Joint Discriminative and Generative Learning for Person Re-identification

Zhedong Zheng\textsuperscript{1,2*}, Xiaodong Yang\textsuperscript{1}, Zhiding Yu\textsuperscript{1}, Liang Zheng\textsuperscript{3}, Yi Yang\textsuperscript{2}, Jan Kautz\textsuperscript{1}
\textsuperscript{1}NVIDIA \hspace{0.5cm} \textsuperscript{2}CAI, University of Technology Sydney \hspace{0.5cm} \textsuperscript{3}Australian National University

Abstract

Person re-identification (re-id) remains challenging due to significant intra-class variations across different cameras. Recently, there has been a growing interest in using generative models to augment training data and enhance the invariance to input changes. The generative pipelines in existing methods, however, stay relatively separate from the discriminative re-id learning stages. Accordingly, re-id models are often trained in a straightforward manner on the generated data. In this paper, we seek to improve learned re-id embeddings by better leveraging the generated data. To this end, we propose a joint learning framework that couples re-id learning and data generation end-to-end. Our model involves a generative module that separately encodes each person into an appearance code and a structure code, and a discriminative module that shares the appearance encoder with the generative module. By switching the appearance or structure codes, the generative module is able to generate high-quality cross-id composed images, which are online fed back to the appearance encoder and used to improve the discriminative module. The proposed joint learning framework renders significant improvement over the baseline without using generated data, leading to the state-of-the-art performance on several benchmark datasets.

1. Introduction

Person re-identification (re-id) aims to establish identity correspondences across different cameras. It is often approached as a metric learning problem [54], where one seeks to retrieve images containing the person of interest from non-overlapping cameras given a query image. This is challenging in the sense that images captured by different cameras often contain significant intra-class variations caused by the changes in background, viewpoint, human pose, etc. As a result, designing or learning representations that are robust against intra-class variations as much as possible has been one of the major targets in person re-id.

Convolutional neural networks (CNNs) have recently become increasingly predominant choices in person re-id thanks to their strong representation power and the ability to learn invariant deep embeddings. Current state-of-the-art re-id methods widely formulate the tasks as deep metric learning problems [13, 55], or use classification losses as the proxy targets to learn deep embeddings [23, 39, 41, 49, 54, 57]. To further reduce the influence from intra-class variations, a number of existing methods adopt part-based matching or ensemble to explicitly align and compensate the variations [35, 37, 47, 52, 57].

*Work done during an internship at NVIDIA Research.
is able to generate realistic and diverse intra/cross-id composed images by manipulating the codes. An example of such composed image generation on Market-1501 [53] is shown in Figure 1. Our design of the generative pipeline not only leads to high-fidelity generation, but also yields substantial diversity given the combinatorial compositions of existing identities. Unlike the unconditional GANs [17, 56], our method allows more controllable generation with better quality. Unlike the pose-guided generations [10, 27, 31], our method does not require any additional auxiliary data, but takes the advantage of existing intra-dataset pose variations as well as other diversities beyond pose.

This generative module design specifically serves for our discriminative module to better make use of the generated data. For one pedestrian image, by keeping its appearance code and combining with different structure codes, we can generate multiple images that remain clothing and shoes but change pose, viewpoint, background, etc. As demonstrated in each row of Figure 1, these images correspond to the same clothing dressed on different people. To better capture such composed cross-id information, we introduce the “primary feature learning” via a dynamic soft labeling strategy. Alternatively, we can keep one structure code and combine with different appearance codes to produce various images, which maintain the pose, background and some identity related fine details but alter clothes and shoes. As shown in each column of Figure 1, these images form an interesting simulation of the same person wearing different clothes and shoes. This creates an opportunity for further mining the subtle identity attributes that are independent of clothing, such as carrying, hair, body size, etc. Thus, we propose the complementary “fine-grained feature mining” to learn additional subtle identity properties.

To our knowledge, this work provides the first framework that is able to end-to-end integrate discriminative and generative learning in a single unified network for person re-id. Extensive qualitative and quantitative experiments show that our image generation compares favorably against the existing ones, and more importantly, our re-id accuracy consistently outperforms the competing algorithms by large margins on several benchmarks.

### 2. Related Work

A large family of person re-id research focuses on metric learning loss. Some methods combine identification loss with verification loss [48, 55], others apply triplet loss with hard sample mining [6, 13, 33]. Several recent works employ pedestrian attributes to enforce more supervisions and perform multi-task learning [26, 36, 44]. Alternatives harness pedestrian alignment and part matching to leverage on the human structure prior. One of the common practice is to split input images or feature maps horizontally to take advantage of local spatial cues [23, 39, 50].

<table>
<thead>
<tr>
<th>Appearance Space</th>
<th>Structure Space</th>
</tr>
</thead>
<tbody>
<tr>
<td>clothing/shoes color,</td>
<td>body size, hair, carrying,</td>
</tr>
<tr>
<td>texture and style, other id-related</td>
<td>pose, background, position,</td>
</tr>
<tr>
<td>cues, etc.</td>
<td>viewpoint, etc.</td>
</tr>
</tbody>
</table>

Table 1: Description of the information encoded in the latent appearance and structure spaces.

