

Deep Image Spatial Transformation for Person Image Generation

Yurui Ren^{1,2} Xiaoming Yu^{1,2} Junming Chen^{1,2} Thomas H. Li^{3,1} Ge Li \bowtie ^{1,2}

¹School of Electronic and Computer Engineering, Peking University ²Peng Cheng Laboratory
³Advanced Institute of Information Technology, Peking University

{yrren, xiaomingyu, junming.chen}@pku.edu.cn tli@aiit.org.cn geli@ece.pku.edu.cn

Abstract

Pose-guided person image generation is to transform a source person image to a target pose. This task requires spatial manipulations of source data. However, Convolutional Neural Networks are limited by the lack of ability to spatially transform the inputs. In this paper, we propose a differentiable global-flow local-attention framework to reassemble the inputs at the feature level. Specifically, our model first calculates the global correlations between sources and targets to predict flow fields. Then, the flowed local patch pairs are extracted from the feature maps to calculate the local attention coefficients. Finally, we warp the source features using a content-aware sampling method with the obtained local attention coefficients. The results of both subjective and objective experiments demonstrate the superiority of our model. Besides, additional results in video animation and view synthesis show that our model is applicable to other tasks requiring spatial transformation. Our source code is available at <https://github.com/RenYurui/Global-Flow-Local-Attention>.

1 . Introduction

Image spatial transformation can be used to deal with the generation task where the output images are the spatial deformation versions of the input images. Such deformation can be caused by object motions or viewpoint changes. Many conditional image generation tasks can be seen as a type of spatial transformation tasks. For example, pose-guided person image generation [20, 26, 28, 42, 29, 30] transforms a person image from a source pose to a target pose while retaining the appearance details. As shown in Figure 1, this task can be tackled by reasonably reassembling the input data in the spatial domain.

However, Convolutional Neural Networks (CNNs) are inefficient to spatially transform the inputs. CNNs calculate the outputs with a particular form of parameter sharing, which leads to an important property called equiv-



Figure 1. The visualization of data spatial transformation. For each image pair, the left image is the generated result of our model, while the right image is the input source image. Our model spatially transforms the information from sources to targets at the features level. The heat maps indicate the attention coefficients.

ariance to transformation [5]. It means that if the input spatially shifts the output shifts in the same way. This property can benefit tasks such as segmentation [4, 8], detection [27, 11] and image translation with aligned structures [12, 36] etc. However, it limits the networks by lacking abilities to spatially rearrange the input data. Spatial Transformer Networks (STN) [13] solves this problem by introducing a Spatial Transformer module to standard neural networks. This module regresses global transformation parameters and warps input features with an affine transformation. However, since it assumes a global affine transformation between sources and targets, this method cannot deal with the transformations of non-rigid objects.

Attention mechanism [32, 37] allows networks to take use of non-local information, which gives networks abilities to build long-term correlations. It has been proved to be efficient in many tasks such as natural language processing [32], image recognition [34, 10], and image generation [37]. However, for spatial transformation tasks in which target images are the deformation results of source images, each output position has a clear one-to-one relationship with the source positions. Therefore, the attention coefficient matrix between the source and target should be a sparse matrix instead of a dense matrix.

Flow-based operation forces the attention coefficient matrix to be a sparse matrix by sampling a very local source patch for each output position. These methods predict 2-D

coordinate offsets specifying which positions in the sources could be sampled to generate the targets. However, in order to stabilize the training, most of the flow-based methods [41, 3] warp input data at the pixel level, which limits the networks to be unable to generate new contents. Meanwhile, large motions are difficult to be extracted due to the requirement of generating full-resolution flow fields [22]. Warping the inputs at the feature level can solve these problems. However, the networks are easy to be stuck within bad local minima [23, 35] due to two reasons. (1) The input features and flow fields are mutually constrained. The input features can not obtain reasonable gradients without correct flow fields. The network also cannot extract similarities to generate correct flow fields without reasonable features. (2) The poor gradient propagation provided by the commonly used Bilinear sampling method further lead to instability in training [14, 23]. See Section A for more discussion.

