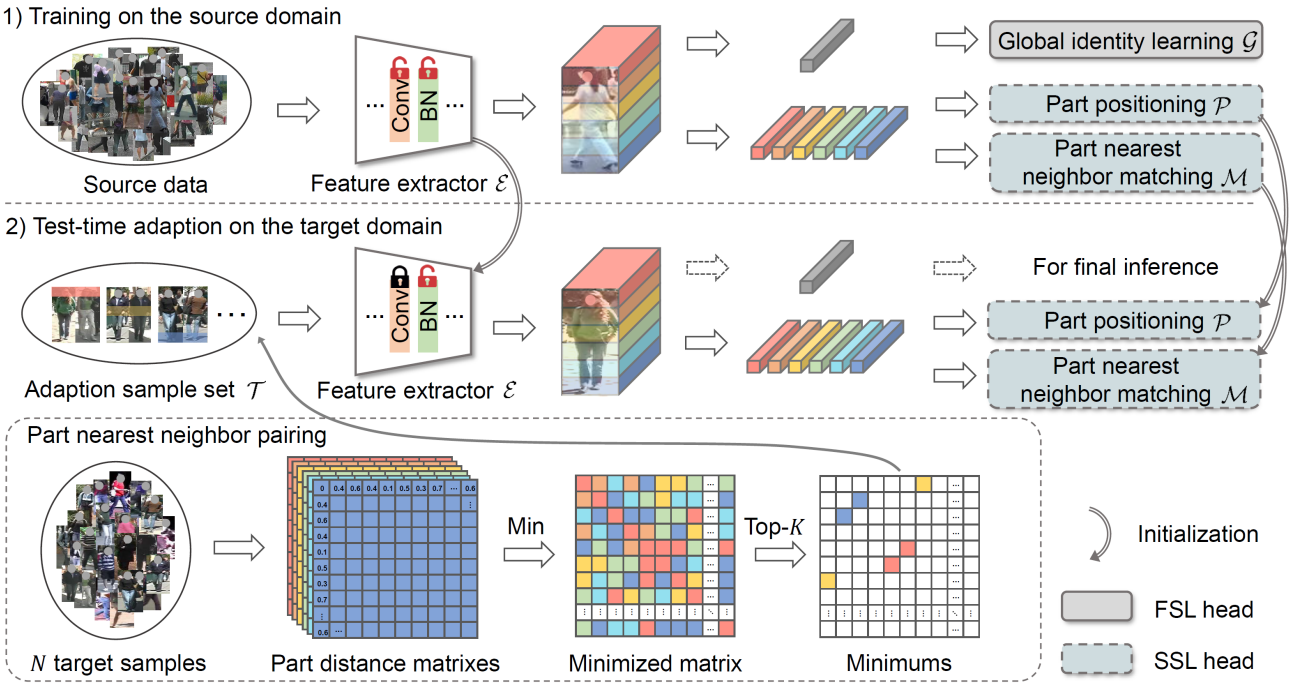

Figure 2: Illustration of the BNTA re-id framework. A FSL head $\mathcal{G}$ and two auxiliary SSL heads $\mathcal{P}, \mathcal{M}$ enable the training of the whole model on the labeled source domain based on global and local features, respectively. During test-time adaption, BN layers are updated via SSL using the adaption sample set, which includes K image pairs with part nearest neighbors selected from N target samples. The updated model extracts global features for the final inference.

learnability, which can perceive the target distribution to adaptively update the model for re-regularizing the feature distribution and enhancing the generalization performance.

Batch Normalization. Since originally designed to stabilize the neural responses and training procedure, BN (Ioffe and Szegedy 2015) has been used widely in deep neural networks. However, BN layers are inclined to normalize the training data to some specific Gaussian-like distributions, and if the target data deviates from that dramatically, the model will generalize poorly. To this end, CBN (Zhuang et al. 2020) optimizes BN layers with independent statistics on each training domain, and then recomputes the statistics on the target domain like AdaBN (Li et al. 2018). Also, some works (Seo et al. 2020; Chang et al. 2019) make a step forward and learn both domain-specific statistics and affine parameters, considering the regulatory effect of affine parameters is also closely tied to the domain. However, their learnable affine parameters are not updatable on the unlabeled target domain. Unlike them, our proposed SSL strategy can explore the domain-aware information for orienting BN parameters (both statistics and affine parameters) to the target distribution adaptively.

Method

Overview

In this paper, we propose a Batch Norm Test-time Adaption (BNTA) framework to improve the generalization ability for re-id, through updating test-time BN parameters via self-supervision. The overview is illustrated in Fig. 2.

We begin the detailed methodology with defining some notations. Following the generalizable re-id setting (Jia, Ruan, and Hospedales 2019), the training set $\mathcal{D}$ is a mixture of S source domains $\mathcal{D} = \{\mathcal{D}_1, \mathcal{D}_2, \cdots, \mathcal{D}_s\}$. Each domain is composed of labeled image pairs, and M_i is the number of training identities in $\mathcal{D}_i$. Since S source label sets are non-overlapping, there are totally $M = \sum_i M_i$ training identities in $\mathcal{D}$. The test set is collected from a target domain that is different from all the source domains.