Another possibility to enhance robustness against input variations is to let the re-id model potentially “see” these variations (particularly intra-class variations) during training. With recent progress in the generative adversarial networks (GANs) [11], generative models have become appealing choices to introduce additional augmented data for free [56]. Despite the different forms, the general considerations behind these methods are “realism”: generated images should possess good qualities to close the domain gap between synthesized scenarios and real ones; and “diversity”: generated images should contain sufficient diversity to adequately cover unseen variations. Within this context, some prior works have explored unconditional GANs and human pose conditioned GANs [10, 17, 27, 31, 56] to generate pedestrian images to improve re-id learning. However, a common issue behind these methods is that their generative pipelines are typically presented as standalone models, which are relatively separate from the discriminative re-id models. Therefore, the optimization target of a generative module may not be well aligned with the re-id task, limiting the gain from generated data.

In light of the above observation, we propose a learning framework that jointly couples discriminative and generative learning in a unified network called DG-Net. Our strategy towards achieving this goal is to introduce a generative module, of which encoders decompose each pedestrian image into two latent spaces: an appearance space that mostly encodes appearance and other identity related semantics; and a structure space that encloses geometry and position related structural information as well as other additional variations. We refer to the encoded features in the space as “codes”. The properties captured by the two latent spaces are summarized in Table 1. The appearance space encoder is also shared with the discriminative module, serving as a re-id learning backbone. This design leads to a single unified framework that subsumes these interactions between generative and discriminative modules: (1) the generative module produces synthesized images that are taken to refine the appearance encoder online; (2) the encoder, in turn, influences the generative module with improved appearance encoding; and (3) both modules are jointly optimized, given the shared appearance encoder.

We formulate the image generation as switching the appearance or structure codes between two images. Given any pairwise images with the same/different identities, one
Figure 2: A schematic overview of DG-Net. (a) Our discriminative re-id learning module is embedded in the generative module by sharing appearance encoder \( E_a \). A dash black line denotes the input image to structure encoder \( E_s \) is converted to gray. The red line indicates the generated images are online fed back to \( E_a \). Two objectives are enforced in the generative module: (b) self-identity generation by the same input identity and (c) cross-identity generation by different input identities. (d) To better leverage generated data, the re-id learning involves primary feature learning and fine-grained feature mining.

Meanwhile, some recent studies also exploit synthetic data for style transfer of pedestrian images to compensate for the disparity between the source and target domains. CycleGAN [61] is applied in [9, 60] to transfer pedestrian image style from one dataset to another. StarGAN [7] is used in [59] to generate pedestrian images with different camera styles. Bak et al. [3] employ a game engine to render pedestrians using various illumination conditions. Wei et al. [46] take semantic segmentation to extract foreground mask in assisting style transfer. In contrast to the global style transfer, we aim for manipulating appearance and structure details to facilitate more robust re-id learning.

3. Method

As illustrated in Figure 2, DG-Net tightly couples the generative module for image generation and the discriminative module for re-id learning. We introduce two image mappings: self-identity generation and cross-identity generation to synthesize high-quality images that are online fed into re-id learning. Our discriminative module involves primary feature learning and fine-grained feature mining, which are co-designed with the generative module to better leverage the generated data.
3.1. Generative Module

Formulation. We denote the real images and identity labels as $X = \{x_i\}_{i=1}^N$ and $Y = \{y_i\}_{i=1}^N$, where $N$ is the number of images, $y_i \in [1, K]$ and $K$ indicates the number of classes or identities in the dataset. Given two real images $x_i$ and $x_j$ in the training set, our generative module generates a new pedestrian image by swapping the appearance or structure codes of the two images. As shown in Figure 2, the generative module consists of an appearance encoder $E_a : x_i \rightarrow a_i$, a structure encoder $E_s : x_j \rightarrow s_j$, a decoder $G : (a_i, s_j) \rightarrow x_j^i$, and a discriminator $D$ to distinguish between generated images and real ones. In the case $i = j$, the generator can be viewed as an auto-encoder, so $x_j^i \approx x_i$. Note: for generated images, we use subscript to denote the real image providing appearance code and subscript to indicate the one offering structure code, while real images only have subscript as image index. Compared to the appearance code $a_i$, the structure code $s_j$ maintains more spatial resolution to preserve geometric and positional properties. However, this may result in a trivial solution for $G$ to only use $s_j$ but ignore $a_i$ in image generation since decoders tend to rely on the feature with more spatial information. In practice, we convert input images of $E_s$ into gray-scale to drive $G$ to leverage both $a_i$ and $s_j$. We enforce the two objectives for the generative module: (1) self-identity generation to regularize the generator and (2) cross-identity generation to make generated images controllable and match real data distribution.