In order to deal with these problems, in this paper, we combine flow-based operation with attention mechanisms. We propose a novel global-flow local-attention framework to force each output location to be only related to a local feature patch of sources. The architecture of our model can be found in Figure 2. Specifically, our network can be divided into two parts: Global Flow Field Estimator and Local Neural Texture Renderer. The Global Flow Filed Estimator is responsible for extracting the global correlations and generating flow fields. The Local Neural Texture Renderer is used to sample vivid source textures to targets according to the obtained flow fields. To avoid the poor gradient propagation of the Bilinear sampling, we propose a local attention mechanism as a content-aware sampling method. We compare our model with several state-of-the-art methods. The results of both subjective and objective experiments show the superior performance of our model. We also conduct comprehensive ablation studies to verify our hypothesis. Besides, we apply our model to other tasks requiring spatial transformation manipulation including view synthesis and video animation. The results show the versatility of our module. The main contributions of our paper can be summarized as:

- A global-flow local-attention framework is proposed for the pose-guided person image generation task. Experiments demonstrate the effectiveness of the proposed method.
- The carefully-designed framework and content-aware sampling operation ensure that our model is able to warp and reasonably reassemble the input data at the feature level. This operation not only enables the model to generate new contents, but also reduces the difficulty of the flow field estimation task.
- Additional experiments on view synthesis and video animation show that our model can be flexibly applied to different tasks requiring spatial transformation.

2 . Related Work

Pose-guided Person Image Generation. An early attempt [20] on the pose-guided person image generation task proposes a two-stage network to first generate a coarse image with target pose and then refine the results in an adversarial way. Essner *et al.* [2] try to disentangle the appearance and pose of person images. Their model enables both conditional image generation and transformation. However, they use U-Net based skip connections, which may lead to feature misalignments. Siarohin *et al.* [26] solve this problem by introducing deformable skip connections to spatially transform the textures. It decomposes the overall deformation by a set of local affine transformations (*e.g.* arms and legs *etc.*). Although it works well in person image generation, the requirement of the pre-defined transformation components limits its application. Zhu *et al.* [42] propose a more flexible method by using a progressive attention module to transform the source data. However, useful information may be lost during multiple transfers, which may result in blurry details. Han *et al.* [7] use a flow-based method to transform the source information. However, they warp the sources at the pixel level, which means that further refinement networks are required to fill the holes of occlusion contents. Liu *et al.* [18] and Li *et al.* [16] warp the inputs at the feature level. But both of them need additional 3D human models to calculate the flow fields between sources and targets, which limits the application of these models. Our model does not require any supplementary information and obtains the flow fields in an unsupervised manner.

Image Spatial Transformation. Many methods have been proposed to enable the spatial transformation capability of Convolutional Neural Networks. Jaderberg *et al.* [13] introduce a differentiable Spatial Transformer module that estimates global transformation parameters and warps the features with affine transformation. Several variants have been proposed to improve the performance. Zhang *et al.* add controlling points for free-form deformation [38]. The model proposed in paper [17] sends the transformation parameters instead of the transformed features to the network to avoid sampling errors. Jiang *et al.* [14] demonstrate the poor gradient propagation of the commonly used Bilinear sampling. They propose a linearized multi-sampling method for spatial transformation.

Flow-based methods are more flexible than affine transformation methods. They can deal with complex deformations. Appearance flow [41] predicts flow fields and generates the targets by warping the sources. However, it warps image pixels instead of features. This operation limits the model to be unable to generate new contents. Besides, it requires the model to predict flow fields with the same resolution as the result images, which makes it difficult for the model to capture large motions [43, 22]. Vid2vid [33] deals with these problems by predicting the ground-truth