FSL: Global Identity Learning

The FSL head $\mathcal{G}$ aims to learn discriminative identity features from the source labeled data for re-id. Many re-id models containing BN layers (Han et al. 2020; Niu et al. 2020; Niu, Huang, and Wang 2020) can be adopted as the backbone of our feature extractor $\mathcal{E}$. Here we choose DualNorm (Jia, Ruan, and Hospedales 2019) for its competitive generalization ability and relatively concise structure. For a given input image $\boldsymbol{x}_i \in \mathcal{D}$, we denote the extracted feature maps as $\mathcal{E}(\boldsymbol{x}_i) \in \mathbb{R}^{C\times H\times W}$, where C is the number of channels, H and W are the height and width, respectively. The following operation for identity learning is expressed as

$$\boldsymbol{f}_i = \text{Avgpool2d}(\mathcal{E}(\boldsymbol{x}_i)), \tag{1}$$

$$\mathcal{L}_{id} = -\boldsymbol{y}_i \cdot \log(\text{FC}_{\mathcal{G}}(\boldsymbol{f}_i)), \tag{2}$$

where $\boldsymbol{f}_i$ is the identity feature vector after global average pooling with the dimension C, $\text{FC}_{\mathcal{G}}(\cdot)$ is a fully-connected layer followed by a softmax function, and $\boldsymbol{y}_i$ is the binary identity label. The identity loss $\mathcal{L}_{id}$ is a cross-entropy loss

for making identity features discriminative by identity classification.

$$\mathcal{L}^{tri} = \sum_{p=1}^{P} \max(0, \phi_{tri} + d(f^*_{v_i}, f^{*}_{v_i^{p+}}) - d(f^*_{v_i}, f^{*}_{v_i^{p-}})), \tag{3}$$

where $d(\cdot,\cdot)$ is the Euclidean distance function. $\{f^*_{v_i}, f^{*}_{v_i^{p+}}, f^{*}_{v_i^{p-}}\}$ is a set of triplet training samples where $f^*_{v_i}$ is an anchor, $f^{*}_{v_i^{p+}}$ and $f^{*}_{v_i^{p-}}$ have the same and different identity label to $f^*_{v_i}$, respectively. ϕ_{tri} is the margin parameter, and P is the number of the triplet sets for v_i in a training batch. This loss could pull intra-class feature distances closer and push inter-class feature distances away.

SSL: Part Positioning

The body structure information plays an important role in re-id but is easily neglected (Quan et al. 2019). In addition, person images always have a clear structural prior, *i.e.*, on a person image from top to bottom is always head to feet, even captured on different domains. Inspired by the two findings, we present a SSL auxiliary head $\mathcal{P}$ named part positioning to explore the body structure within the images by predicting the positions of body parts. Specifically,

$$\boldsymbol{f}_i^h = \mathrm{Avgpool2d}_h(\mathcal{E}(\boldsymbol{x}_i)), \tag{4}$$

$$\mathcal{L}_{pos} = -\sum_{h=1}^{H} \boldsymbol{y}_i^h \cdot \log(\mathrm{FC}_{\mathcal{P}}(\boldsymbol{f}_i^h)). \tag{5}$$

where $\{\boldsymbol{f}_i^h\}_{h=1}^H$ are local feature vectors obtained by dividing the feature map evenly and average pooling, corresponding to H vertical body parts from top to bottom on $\boldsymbol{x}_i$. They are sent into a fully-connected layer $\mathcal{P}$ all together to predict the vertical position indexes (1, 2, ..., or H), along with the given binary labels $\boldsymbol{y}_i^h$,which is supervised by the positioning loss $\mathcal{L}_{pos}$. When applying our model to a new domain, $\mathcal{L}_{pos}$ can promote the model to reduce the feature distribution shift from the source domain by perceiving and aligning the structure information within images.

SSL: Part Nearest Neighbor Matching

Since re-id relies on identity characteristics for image retrieval, the inter-image identity similarity also has a significant impact on the feature distribution, apart from the body structure. We thereby design another SSL head $\mathcal{M}$, namely part nearest neighbor matching, to mine the identity distribution on the target domain based on the local similarity. The motivation of exploiting local instead of global similarity is that the local one has more reliability and potential for unlabeled target images. For example, when two images contain the seemingly same black shirts, even with different identities, we can still exploit them to simulate local positive pairs to explore the underlying inter-image identity similarity.

Training Version. During training, local features are required to be initialized to be discriminative, so that they can be used to modulate the identity distribution in place of global features during test-time adaption. The process for the local identity learning is formulated as

$$\boldsymbol{f}_i^h \leftarrow \mathrm{Conv}_{\mathcal{M}}^h(\boldsymbol{f}_i^h), \tag{6}$$

$$\mathcal{L}_{mat}^t = -\sum_{h=1}^{H} \boldsymbol{y}_i \cdot \log(\mathrm{FC}_{\mathcal{M}}^h(\boldsymbol{f}_i^h)), \tag{7}$$

where $\mathrm{Conv}_{\mathcal{M}}^h(\cdot)$ is the convolutional layer that has the kernel size of 1×1 and transforms the dimension of $\boldsymbol{f}_i^h$ from C to C_l. Similar to Eq. (2) formally, $\mathcal{L}_{mat}^t$ is the training version of the part nearest neighbor matching loss, and capacitates our model to extract discriminative local features.