Self-identity generation. As illustrated in Figure 2(b), given an image $x_i$, the generative module first learns how to reconstruct $x_i$ from itself. This simple self-reconstruction task serves as an important regularization role to the whole generation. We reconstruct the image using the pixel-wise $\ell_1$ loss:

$$L_{\text{recon}}^\text{img1} = \mathbb{E}[||x_i - G(a_i, s_i)||_1].$$

(1)

Based on the assumption that the appearance codes of the same person in different images are close, we further propose another reconstruction task between any two images of the same identity. In other words, the generator should be able to reconstruct $x_i$ through an image $x_t$ with the same identity $y_i = y_t$:

$$L_{\text{recon}}^\text{img2} = \mathbb{E}[||x_i - G(a_t, s_i)||_1].$$

(2)

This same-identity but cross-image reconstruction loss encourages the appearance encoder to pull appearance codes of the same identity together so that intra-class feature variations are reduced. In the meantime, to force the appearance codes of different images to stay apart, we use identification loss to distinguish different identities:

$$L_{\text{id}}^a = \mathbb{E}[-\log(p(y_i|x_i))],$$

(3)

where $p(y_i|x_i)$ is the predicted probability that $x_i$ belongs to the ground-truth class $y_i$ based on its appearance code.

Cross-identity generation. Different from self-identity generation that works with image reconstruction using the same identity, cross-identity generation focuses on image generation with different identities. In this case, there is no pixel-level ground-truth supervision. Instead, we introduce the latent code reconstruction based on appearance and structure codes to control such image generation. As shown in Figure 2(c), given two images $x_i$ and $x_j$ of different identities $y_i \neq y_j$, the generated image $x_j^i = G(a_i, s_j)$ is required to retain the information of appearance code $a_i$ from $x_i$ and structure code $s_j$ from $x_j$, respectively. We should then be able to reconstruct the two latent codes after encoding the generated image:

$$L_{\text{recon}}^1 = \mathbb{E}[||a_i - E_a(G(a_i, s_j))||_1],$$

$$L_{\text{recon}}^2 = \mathbb{E}[||s_j - E_s(G(a_i, s_j))||_1].$$

(4)

(5)

Similar for self-identity generation, we also enforce identification loss on the generated image based on its appearance code to keep the identity consistency:

$$L_{\text{id}}^i = \mathbb{E}[-\log(p(y_i|x_j^i))],$$

(6)

where $p(y_i|x_j^i)$ is the predicted probability of $x_j^i$ belonging to the ground-truth class $y_i$ of $x_i$, the image that provides appearance code in generating $x_j^i$. Additionally, we employ adversarial loss to match the distribution of generated images to the real data distribution:

$$L_{\text{adv}} = \mathbb{E}[\log D(x_i) + \log(1 - D(G(a_i, s_j))].$$

(7)

Discussion. By using the proposed generation mechanism, we enable the generative module to learn appearance and structure codes with explicit and complementary meanings and generate high-quality pedestrian images based on the latent codes. This largely eases the generation complexity. In contrast, the previous methods [10, 17, 27, 31, 56] have to learn image generation either from random noise or managing the pose factor only, which is hard to manipulate the outputs and inevitably introduces artifacts. Moreover, due to using the latent codes, the variants in our generated images are explainable and constrained in the existing contents of real images, which also ensures the generation realism. In theory, we can generate $O(N \times N)$ different images by sampling various image pairs, resulting in a much larger online generated training sample pool than the ones with $O(2 \times N)$ images offline generated in [17, 31, 56].

3.2. Discriminative Module

Our discriminative module is embedded in the generative module by sharing the appearance encoder as the backbone for re-id learning. In accordance with the images generated by switching either appearance or structure codes, we propose the primary feature learning and fine-grained feature
mining to better take advantage of the online generated images. Since the two tasks focus on different aspects of generated images, we branch out two lightweight headers on top of the appearance encoder for the two types of feature learning, as illustrated in Figure 2(d).

**Primary feature learning.** It is possible to treat the generated images as training samples similar to the existing work [17,31,56]. But the inter-class variations in the cross-id composed images motivate us to adopt a teacher-student type supervision with dynamic soft labeling. We use a teacher model to dynamically assign a soft label to $x^n_j$, depending on its compound appearance and structure from $x_i$ and $x_j$. The teacher model is simply a baseline CNN trained with identification loss on the original training set. To train the discriminative module for primary feature learning, we minimize the KL divergence between the probability distribution $p(x^n_j)$ predicted by the discriminative module and the probability distribution $q(x^n_j)$ predicted by the teacher:

$$L_{\text{prim}} = \mathbb{E}[\sum_{k=1}^{K} q(k|x^n_j) \log \left( \frac{p(k|x^n_j)}{q(k|x^n_j)} \right)],$$  \hspace{1cm} (8)

where $K$ is the number of identities. In comparison with the fixed one-hot label [31,62] or static smoothing label [56], this dynamic soft labeling fits better in our case, as each synthetic image is formed by the visual contents from two real images. In the experiments, we show that a simple baseline CNN serving as the teacher model is reliable to provide the dynamic labels and improve the performance.