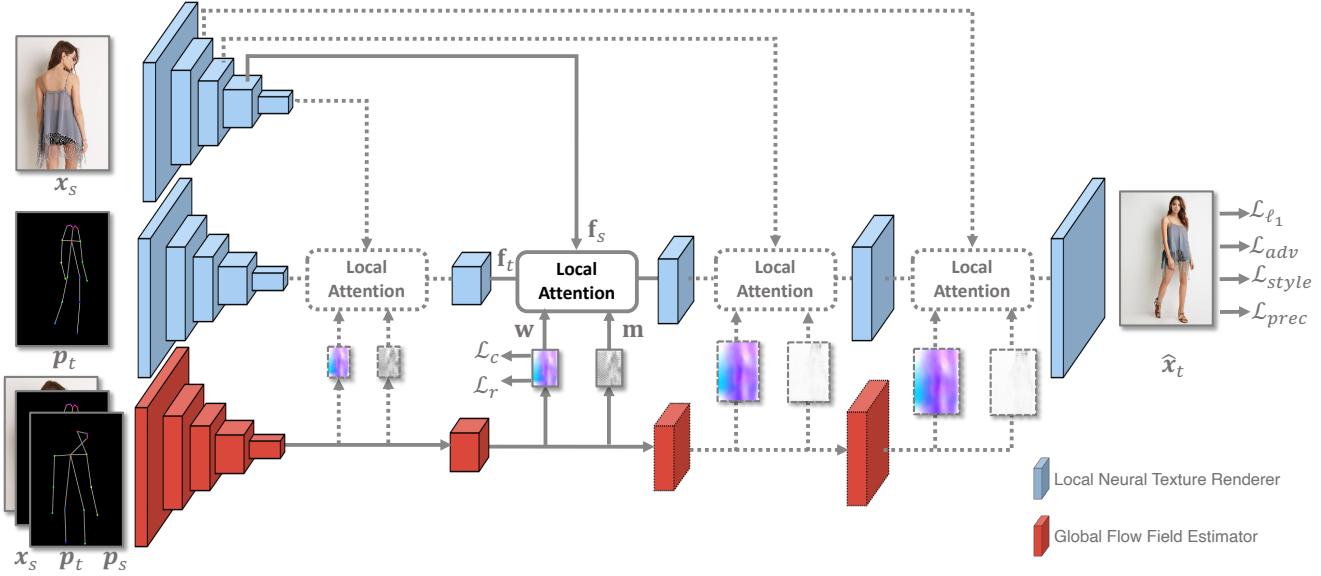


Figure 2. Overview of our model. The Global Flow Field Estimator is used to generate flow fields. The Local Neural Texture Renderer yields results by spatially transforming the source features using local attention. Dotted lines indicate that our local attention module can be used at different scales.

flow fields using FlowNet [3] first and then trains their flow estimator in a supervised manner. They also use a generator for occluded content generation. Warping the sources at the feature level can avoid these problems. In order to stabilize the training, some papers propose to obtain the flow-fields by using some assumptions or supplementary information. Paper [25] assumes that keypoints are located on object parts that are locally rigid. They generate dense flow fields from sparse keypoints. Papers [18, 16] use the 3D human models and the visibility maps to calculate the flow fields between sources and targets. Paper [23] proposes a sampling correctness loss to constraint flow fields and achieve good results.

3 . Our Approach

For the pose-guided person image generation task, target images are the deformation results of source images, which means that each position of targets is only related to a local region of sources. Therefore, we design a global-flow local-attention framework to reasonably sample and reassemble source features. Our network architecture is shown in Figure 2. It consists of two modules: *Global Flow Field Estimator* F and *Local Neural Texture Renderer* G . The Global Flow Field Estimator is responsible for estimating the motions between sources and targets. It generates global flow fields \mathbf{w} and occlusion masks \mathbf{m} for the local attention blocks. With \mathbf{w} and \mathbf{m} , the Local Neural Texture Renderer renders the target images with vivid source features using the local attention blocks. We describe the

details of these modules in the following sections. Please note that to simplify the notations, we describe the network with a single local attention block. As shown in Figure 2, our model can be extended to use multiple attention blocks at different scales.

3 .1. Global Flow Field Estimator

Let \mathbf{p}_s and \mathbf{p}_t denote the structure guidance of the source image \mathbf{x}_s and the target image \mathbf{x}_t respectively. Global Flow Field Estimator F is trained to predict the motions between \mathbf{x}_s and \mathbf{x}_t in an unsupervised manner. It takes \mathbf{x}_s , \mathbf{p}_s and \mathbf{p}_t as inputs and generates flow fields \mathbf{w} and occlusion masks \mathbf{m} .

$$\mathbf{w}, \mathbf{m} = F(\mathbf{x}_s, \mathbf{p}_s, \mathbf{p}_t) \quad (1)$$

where \mathbf{w} contains the coordinate offsets between sources and targets. The occlusion mask \mathbf{m} with continuous values between 0 and 1 indicates whether the information of a target position exists in the sources. We design F as a fully convolutional network. \mathbf{w} and \mathbf{m} share all weights of F other than their output layers.