Test-Time Adaption Version. To explore the inter-image identity similarity from unlabeled target data, we exploit the most similar parts among target samples as positive samples to allow modulating the identity distribution.

In generally, gallery images are readily accessible and plentiful when deploying a re-id model in a new scenario, so we do not need to collect extra images and simply sample N gallery images for the test-time adaption. The part nearest neighbor pairing scheme is proposed to compare the local similarity among N gallery images, and then select K pairs of image with the highest local similarity ($2K \leq N$, obviously). The detailed algorithm is shown in Alg. 1. It is worth noting that K image pairs are non-overlapping, which means that one image is paired with another *one* at most, thus making maximum use of more samples to stimulate the target distribution more accurately.

The adaption sample set is denoted as $\mathcal{T} = \{(\boldsymbol{t}_k^n, \boldsymbol{t}_k^{n+})\}_{k=1}^K, n \in \{h\}_{h=1}^H$, where $(\boldsymbol{t}_k^n, \boldsymbol{t}_k^{n+})$ is a pair of images whose n-th body parts are used for test-time adaption. The test-time adaption version of the part nearest neighbor matching loss $\mathcal{L}_{mat}^{tta}$ is defined as

$$\mathcal{L}_{mat}^{tta} = \sum_{k=1}^{K} \max(0, \phi + \mathrm{d}(\boldsymbol{f}_k^n, \boldsymbol{f}_k^{n+}) - \mathrm{d}(\boldsymbol{f}_k^n, \boldsymbol{f}_k^{n-})), \tag{8}$$

Algorithm 1: Part nearest neighbor pairing

Input: The trained feature extractor $\mathcal{E}$, the trained part nearest matching head $\mathcal{M}$, N gallery images, the hyper-parameter K.
Output: The adaption sample set $\mathcal{T}$.

1: **for** $i = 1$ **to** N **do**
2: Extract features of body parts $\{\boldsymbol{f}_i^h\}_{h=1}^H$ via Eq. (4) and Eq. (6). // $\boldsymbol{f}_i^h \in \mathbb{R}^{C_l}$
3: **end for**
4: Construct feature matrixes $\{\boldsymbol{M}_h\}_{h=1}^H$. // $\boldsymbol{M}_h \in \mathbb{R}^{N \times C_l}$
5: Calculate part distance matrixes $\{\boldsymbol{D}_h\}_{h=1}^H$ via Euclidean distance, and concatenate them as $\boldsymbol{D}$. // $\boldsymbol{D} \in \mathbb{R}^{H \times N \times N}$
6: Take the minimum among H part distances: $\boldsymbol{D} \leftarrow \min(\boldsymbol{D}, \dim = 0)$. // $\boldsymbol{D} \in \mathbb{R}^{N \times N}$
7: Construct $\mathcal{T}$, composed of K non-overlapping image pairs with the minimum part distances in $\boldsymbol{D}$ and the corresponding position labels.

where $\mathrm{d}(\boldsymbol{f}_k^n, \boldsymbol{f}_k^{n+})$ and $\mathrm{d}(\boldsymbol{f}_k^n, \boldsymbol{f}_k^{n-})$ are the Euclidean distances between the local feature $\boldsymbol{f}_k^h$ and its positive sample $\boldsymbol{f}_k^{n+}$ and the hardest negative sample $\boldsymbol{f}_k^{n-}$ in a mini-batch, respectively. ϕ is the margin parameter. Similar to the hardest triple loss (Hermans, Beyer, and Leibe 2017), this loss pulls the local features with the high similarity closer to each other while pushing those with the low similarity away. By resorting to the inter-image local similarity, $\mathcal{L}_{mat}^{tta}$ drives the fine-tuning of the identity-aware target distribution.

Training

At the training phase, **all** the parameters of our model are optimized by a FSL loss and two SSL losses end-to-end. The training loss $\mathcal{L}_t$ and optimization scheme are formulated as

$$\mathcal{L}_t = \mathcal{L}_{id} + \lambda_1 \cdot \mathcal{L}_{pos} + \lambda_2 \cdot \mathcal{L}_{mat}^t, \tag{9}$$

$$\theta_*^{all} \leftarrow \theta_*^{all} - \eta_t \nabla_{\theta_*^{all}} \mathcal{L}_t, \tag{10}$$

where $* \in \{\mathcal{E}, \mathcal{G}, \mathcal{P}, \mathcal{M}\}$, θ_*^{all} is all the parameters of $*$. η_t is the learning rate. λ_1 and λ_2 are weighting factors. The joint learning makes it possible to resort to adjusting the domain-aware structure and identity information for modulating the global feature distribution on the target domain, through associating the three distributions with each other.