**Fine-grained feature mining.** Beyond the direct usage of generated data for learning primary features, an interesting alternative, made possible by our specific generation pipeline, is to simulate the change of clothing for the same person, as shown in each column of Figure 1. When training on images organized in this manner, the discriminative module is forced to learn the fine-grained id-related attributes (such as hair, hat, bag, body size, and so on) that are independent to clothing. We view the images generated by one structure code combining with different appearance codes as the same class as the real image providing the structure code. To train the discriminative module for fine-grained feature mining, we enforce identification loss on this particular categorizing:

$$L_{\text{fine}} = \mathbb{E}[-\log(p(y_j|x^n_j))].$$  \hspace{1cm} (9)

This loss imposes additional identity supervision to the discriminative module in a multi-tasking way. Moreover, unlike the previous works using manually labeled pedestrian attributes [26,36,44], our approach performs automatic fine-grained attribute mining by leveraging on the synthetic images. Furthermore, compared to the hard sampling policy applied in [13,33], there is no need to explicitly search for the hard training samples that usually possess fine-grained details, since our discriminative module learns to attention on the subtle identity properties through this fine-grained feature mining.

**Discussion.** We argue that our high-quality synthetic images, in nature, can be viewed as “inliers” (contrary to “outliers”), as our generated images maintain and re-compose the visual contents from real data. Via the above two feature learning tasks, our discriminative module makes specific use of the generated data in line with the way how we manipulate the appearance and structure codes. Instead of using a single supervision as in almost all previous methods [17,31,56], we treat the generated images in two different perspectives through the primary feature learning and fine-grained feature mining, where the former focuses on the structure-invariant clothing information and the latter attentions to the appearance-invariant structural cues.

### 3.3. Optimization.

We jointly train the appearance and structure encoders, decoder, and discriminator to optimize the total objective, which is a weighted sum of the following losses:

$$L_{\text{total}}(E_a, E_s, G, D) = \lambda_{\text{img}} L_{\text{img}} + \lambda_{\text{code}} L_{\text{code}} +$$

$$L_{\text{adv}} + \lambda_{\text{id}} L_{\text{id}} + \lambda_{\text{prim}} L_{\text{prim}} + \lambda_{\text{fine}} L_{\text{fine}},$$  \hspace{1cm} (10)

where $L_{\text{img}} = L_{\text{img1}} + L_{\text{img2}}$ is the image reconstruction loss in self-identity generation, $L_{\text{code}} = L_{\text{code1}} + L_{\text{code2}}$ is the latent code reconstruction loss in cross-identity generation, $L_{\text{adv}}$, $L_{\text{id}}$, $L_{\text{prim}}$, and $L_{\text{fine}}$ are weights to control the importance of related loss terms. Following the common practice in image-to-image translations [16,21,61], we use a large weight $\lambda_{\text{img}} = 5$ for the image reconstruction loss. Since the quality of cross-id generated images is not great at the beginning, the identification loss $L_{\text{id}}$ may make the training unstable, so we set a small weight $\lambda_{\text{id}} = 0.5$. We fix the two weights during the whole training process in all experiments. We do not involve the discriminative feature learning losses $L_{\text{prim}}$ and $L_{\text{fine}}$ until the generation quality is stable. As an example, we add in the two losses after 30K iterations on Market-1501, then linearly increase $\lambda_{\text{prim}}$ from 0 to 2 in 4K iterations and set $\lambda_{\text{fine}} = 0.2\lambda_{\text{prim}}$. See more details on how to determine the weights in Section 4.3. Similar to the alternative updating policy for GANs, in the cross-identity generation as shown in Figure 2(a), we alternatively train $E_a$, $E_s$ and $G$ before the generated image and $E_a$, $E_s$ and $D$ after the generated image.

### 4. Experiments

We evaluate the proposed approach following standard protocols on three benchmark datasets: Market-1501 [53], DukeMTMC-reID [32,56], and MSMT17 [46]. We qualitatively and quantitatively compare DG-Net with state-of-the-art methods on both generative and discriminative results.
Extensive experiments demonstrate that DG-Net produces more realistic and diverse images, and meanwhile, consistently outperforms the most recent competing algorithms by large margins on re-id accuracy across all benchmarks.

### 4.1. Implementation Details

Our network is implemented in PyTorch. In the following, we use channel×height×width to indicate the size of feature maps. (i) $E_a$ is based on ResNet50 [12] pre-trained on ImageNet [8], and we remove its global average pooling layer and fully-connected layer then append an adaptive max pooling layer to output the appearance code $a$ in $2048 \times 4 \times 1$. It is mapped to primary feature $f_{\text{prim}}$ and fine-grained feature $f_{\text{fine}}$, both are 512-dim vectors, through two fully-connected layers. (ii) $E_s$ is a shallow network that outputs the structure code $s$ in $128 \times 64 \times 32$. It consists of four convolutional layers followed by four residual blocks [12]. (iii) $G$ processes $s$ by four residual blocks and four convolutional layers. As in [16] every residual block contains two adaptive instance normalization layers [15], which integrate in $a$ as scale and bias parameters. (iv) $D$ follows the popular multi-scale PatchGAN [18]. We employ discriminators on the three different input image scales: $64 \times 32$, $128 \times 64$, and $256 \times 128$. We also apply the gradient punishment [30] when updating $D$ to stabilize training. (v) For training, all input images are resized to $256 \times 128$. Similar to the previous deep re-id models [54], SGD is used to train $E_a$ with learning rate 0.002 and momentum 0.9. We apply Adam [20] to optimize $E_s$, $G$, and $D$, and set learning rate to 0.0001, and $(\beta_1, \beta_2) = (0, 0.999)$. (vi) At test time, our re-id model only involves $E_a$ (along with two lightweight headers), which is of a comparable network size to most methods using ResNet50 as the backbone. We concatenate $f_{\text{prim}}$ and $f_{\text{fine}}$ into a 1024-dim vector as the final pedestrian representation. More architecture details can be found in the appendix.