As the labels of the flow fields \mathbf{w} are always unavailable in this task, we use the sampling correctness loss proposed by [23] to constraint \mathbf{w} . It calculates the similarity between the warped source feature and ground-truth target feature at the VGG feature level. Let \mathbf{v}_s and \mathbf{v}_t denote the features generated by a specific layer of VGG19. $\mathbf{v}_{s,w} = \mathbf{w}(\mathbf{v}_s)$ is the warped results of the source feature \mathbf{v}_s using \mathbf{w} . The sampling correctness loss calculates the relative cosine sim-

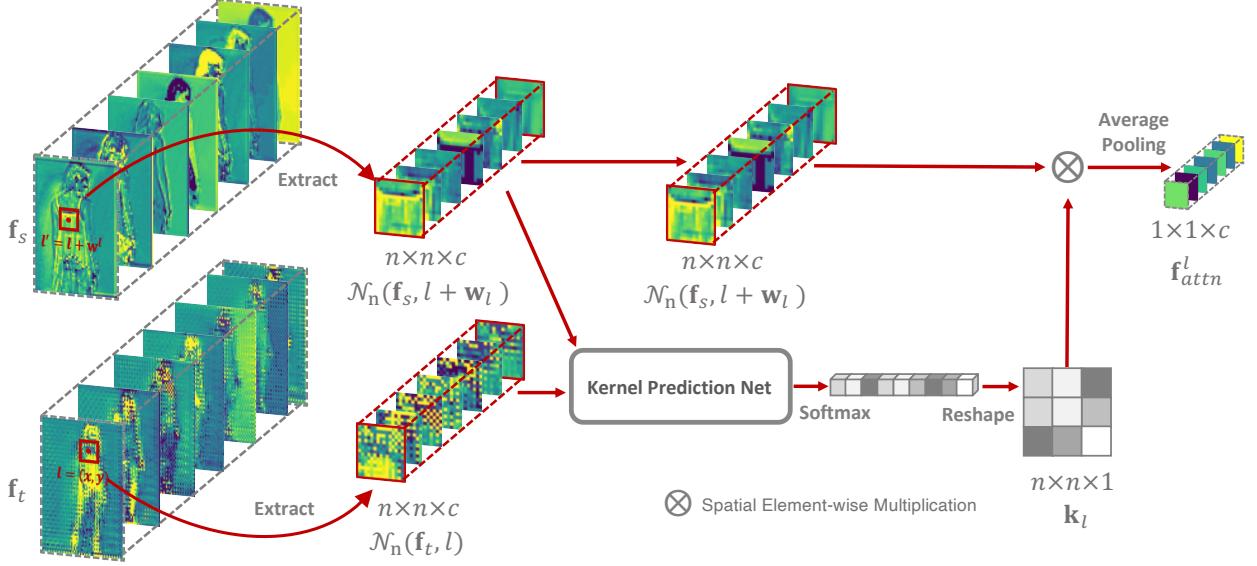


Figure 3. Overview of our Local Attention. We first extract the feature patch pair from the source and target according to the flow fields. Then the context-aware sampling kernel is calculated by the kernel prediction net. Finally, we sample the source feature and obtain the warped result located at l .

ilarity between $\mathbf{v}_{s,w}$ and \mathbf{v}_t .

$$\mathcal{L}_c = \frac{1}{N} \sum_{l \in \Omega} \exp\left(-\frac{\mu(\mathbf{v}_{s,w}^l, \mathbf{v}_t^l)}{\mu_{max}^l}\right) \quad (2)$$

where $\mu(*)$ denotes the cosine similarity. Coordinate set Ω contains all N positions in the feature maps, and $\mathbf{v}_{s,w}^l$ denotes the feature of $\mathbf{v}_{s,w}$ located at the coordinate $l = (x, y)$. The normalization term μ_{max}^l is calculated as

$$\mu_{max}^l = \max_{l' \in \Omega} \mu(\mathbf{v}_s^{l'}, \mathbf{v}_t^l) \quad (3)$$

It is used to avoid the bias brought by occlusion.