Batch Norm Test-Time Adaption

BN is formulated as

$$\hat{\boldsymbol{x}_b} = \frac{\boldsymbol{x}_b - \mu}{\sqrt{\sigma^2 + \epsilon}}, \tag{11}$$

$$\boldsymbol{y}_b = \gamma \hat{\boldsymbol{x}_b} + \beta, \tag{12}$$

where $\boldsymbol{x}_b$ is the input on the dimension b to a BN layer. μ and σ^2 are the empirical mean and variance of the random variable $\boldsymbol{x}_b$, which are estimated with a batch of training samples. γ and η are learnable affine parameters used for linear transformation. Our experiments in Section suggest all of the four parameters are biased to the training data to different degrees, leading to the large feature distribution shift between domains. To this end, two SSL losses are used to update **BN** parameters for adapting them to the target domain during test-time adaption. We express the test-time adaption loss $\mathcal{L}_{tta}$ and optimization scheme as

$$\mathcal{L}_{tta} = \mathcal{L}_{pos} + \lambda_3 \cdot \mathcal{L}_{mat}^{tta}, \tag{13}$$

$$\theta_*^{bn} \leftarrow \theta_*^{bn} - \eta_{tta} \nabla_{\theta_*^{bn}} \mathcal{L}_{tta}, \tag{14}$$

where $* \in \{\mathcal{E}, \mathcal{P}, \mathcal{M}\}$, θ_*^{bn} indicate BN parameters including the statistics μ, σ^2 and affine parameters γ and β. η_{tta} is the learning rate and λ_3 is a weighting factor. Through fine-tuning parameters, BN layers re-regularize the feature distribution and pull it towards a more stable distribution that can be better processed by the following layers, thereby improving the generalization performance.

Experiments

Datasets and Settings

Datasets. Following (Jia, Ruan, and Hospedales 2019), we construct the training set by mixing five source domains: CUHK02 (Li and Wang 2013), CUHK03 (Li et al. 2014), Market-1501 (Zheng et al. 2015), DukeMTMC-ReID (Zheng, Zheng, and Yang 2017) and CUHK-SYSU Person-Search (Xiao et al. 2016). All images in the source domains are used for training regardless of train or test splits, covering 121, 765 images of 18, 530 identities in total. The test sets include VIPeR (Gray and Tao 2008), GRID (Loy, Xiang, and Gong 2009) and iLIDS (Zheng, Gong, and Xiang 2009). Some statistics are listed in Table 1.

Datasets	Test IDs		Test images	
	Probe	Gallery	Probe	Gallery
VIPeR	316	316	316	316
GRID	125	900	125	900
iLIDS	60	60	60	60

Table 1: Statistics of test sets.

Evaluation Protocol. We follow the common evaluation metrics for re-id, *i.e.*, mean average precision (mAP) and cumulative matching characteristic (CMC) at rank 1, 5 and 10. For all the test sets, the average results over 10 random splits are reported.

Implementation Details. Our model is pre-trained on ImageNet (Deng et al. 2009), and then trained on the training set for 60 epochs with the Adam optimizer (Kingma and Ba 2014) (β_1=0.9 and β_2=0.999). The learning rate η_t is initialized at 0.005, and decayed by 10 after 40 epochs. The batch size is set to 64. During test-time adaption, we perform the part nearest neighbor pairing among all the gallery images, which means N=316, 900 and 60 for VIPeR, GRID and iLIDS, respectively. The number of the selected image pairs K is set as K=$\min\{128, \frac{N}{2}\}$. We update our model for only 1 epoch, and randomly sample 32 pairs of images in each batch. Other hyper-parameters are set as follows: the number of stripes H=6, the weight factor λ_1=λ_2=0.1, λ_3=1, the learning rate η_{tta}=0.0005, the margin ϕ =0.3, the dimension C=2048, C_l=256. All the experiments are conducted on a single NVIDIA Titan Xp GPU with Pytorch.

Comparison with State-of-the-art Methods

We compare our model with the existing generalizable re-id methods, including DIMN (Song et al. 2019), AugMining (Tamura and Murakami 2019), DualNorm (Jia, Ruan, and Hospedales 2019), BoT (Luo et al. 2020), DDAN (Chen et al. 2021), DMG-Net (Bai et al. 2021) and MetaBIN (Choi et al. 2021). The comparison results are displayed in Table 2, showing that BNTA establishes the new state-of-the-art (SOTA) performance on VIPeR, GRID and iLIDS test sets. Whether the meta-learning based methods (DMG-Net and MetaBIN), or the methods for domain-invariant learning (DIMN, AugMining and DDAN), or the methods developing BN (DualNorm and BoT), are directly applied to a target domain without being updated. The superiority of our method over them lies in possessing a domain-adaptive learnability, which capacitates our model to update the model to fit in the target distribution automatically and reduce the distribution shift.