### 4.2. Generative Evaluations

**Qualitative evaluations.** We first qualitatively compare DG-Net with its two variants that ablate online feeding and identity supervision. As shown in Figure 4, without online feeding generated images to appearance encoder, the model suffers from blurry edges and undesired textures. If further removing identity supervision, the image quality is unsatisfying as the model fails to produce the accurate clothing color or style. This clearly shows that our joint discriminative learning is beneficial to the image generation.

Next we compare our full model with other generative approaches, including one unconditional GAN (LSGAN [29]) and three open-source conditional GANs (PG$^2$-GAN [28], PN-GAN [31] and FD-GAN [10]). As compared in Figure 3, the images generated by LSGAN have severe artifacts and duplicated patterns. FD-GAN are prone to generate very blurry images, which largely deteriorate...
Figure 6: Examples of our generated images by swapping appearance or structure codes on the three datasets. All images are sampled from the test sets.

Table 2: Comparison of FID (lower is better) and SSIM (higher is better) to evaluate realism and diversity of the real and generated images on Market-1501.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Realism (FID)</th>
<th>Diversity (SSIM)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Real</td>
<td>7.22</td>
<td>0.350</td>
</tr>
<tr>
<td>LSGAN [29]</td>
<td>136.26</td>
<td>-</td>
</tr>
<tr>
<td>PG²-GAN [28]</td>
<td>151.16</td>
<td>-</td>
</tr>
<tr>
<td>PN-GAN [31]</td>
<td>54.23</td>
<td>0.335</td>
</tr>
<tr>
<td>FD-GAN [10]</td>
<td>257.00</td>
<td>0.247</td>
</tr>
<tr>
<td>Ours</td>
<td>18.24</td>
<td>0.360</td>
</tr>
</tbody>
</table>

Table 2: Comparison of FID (lower is better) and SSIM (higher is better) to evaluate realism and diversity of the real and generated images on Market-1501.

the realism. PG²-GAN and PN-GAN, both conditioned on pose, generate relatively good visual results, but still contain visible blurs and artifacts especially in background. In comparison, our generated images are more realistic and close to the real in both foreground and background.

To better understand the learned appearance space, which is the foundation for our pedestrian representations, we perform a linear interpolation between two appearance codes and generate the corresponding images as shown in Figure 5. These interpolation results verify the continuity in the appearance space, and show that our model is able to generalize in the space instead of simply memorizing trivial visual information. As a complementary study, we also generate images by linearly interpolating between two structure codes while keeping the appearance code intact. See more discussions regarding this study in the appendix. We then demonstrate our generation results on the three benchmarks in Figure 6, where DG-Net is found to be able to consistently generate realistic and diverse images across the different datasets.

Quantitative evaluations. Our qualitative observations above are confirmed by the quantitative evaluations. We use two metrics: Fréchet Inception Distance (FID) [14] and Structural SIMilarity (SSIM) [45] to measure realism and diversity of generated images, respectively. FID measures how close the distribution of generated images is to the real. It is sensitive to visual artifacts and thus indicates the realism of generated images. For the identity conditioned generation, we apply SSIM to compute intra-class similarity, which can be used to reflect the generation diversity. As shown in Table 2, our approach significantly outperforms other methods on both realism and diversity, suggesting the high quality of our generated images. Remarkably, we obtain a higher SSIM than the original training set thanks to the various poses, carryings, backgrounds, etc. introduced by switching structure codes.

Limitation. We notice that due to data bias in the original training set, our generative module tends to learn the regular textures (e.g., stripes and dots) but ignores some rare patterns (e.g., logos on shirts), as shown in Figure 7.

4.3. Discriminative Evaluations

Ablation studies. We first study the contributions of primary feature and fine-grained feature in Table 3. We train ResNet50 with identification loss on each original training set as the baseline. It also serves as the teacher model in primary feature learning to perform dynamic soft labeling on the generated images. Our primary feature is found to largely improve over the baseline. Notably, the fine-grained
Table 3: Comparison of baseline, primary feature, fine-grained feature, and their combination on the three datasets.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Market-1501 Rank@1 mAP</th>
<th>DukeMTMC-reID Rank@1 mAP</th>
<th>MSMT17 Rank@1 mAP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>89.6 74.5</td>
<td>82.0 65.3</td>
<td>68.8 36.2</td>
</tr>
<tr>
<td>$f_{\text{prim}}$</td>
<td>94.0 84.4</td>
<td>85.6 72.7</td>
<td>76.0 49.7</td>
</tr>
<tr>
<td>$f_{\text{fine}}$</td>
<td>91.6 75.3</td>
<td>78.7 61.2</td>
<td>71.5 43.5</td>
</tr>
<tr>
<td>$f_{\text{prim}}$, $f_{\text{fine}}$</td>
<td><strong>94.8</strong> <strong>86.0</strong></td>
<td><strong>86.6</strong> <strong>74.8</strong></td>
<td><strong>77.2</strong> <strong>52.3</strong></td>
</tr>
</tbody>
</table>