The sampling correctness loss can constrain the flow fields to sample semantically similar regions. However, as the deformations of image neighborhoods are highly correlated, it would benefit if we could extract this relationship. Therefore, we further add a regularization term to our flow fields. This regularization term is used to punish local regions where the transformation is not an affine transformation. Let \mathbf{c}_t be the 2D coordinate matrix of the target feature map. The corresponding source coordinate matrix can be written as $\mathbf{c}_s = \mathbf{c}_t + \mathbf{w}$. We use $\mathcal{N}_n(\mathbf{c}_t, l)$ to denote local $n \times n$ patch of \mathbf{c}_t centered at the location l . Our regularization assumes that the transformation between $\mathcal{N}_n(\mathbf{c}_t, l)$ and $\mathcal{N}_n(\mathbf{c}_s, l)$ is an affine transformation.

$$\mathbf{T}_l = \mathbf{A}_l \mathbf{S}_l = \begin{bmatrix} \theta_{11} & \theta_{12} & \theta_{13} \\ \theta_{21} & \theta_{22} & \theta_{23} \end{bmatrix} \mathbf{S}_l \quad (4)$$

where $\mathbf{T}_l = \begin{bmatrix} x_1 & x_2 & \dots & x_{n \times n} \\ y_1 & y_2 & \dots & y_{n \times n} \end{bmatrix}$ with each coordinate $(x_i, y_i) \in \mathcal{N}_n(\mathbf{c}_t, l)$ and $\mathbf{S}_l = \begin{bmatrix} x_1 & x_2 & \dots & x_{n \times n} \\ y_1 & y_2 & \dots & y_{n \times n} \\ 1 & 1 & \dots & 1 \end{bmatrix}$ with each coordinate $(x_i, y_i) \in \mathcal{N}_n(\mathbf{c}_s, l)$. The estimated affine transformation parameters $\hat{\mathbf{A}}_l$ can be solved using the least-squares estimation as

$$\hat{\mathbf{A}}_l = (\mathbf{S}_l^H \mathbf{S}_l)^{-1} \mathbf{S}_l^H \mathbf{T}_l \quad (5)$$

Our regularization is calculated as the ℓ_2 distance of the error.

$$\mathcal{L}_r = \sum_{l \in \Omega} \left\| \mathbf{T}_l - \hat{\mathbf{A}}_l \mathbf{S}_l \right\|_2^2 \quad (6)$$

3.2. Local Neural Texture Renderer

With the flow fields \mathbf{w} and occlusion masks \mathbf{m} , our Local Neural Texture Renderer G is responsible for generating the results by spatially transforming the information from sources to targets. It takes \mathbf{x}_s , \mathbf{p}_t , \mathbf{w} and \mathbf{m} as inputs and generate the result image $\hat{\mathbf{x}}_t$.

$$\hat{\mathbf{x}}_t = G(\mathbf{x}_s, \mathbf{p}_t, \mathbf{w}, \mathbf{m}) \quad (7)$$

Specifically, the information transformation occurs in the local attention module. As shown in Figure 2, this module works as a neural renderer where the target bones are rendered by the neural textures of the sources. Let \mathbf{f}_t and \mathbf{f}_s represent the extracted features of target bones \mathbf{p}_t and

	DeepFashion			Market-1501				Number of Parameters
	FID	LPIPS	JND	FID	LPIPS	Mask-LPIPS	JND	
Def-GAN	18.457	0.2330	9.12%	25.364	0.2994	0.1496	23.33%	82.08M
VU-Net	23.667	0.2637	2.96%	20.144	0.3211	0.1747	24.48%	139.36M
Pose-Attn	20.739	0.2533	6.11%	22.657	0.3196	0.1590	16.56%	41.36M
Intr-Flow	16.314	0.2131	12.61%	27.163	0.2888	0.1403	30.85%	49.58M
Ours	10.573	0.2341	24.80%	19.751	0.2817	0.1482	27.81%	14.04M