Method	VIPeR				GRID				iLIDS			
	mAP	R-1	R-5	R-10	mAP	R-1	R-5	R-10	mAP	R-1	R-5	R-10
DIMN (Song et al. 2019)	60.1	51.2	70.2	76.0	41.1	29.3	53.3	65.8	78.4	70.2	89.7	94.5
AugMining (Tamura and Murakami 2019)	-	49.8	70.8	77.0	-	46.6	67.5	76.1	-	76.3	93.0	95.3
DualNorm (Jia, Ruan, and Hospedales 2019)	58.0	53.9	62.5	75.3	45.7	41.4	47.4	64.7	78.5	74.8	82.0	91.5
BoT (Luo et al. 2020)	56.7	48.2	-	-	49.6	40.5	-	-	81.3	74.7	-	-
DDAN (Chen et al. 2021)	56.4	52.3	60.6	71.8	55.7	50.6	62.1	73.8	81.5	78.5	85.3	92.5
DMG-Net (Bai et al. 2021)	60.4	53.9	-	-	56.6	51.0	-	-	83.9	79.3	-	-
MetaBIN (Choi et al. 2021)	66.0	56.9	76.7	82.0	58.1	49.7	67.6	76.8	85.5	79.7	93.3	97.3
Baseline	60.4	50.9	72.3	80.0	48.3	38.5	57.6	65.0	82.0	76.1	89.5	94.3
BNTA w/o TTA	62.1	52.8	72.8	80.1	52.3	42.4	64.0	70.9	84.2	78.9	91.2	96.2
BNTA (Ours)	**67.3**	**57.4**	**77.6**	**82.2**	**58.7**	**51.1**	**68.5**	**77.3**	**85.8**	**80.6**	**93.6**	**97.7**

Table 2: Comparison with the state-of-the-art generalizable re-id methods (%). The best results are indicted in bold.

Training \ TTA	None	$\mathcal{L}_{pos}$	$\mathcal{L}_{mat}^{tta}$	$\mathcal{L}_{pos}, \mathcal{L}_{mat}^{tta}$
$\mathcal{L}_{id}$	50.9	-	-	-
$\mathcal{L}_{pos}$	21.6	21.3	-	-
$\mathcal{L}_{mat}^{t}$	48.2	-	51.8	-
$\mathcal{L}_{id}, \mathcal{L}_{pos}$	52.6	52.8	-	-
$\mathcal{L}_{id}, \mathcal{L}_{mat}^{t}$	52.3	-	55.6	-
$\mathcal{L}_{id}, \mathcal{L}_{pos}, \mathcal{L}_{mat}^{t}$	52.8	53.3	56.8	**57.4**

Table 3: Rank 1 of employing different losses during training and TTA on VIPeR (%). None means not performing TTA.

Ablation Study

Is BNTA effective? To demonstrate the effectiveness of BNTA, we compare it with the baseline and BNTA without test-time adaption (w/o TTA) on three datasets in Table 2. The **baseline** is trained only via FSL, while the **BNTA w/o TTA** is jointly trained via FSL and SSL, both of which are directly tested on the target domain without being updated.

BNTA w/o TTA achieves 1.6%, 3.9% and 2.7% higher rank 1 scores than the baseline on VIPeR, GRID and iLIDS, respectively. The performance gain of the additional SSL training results from facilitating the model to mine local image details for re-id, by predicting the position and learning the similarity for local features. This phenomenon is similar to the previous findings that exploiting local features boosts the re-id performance (Sun et al. 2018; Kalayeh et al. 2018).

Compared with BNTA w/o TTA, BNTA improves the rank 1 by 4.9%, 8.7% and 1.7% on three datasets, which confirms the effectiveness of exploiting body parts to correct the domain bias in BN. An explanation about the correlation between body parts and domains is that body parts contain local clothing information, which has different distributions on different domains, due to variation in clothing style, hue, brightness, etc. The domain shift can be reflected on the change of two self-supervision losses that are built on body parts. Then by self-supervised optimization of BNTA, our model can be adapted to the target distribution better.

Are both two SSL tasks effective? We validate the effectiveness of two SSL auxiliary tasks, *i.e.*, part positioning and part nearest neighbor matching, by the ablation study of $\mathcal{L}_{pos}$ and $\mathcal{L}_{mat}^{t}$ (or $\mathcal{L}_{mat}^{tta}$). As shown in Table 3, among multiple combinations of losses, employing L_{id}, $\mathcal{L}_{pos}$ and $\mathcal{L}_{mat}^{t}$ for training, $\mathcal{L}_{pos}$ and $\mathcal{L}_{mat}^{tta}$ for test-time adaption achieves the highest rank 1 score. Removing $\mathcal{L}_{pos}$ or $\mathcal{L}_{mat}^{t}$ (or $\mathcal{L}_{mat}^{tta}$) always decreases the performance in different extents. The combination of two SSL tasks improves the adaption of the model more obviously, since they help take in the target distribution with respect to the structure and identity information of body parts, respectively.