Table 4: Comparison with the state-of-the-art methods on the Market-1501 and DukeMTMC-reID datasets. Group 1: the methods not using generated data. Group 2: the methods using separately generated images.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Market-1501 Rank@1</th>
<th>Market-1501 Rank@5</th>
<th>Market-1501 Rank@10</th>
<th>DukeMTMC-reID Rank@1</th>
<th>DukeMTMC-reID Rank@5</th>
<th>DukeMTMC-reID Rank@10</th>
<th>mAP</th>
</tr>
</thead>
<tbody>
<tr>
<td>GLAD [47]</td>
<td>79.5 59.9</td>
<td>-</td>
<td>-</td>
<td>68.9 49.3</td>
<td>-</td>
<td>-</td>
<td>43.0</td>
</tr>
<tr>
<td>DCF [22]</td>
<td>80.3 57.5</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>56.8</td>
</tr>
<tr>
<td>SSM [2]</td>
<td>82.2 68.8</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>51.5</td>
</tr>
<tr>
<td>SVDNet [38]</td>
<td>82.3 62.1</td>
<td>-</td>
<td>-</td>
<td>76.7 56.8</td>
<td>80.5 63.8</td>
<td>81.6 58.0</td>
<td>52.8</td>
</tr>
<tr>
<td>PAN [57]</td>
<td>82.8 63.4</td>
<td>-</td>
<td>-</td>
<td>71.6 51.5</td>
<td>-</td>
<td>-</td>
<td>62.8</td>
</tr>
<tr>
<td>GLAD [47]</td>
<td>89.9 73.9</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>69.3</td>
</tr>
<tr>
<td>HA-CNN [24]</td>
<td>91.2 75.7</td>
<td>-</td>
<td>-</td>
<td>80.5 63.8</td>
<td>81.6 58.0</td>
<td>81.6 58.0</td>
<td>69.2</td>
</tr>
<tr>
<td>MLFN [4]</td>
<td>90.0 74.3</td>
<td>-</td>
<td>-</td>
<td>81.0 62.8</td>
<td>-</td>
<td>-</td>
<td>71.8</td>
</tr>
<tr>
<td>Part-aligned [37]</td>
<td>91.7 79.6</td>
<td>-</td>
<td>-</td>
<td>84.4 69.3</td>
<td>-</td>
<td>-</td>
<td>69.2</td>
</tr>
<tr>
<td>PCB [39]</td>
<td>93.8 81.6</td>
<td>-</td>
<td>-</td>
<td>83.3 69.2</td>
<td>-</td>
<td>-</td>
<td>71.8</td>
</tr>
<tr>
<td>Mancs [43]</td>
<td>93.1 82.3</td>
<td>-</td>
<td>-</td>
<td>84.9 71.8</td>
<td>-</td>
<td>-</td>
<td>64.5</td>
</tr>
<tr>
<td>DeformGAN [34]</td>
<td>80.6 61.3</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>47.1</td>
</tr>
<tr>
<td>LSRO [56]</td>
<td>84.0 66.1</td>
<td>-</td>
<td>-</td>
<td>67.7 47.1</td>
<td>-</td>
<td>-</td>
<td>58.6</td>
</tr>
<tr>
<td>Multi-pseudo [17]</td>
<td>85.8 67.5</td>
<td>-</td>
<td>-</td>
<td>76.8 58.6</td>
<td>81.0 62.8</td>
<td>81.6 58.0</td>
<td>56.9</td>
</tr>
<tr>
<td>PT [27]</td>
<td>87.7 68.9</td>
<td>-</td>
<td>-</td>
<td>78.5 56.9</td>
<td>81.0 62.8</td>
<td>81.6 58.0</td>
<td>56.9</td>
</tr>
<tr>
<td>PN-GAN [31]</td>
<td>89.4 72.6</td>
<td>-</td>
<td>-</td>
<td>73.6 53.2</td>
<td>-</td>
<td>-</td>
<td>62.8</td>
</tr>
<tr>
<td>FD-GAN [10]</td>
<td>90.5 77.7</td>
<td>-</td>
<td>-</td>
<td>80.0 64.5</td>
<td>-</td>
<td>-</td>
<td>69.2</td>
</tr>
<tr>
<td>Ours</td>
<td><strong>94.8</strong> <strong>86.0</strong></td>
<td><strong>86.6</strong> <strong>74.8</strong></td>
<td><strong>77.2</strong> <strong>52.3</strong></td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 5: Comparison with the state-of-the-art methods on the MSMT17 dataset.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Market-1501 Rank@1</th>
<th>DukeMTMC-reID Rank@1</th>
<th>mAP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Verif-Identif [55]</td>
<td>79.5 59.9</td>
<td>68.9 49.3</td>
<td>43.0</td>
</tr>
<tr>
<td>DCF [22]</td>
<td>80.3 57.5</td>
<td>-</td>
<td>56.8</td>
</tr>
<tr>
<td>SSM [2]</td>
<td>82.2 68.8</td>
<td>-</td>
<td>51.5</td>
</tr>
<tr>
<td>SVDNet [38]</td>
<td>82.3 62.1</td>
<td>76.7 56.8</td>
<td>52.8</td>
</tr>
<tr>
<td>PAN [57]</td>
<td>82.8 63.4</td>
<td>71.6 51.5</td>
<td>58.6</td>
</tr>
<tr>
<td>GLAD [47]</td>
<td>89.9 73.9</td>
<td>-</td>
<td>69.3</td>
</tr>
<tr>
<td>HA-CNN [24]</td>
<td>91.2 75.7</td>
<td>80.5 63.8</td>
<td>69.2</td>
</tr>
<tr>
<td>MLFN [4]</td>
<td>90.0 74.3</td>
<td>81.0 62.8</td>
<td>69.2</td>
</tr>
<tr>
<td>Part-aligned [37]</td>
<td>91.7 79.6</td>
<td>84.4 69.3</td>
<td>69.2</td>
</tr>
<tr>
<td>PCB [39]</td>
<td>93.8 81.6</td>
<td>83.3 69.2</td>
<td>71.8</td>
</tr>
<tr>
<td>Mancs [43]</td>
<td>93.1 82.3</td>
<td>84.9 71.8</td>
<td>64.5</td>
</tr>
<tr>
<td>DeformGAN [34]</td>
<td>80.6 61.3</td>
<td>-</td>
<td>47.1</td>
</tr>
<tr>
<td>LSRO [56]</td>
<td>84.0 66.1</td>
<td>67.7 47.1</td>
<td>58.6</td>
</tr>
<tr>
<td>Multi-pseudo [17]</td>
<td>85.8 67.5</td>
<td>76.8 58.6</td>
<td>56.9</td>
</tr>
<tr>
<td>PT [27]</td>
<td>87.7 68.9</td>
<td>78.5 56.9</td>
<td>56.9</td>
</tr>
<tr>
<td>PN-GAN [31]</td>
<td>89.4 72.6</td>
<td>73.6 53.2</td>
<td>53.2</td>
</tr>
<tr>
<td>FD-GAN [10]</td>
<td>90.5 77.7</td>
<td>80.0 64.5</td>
<td>64.5</td>
</tr>
<tr>
<td>Ours</td>
<td><strong>94.8</strong> <strong>86.0</strong></td>
<td><strong>86.6</strong> <strong>74.8</strong></td>
<td><strong>52.3</strong></td>
</tr>
</tbody>
</table>

Influence of hyper-parameters. Here we show how to set the re-id learning related weights: one is $\alpha$, the ratio between $\lambda_{\text{fine}}$ and $\lambda_{\text{prim}}$ to control the importance of $L_{\text{fine}}$ and $L_{\text{prim}}$ in training; the other is $\beta$ to weight $f_{\text{fine}}$ when combined with $f_{\text{prim}}$ as the final pedestrian representation in testing. We search the two hyper-parameters on a validation set split out from the original training set of Market-1501 (first 651 classes for training and rest 100 classes for validation). Based on the validation results in Figure 8, we choose $\alpha = 0.2$ and $\beta = 0.5$ in all experiments.

Comparison with state-of-the-art methods. Finally we report the performance of our approach with other state-of-the-art results in Tables 4 and 5. Note that we do not apply any post processing such as re-ranking [51] or multi-query fusion [55]. On each dataset, our approach attains the best performance. Comparing with the methods using separately generated images, DG-Net achieves clear gains of 8.3% and 10.3% for mAP on Market-1501 and DukeMTMC-reID, indicating the advantage of the proposed joint learning. Moreover, our framework is more training efficient: we use only one training phase for joint image generation and re-id learning, while others require two training phases to sequentially train generative models and re-id models. DG-Net also outperforms other non-generative methods by large margins on the two datasets. As for the recent released large-scale dataset MSMT17, DG-Net performs significantly better than the second best method by 9.0% for Rank@1 and 11.9% for mAP.