Updated Layers	Size	R-1	Updated Parameters	R-1
None	-	42.4	None	42.4
Conv	93M	47.8	Statistic μ	44.9
IN	3K	43.1	Statistics σ^2	43.5
Conv+IN	93M	47.8	Statistics μ, σ^2	45.2
BN+Conv	93M	48.1	Affine γ	47.5
BN+IN	93M	50.9	Affine β	46.8
BN+Conv+IN	93M	48.0	Affine γ, β	48.8
BN	60K	**51.1**	All $(\mu, \sigma^2, \gamma, \beta)$	**51.1**

Table 4: Left: rank 1 of updating different layers of the BNTA model on GRID and the corresponding parameter sizes. Right: rank 1 of updating different parameters of BN layers on GRID.

Updating BN or others layers? Our model includes three types of layers, *i.e.*, convolution, BN and IN, which are updatable on a new domain. Table 4 (left) shows the effect of updating different layers during TTA, and only updating BN contributes to the best accuracy. On the one hand, "BN" outperforms "Conv", "IN", "Conv+IN" by 3.3%, 8.0% and 3.3% in rank 1, respectively. This suggests that BN is biased to the training distribution much more seriously than convolution and IN due to the function of normalizing the feature distribution, and is thus more in need of updating on the target distribution. In fact, whether adding IN ("+IN") always has quite a small, and even negligible effect on the performance, partly because IN has a much smaller size of parameters than convolution and BN. Another reason is that IN only regularizes the features over an instance instead of the whole batch of samples like BN, thus not so heavily biased to the distribution of the whole dataset as BN. On the other hand, "BN" achieves the higher accuracy than "BN+Conv" and "BN+Conv+IN". This is because the parameter size of convolution (93M) is so larger than BN (60K) and IN (3K) that updating convolution dominates the performance

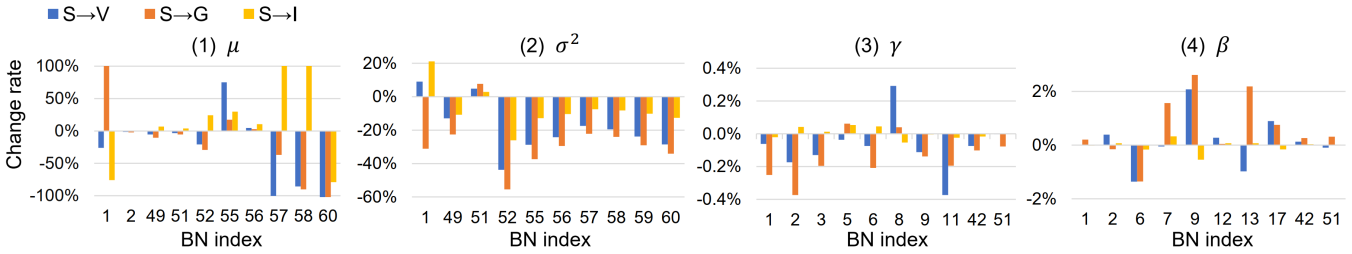

Figure 3: The change rates of BN parameters during test-time adaption. 'S→V', 'S→G' and 'S→I' indicate the transfer from the source dataset to VIPeR, GRID and iLIDS, respectively. For each parameter, the top-10 BN layers that have the largest average change rates in our model are shown here. BN is indexed according to the order from shallow to deep layers.

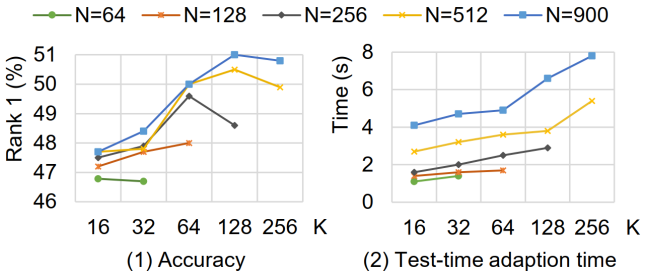

Figure 4: The effect of the hyper-parameter N and K on the accuracy and test-time adaption time on GRID.

change, and the effect of updating BN is largely weakened.

Updating which parameters of BN layers? As formulated in Eq. (11) and Eq. (12), updatable BN parameters include statistics μ, σ^2 and affine parameters γ, β, and Table 4 (right) lists the effects of updating each parameter during adaption. First, updating any one of the four parameters can boost the performance in different degrees, and the top rank 1 score (51.1%) is achieved when updating all of them together. Second, updating affine parameters even brings about more improvements than statistics. These results show that each of the four parameters is closely tied to the domain and suffers from the training bias, but our method can adapt all of them to the target domain and boost the performance fully.

How much do BN parameters change? We illustrate the change rate of each BN parameter during adaption in Fig. 3. There are totally 60 BN layers in our model, and we display the top-10 BN layers that have the largest average change rates for each parameter. The following observations are worth noting. First, there are significant differences between the change rates of S→V, G and I, even for the same BN layer and the same parameter. This reflects various distribution gaps between the source domain and different target domains, and our method can adjust BN parameters to the specific target domain adaptively. Second, the larger change rate usually takes places at some specific BN layers. For example, statistics μ and σ^2 tend to change more at the BN layers with the index 1, 52, 57, 58 and 60, whereas affine parameters μ and σ^2 have larger change rates at 2, 6, 9, 42 and 51, implying these BN layers are more sensitive to the domain shift than others.