5. Conclusion

In this paper, we have proposed a joint learning framework that end-to-end couples re-id learning and image generation in a unified network. There exists an online interactive loop between the discriminative and generative modules to mutually benefit the two tasks. Our two modules are co-designed to let the re-id learning better leverage the generated data, rather than simply training on them. Experiments on three benchmarks demonstrate that our approach consistently brings substantial improvements to both image generation quality and re-id accuracy.
References

[16] Xun Huang, Ming-Yu Liu, Serge Belongie, and Jan Kautz. Multimodal unsupervised image-to-image translation. In ECCV, 2018. 5, 6, 11
[22] Danny Li, Xiaotang Chen, Zhang Zhang, and Kaili Huang. Learning deep context-aware features over body and latent parts for person re-identification. In CVPR, 2017. 8
[29] Xudong Mao, Qing Li, Haoran Xie, Raymond Lau, Zhen Wang, and Stephen Smolley. Least squares generative adversarial networks. In ICCV, 2017. 6, 7
[31] Xuelin Qian, Yanwei Fu, Tao Xiang, Wenxuan Wang, Jie Qiu, Yang Wu, Yu-Gang Jiang, and Xiangyang Xue. Pose-normalized image generation for person re-identification. In ECCV, 2018. 2, 3, 4, 5, 6, 7, 8


[50] Dong Yi, Zhen Lei, Shengcai Liao, and Stan Li. Deep metric learning for person re-identification. In *ICPR*, 2014. 2


Appendix

In this appendix, Section A summarizes the architecture details of DG-Net. Section B presents more re-id evaluations. Section C provides more rationales behind the appearance and structure spaces as well as the primary and fine-grained feature learning on appearance code. Section D demonstrates the example of image generation by interpolating between structure codes.

A. Network Architectures

Our proposed DG-Net consists of the appearance encoder $E_a$, structure encoder $E_s$, decoder $G$, and discriminator $D$. As described in the paper that $E_a$ is modified from ResNet50, we now introduce the architecture details of $E_s$, $G$, and $D$. Following the common practice in GANs, we mainly adopt convolutional layers and residual blocks [12] to construct them.

Table 6 shows the architecture of $E_s$. After each convolutional layer, we apply the instance normalization layer [42] and LReLU (negative slope set to 0.2). We also add the optional atrous spatial pyramid pooling (ASPP) [5], which contains dilated convolutions and can be used to exploit multi-scale features. Table 7 demonstrates the architecture of decoder $G$, which involves several residual blocks followed by upsampling and convolutional layers. Similar to [16], we insert the adaptive instance normalization (AdaIN) layer in every residual block to integrate the appearance code from $E_a$ as the dynamically generated weight and bias parameters of AdaIN. We employ the multi-scale PatchGAN [61] as the discriminator $D$. Given an input image of $256 \times 256$, we resize the image to the three different scales: $256 \times 128$, $256 \times 64$, $128 \times 64$ before feeding them into the discriminator. LReLU (negative slope set to 0.2) is applied after each convolutional layer. We present the architecture of $D$ in Table 8.

B. More Discriminative Evaluations

In order to have a more thorough evaluation of our approach, we further evaluate the performance of DG-Net on a relatively small dataset. So we generalize our approach to CUHK03-NP [58], which contains much fewer images (9.6 training images per person on average) compared to Market-1501 [53], DukeMTMC-reID [32] and MSMT17 [46]. As compared in Table 9, DG-Net achieves 65.6% Rank@1 and 61.1% mAP.

C. Appearance and Structure Codes

Since we cannot quantitatively justify the attributes of appearance/structure codes, Table 1 in the paper is used to qualitatively give an intuition. Our design of $E_s$ (a shallow network) makes the structure space primarily preserve the structural information, such as position and geometry of humans and objects. Thus, the structure code is mainly used to hold the low-level positional and geometric information, such as pose and background that are non-id-related, to facilitate image synthesis. On the other hand, certain structure cues, such as bag/hair/body outline, are clearly id-related.
Table 9: Comparison with the state-of-the-art results on the CUHK03-NP dataset.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Rank@1</th>
<th>mAP</th>
</tr>
</thead>
<tbody>
<tr>
<td>HA-CNN [24]</td>
<td>41.7%</td>
<td>38.6%</td>
</tr>
<tr>
<td>PT [27]</td>
<td>41.6%</td>
<td>38.7%</td>
</tr>
<tr>
<td>MLFN [4]</td>
<td>52.8%</td>
<td>47.8%</td>
</tr>
<tr>
<td>PCB [39]</td>
<td>61.3%</td>
<td>54.2%</td>
</tr>
<tr>
<td>PCB + RPP [39]</td>
<td>63.7%</td>
<td>57.5%</td>
</tr>
<tr>
<td>Ours</td>
<td>65.6%</td>
<td>61.1%</td>
</tr>
</tbody>
</table>

and are better to be captured by the discriminative module. However, softmax loss is generally too “lazy” to be able to capture useful structure information besides appearance features, therefore, the goal of fine-grained feature mining upon the appearance code promotes mining the id-related semantics out of structure cues, also guarantees the complementary nature between primary and fine-grained features.

D. Interpolate between Structure Codes

Figure 5 in the paper shows the examples of synthesized images by linear interpolation between two appearance codes. This qualitatively validates the continuity in the appearance space. As a complementary study, here we generate the images by linearly interpolating between two structure codes while keeping the appearance codes intact in Figure 9. This demonstrates the exact opposite setting to Figure 5. As expected, most images (both foreground and background) look not realistic. Our hypothesis is that the structure codes are extracted by a shallow network and contain the positional and geometric information of inputs. So the interpolation between the low-level features is not able to preserve semantic smoothness or consistency.

Acknowledgement. Yi Yang acknowledges support from Data to Decision Cooperative Research Centre.