How many samples are required for TTA? N and K are two hyper-parameters controlling the number of samples for TTA. The part nearest neighbor pairing is performed among N samples, and the top-K pairs with the highest local similarities are selected for updating the model ($2K \leq N$). Fig. 4 (1) depicts the effect of the two hyper-parameters, whic exhibits two notable tendencies. First, given a N, the rank 1 tends to first rise and then decline as K increases. More part nearest neighbors are not always useful and those with the highest similarities can simulate positive samples to facilitate modulating the identity-aware distribution better. Second, increasing N usually results in the better performance when K is fixed. A larger N provides more available target samples, so that the top-K pairs of part nearest neighbors can have quite high similarities to serve as positive samples more reasonably. The performance of our model is likely to be further improved as the number of available target samples grows, which shows the potential of our model in the real-world scenarios that usually allow easy access to a large number of unlabeled samples.

How much time does TTA cost? We show the time cost of the whole TTA process in Fig. 4 (2), corresponding to the settings of N and K in Fig. 4 (1). TTA only takes about 6.8s to update the model before inference to achieve the top rank 1 score (N=900, K=128). On average, inferencing an image on GRID takes 10.7ms and 4.1ms with and without adaption, respectively. It is worth noting that the adaption is only performed once, and not needed anymore only if the model is used for inference on the same target domain.

Conclusion

In this paper, we have proposed a BNTA framework for generalizable re-id, which updates test-time BN layers adaptively on the target domain to correct the training bias carried by BN. Two part-based SSL auxiliary tasks have been designed to explore the target distribution involved with the structure and identity information within images from unlabeled target samples. Extensive experiments have shown the effectiveness and potential of updating BN layers for improving the generalization ability. Only spending a few seconds with hundreds of gallery images for the test-time adaption before inference, our method achieves the state-of-the-art results on three re-id datasets. In the future work, we will investigate how to update BN and other layers jointly to further enhance the generalization ability.

Acknowledgements

This work was jointly supported by National Key Research and Development Program of China Grant No. 2018AAA0100400, National Natural Science Foundation of China (61633021, 61721004, 61806194, U1803261, and 61976132), Beijing Nova Program (Z201100006820079), Shandong Provincial Key Research and Development Program (2019JZZY010119), Key Research Program of Frontier Sciences CAS Grant No.ZDBS-LY-JSC032, and CAS-AIR.

References

Bai, Y.; Jiao, J.; Wang, C.; Liu, J.; Lou, Y.; Feng, X.; and Duan, L. 2021. Person30K: A Dual-Meta Generalization Network for Person Re-Identification. In *CVPR*.

Chang, W.-G.; You, T.; Seo, S.; Kwak, S.; and Han, B. 2019. Domain-specific batch normalization for unsupervised domain adaptation. In *CVPR*, 7354–7362.

Chen, P.; Dai, P.; Liu, J.; Zheng, F.; Xu, M.; Tian, Q.; and Ji, R. 2021. Dual Distribution Alignment Network for Generalizable Person Re-Identification. In *AAAI*.

Choi, S.; Kim, T.; Jeong, M.; Park, H.; and Kim, C. 2021. Meta Batch-Instance Normalization for Generalizable Person Re-Identification. In *CVPR*, 3425–3435.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In *CVPR*.

Deng, W.; Zheng, L.; Ye, Q.; Kang, G.; Yang, Y.; and Jiao, J. 2018. Image-image domain adaptation with preserved self-similarity and domain-dissimilarity for person re-identification. In *CVPR*, 994–1003.

Gray, D.; and Tao, H. 2008. Viewpoint invariant pedestrian recognition with an ensemble of localized features. In *ECCV*.

Grill, J.; Strub, F.; Altché, F.; Tallec, C.; Richemond, P. H.; Buchatskaya, E.; Doersch, C.; Pires, B. Á.; Guo, Z.; Azar, M. G.; Piot, B.; Kavukcuoglu, K.; Munos, R.; and Valko, M. 2020. Bootstrap Your Own Latent - A New Approach to Self-Supervised Learning. In *NeurIPS*.

Han, K.; Huang, Y.; Chen, Z.; Wang, L.; and Tan, T. 2020. Prediction and recovery for adaptive low-resolution person re-identification. In *ECCV*.

Han, K.; Huang, Y.; Song, C.; Wang, L.; and Tan, T. 2021. Adaptive super-resolution for person re-identification with low-resolution images. *PR*.

He, K.; Fan, H.; Wu, Y.; Xie, S.; and Girshick, R. B. 2020. Momentum Contrast for Unsupervised Visual Representation Learning. In *CVPR*.

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

Huang, Y.; Wu, Q.; Xu, J.; and Zhong, Y. 2019. Sbsgan: Suppression of inter-domain background shift for person re-identification. In *ICCV*.

Huang, Y.; Wu, Q.; Xu, J.; Zhong, Y.; and Zhang, Z. 2021a. Clothing Status Awareness for Long-Term Person Re-Identification. In *ICCV*.

Huang, Y.; Wu, Q.; Xu, J.; Zhong, Y.; and Zhang, Z. 2021b. Unsupervised Domain Adaptation with Background Shift Mitigating for Person Re-Identification. *IJCV*.

Huang, Y.; Xu, J.; Wu, Q.; Zheng, Z.; Zhang, Z.; and Zhang, J. 2018. Multi-pseudo regularized label for generated data in person re-identification. *TIP*.

Ioffe, S.; and Szegedy, C. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, 448–456.

Ji, Z.; Zou, X.; Lin, X.; Liu, X.; Huang, T.; and Wu, S. 2020. An attention-driven two-stage clustering method for unsupervised person re-identification. In *ECCV*, 20–36.

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

Jin, X.; Lan, C.; Zeng, W.; Chen, Z.; and Zhang, L. 2020. Style normalization and restitution for generalizable person re-identification. In *CVPR*, 3143–3152.

Kalayeh, M. M.; Basaran, E.; Gkmen, M.; Kamasak, M. E.; and Shah, M. 2018. Human semantic parsing for person re-identification. In *CVPR*.

Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Li, W.; and Wang, X. 2013. Locally aligned feature transforms across views. In *CVPR*, 3594–3601.

Li, W.; Zhao, R.; Xiao, T.; and Wang, X. 2014. Deepreid: Deep filter pairing neural network for person re-identification. In *CVPR*.

Li, Y.; Wang, N.; Shi, J.; Hou, X.; and Liu, J. 2018. Adaptive batch normalization for practical domain adaptation. *Pattern Recognition*, 80: 109–117.

Lin, S.; Li, H.; Li, C.-T.; and Kot, A. C. 2018. Multi-task mid-level feature alignment network for unsupervised cross-dataset person re-identification.

Loy, C. C.; Xiang, T.; and Gong, S. 2009. Multi-camera activity correlation analysis. In *CVPR*, 1988–1995.

Luo, H.; Jiang, W.; Gu, Y.; Liu, F.; Liao, X.; Lai, S.; and Gu, J. 2020. A Strong Baseline and Batch Normalization Neck for Deep Person Re-Identification. *IEEE Transactions on Multimedia*.

Niu, K.; Huang, Y.; Ouyang, W.; and Wang, L. 2020. Improving description-based person re-identification by multi-granularity image-text alignments. *TIP*.

Niu, K.; Huang, Y.; and Wang, L. 2019. Fusing two directions in cross-domain adaption for real life person search by language. In *ICCVW*.

Niu, K.; Huang, Y.; and Wang, L. 2020. Textual Dependency Embedding for Person Search by Language. In *ACM MM*.

Quan, R.; Dong, X.; Wu, Y.; Zhu, L.; and Yang, Y. 2019. Auto-ReID: Searching for a Part-aware ConvNet for Person Re-Identification. *ICCV*.

Seo, S.; Suh, Y.; Kim, D.; Kim, G.; Han, J.; and Han, B. 2020. Learning to optimize domain specific normalization for domain generalization. In *ECCV*, 68–83.

Song, C.; Huang, Y.; Ouyang, W.; and Wang, L. 2018. Mask-guided contrastive attention model for person re-identification. In *CVPR*.

Song, J.; Yang, Y.; Song, Y.-Z.; Xiang, T.; and Hospedales, T. M. 2019. Generalizable person re-identification by domain-invariant mapping network. In *CVPR*, 719–728.

Sun, Y.; Zheng, L.; Yang, Y.; Tian, Q.; and Wang, S. 2018. Beyond part models: Person retrieval with refined part pooling (and a strong convolutional baseline). In *ECCV*.

Tamura, M.; and Murakami, T. 2019. Augmented hard example mining for generalizable person re-identification. *arXiv preprint arXiv:1910.05280*.

Wei, L.; Zhang, S.; Gao, W.; and Tian, Q. 2018. Person transfer gan to bridge domain gap for person re-identification. In *CVPR*, 79–88.

Xiao, T.; Li, S.; Wang, B.; Lin, L.; and Wang, X. 2016. End-to-end deep learning for person search. *arXiv preprint arXiv:1604.01850*, 2(2): 4.

Zheng, L.; Shen, L.; Tian, L.; Wang, S.; Wang, J.; and Tian, Q. 2015. Scalable person re-identification: A benchmark. In *ICCV*.

Zheng, W.-S.; Gong, S.; and Xiang, T. 2009. Associating Groups of People. In *BMVC*, volume 2, 1–11.

Zheng, Z.; Zheng, L.; and Yang, Y. 2017. Unlabeled samples generated by gan improve the person re-identification baseline in vitro. In *ICCV*.

Zhuang, Z.; Wei, L.; Xie, L.; Zhang, T.; Zhang, H.; Wu, H.; Ai, H.; and Tian, Q. 2020. Rethinking the distribution gap of person re-identification with camera-based batch normalization. In *ECCV*, 140–157.