Table 1. The evaluation results compared with several state-of-the-art methods including Def-GAN [26], VU-Net [2], Pose-Attn[42], and Intr-Flow [16] over dataset DeepFashion [19] and Market-1501 [40]. FID [9] and LPIPS [39] are objective metrics. JND is obtained by human subjective studies. It represents the probability that the generated images are mistaken for real images.

source images \mathbf{x}_s respectively. We first extract local patches $\mathcal{N}_n(\mathbf{f}_t, l)$ and $\mathcal{N}_n(\mathbf{f}_s, l + \mathbf{w}^l)$ from \mathbf{f}_t and \mathbf{f}_s respectively. The patch $\mathcal{N}_n(\mathbf{f}_s, l + \mathbf{w}^l)$ is extracted using bilinear sampling as the coordinates may not be integers. Then, a kernel prediction network M is used to predict local $n \times n$ kernel \mathbf{k}_l as

$$\mathbf{k}_l = M(\mathcal{N}_n(\mathbf{f}_s, l + \mathbf{w}^l), \mathcal{N}_n(\mathbf{f}_t, l)) \quad (8)$$

We design M as a fully connected network, where the local patches $\mathcal{N}_n(\mathbf{f}_s, l + \mathbf{w}^l)$ and $\mathcal{N}_n(\mathbf{f}_t, l)$ are directly concatenated as the inputs. The softmax function is used as the non-linear activation function of the output layer of M . This operation forces the sum of \mathbf{k}_l to 1, which enables the stability of gradient backward. Finally, the flowed feature located at coordinate $l = (x, y)$ is calculated using a content-aware attention over the extracted source feature patch $\mathcal{N}_n(\mathbf{f}_s, l + \mathbf{w}^l)$.

$$\mathbf{f}_{attn}^l = P(\mathbf{k}_l \otimes \mathcal{N}_n(\mathbf{f}_s, l + \mathbf{w}^l)) \quad (9)$$

where \otimes denotes the element-wise multiplication over the spatial domain and P represents the global average pooling operation. The warped feature map \mathbf{f}_{attn} is obtained by repeating the previous steps for each location l .

However, not all contents of target images can be found in source images because of occlusion or movements. In order to enable generating new contents, the occlusion mask \mathbf{m} with continuous value between 0 and 1 is used to select features between \mathbf{f}_{attn} and \mathbf{f}_t .

$$\mathbf{f}_{out} = (\mathbf{1} - \mathbf{m}) * \mathbf{f}_t + \mathbf{m} * \mathbf{f}_{attn} \quad (10)$$

We train the network using a joint loss consisting of a reconstruction ℓ_1 loss, adversarial loss, perceptual loss, and style loss. The reconstruction ℓ_1 loss is written as

$$\mathcal{L}_{\ell_1} = \|\mathbf{x}_t - \hat{\mathbf{x}}_t\|_1 \quad (11)$$

The generative adversarial framework [6] is employed to mimic the distributions of the ground-truth \mathbf{x}_t . The adversarial loss is written as

$$\begin{aligned} \mathcal{L}_{adv} &= \mathbb{E}[\log(1 - D(G(\mathbf{x}_s, \mathbf{p}_t, \mathbf{w}, \mathbf{m}))) \\ &+ \mathbb{E}[\log D(\mathbf{x}_t)] \end{aligned} \quad (12)$$

where D is the discriminator of the Local Neural Texture Renderer G . We also use the perceptual loss and style loss introduced by [15]. The perceptual loss calculates ℓ_1 distance between activation maps of a pre-trained network. It can be written as

$$\mathcal{L}_{perc} = \sum_i \|\phi_i(\mathbf{x}_t) - \phi_i(\hat{\mathbf{x}}_t)\|_1 \quad (13)$$

where ϕ_i is the activation map of the i -th layer of a pre-trained network. The style loss calculates the statistic error between the activation maps as

$$\mathcal{L}_{style} = \sum_j \left\| G_j^\phi(\mathbf{x}_t) - G_j^\phi(\hat{\mathbf{x}}_t) \right\|_1 \quad (14)$$

where G_j^ϕ is the Gram matrix constructed from activation maps ϕ_j . We train our model using the overall loss as

$$\mathcal{L} = \lambda_c \mathcal{L}_c + \lambda_r \mathcal{L}_r + \lambda_{\ell_1} \mathcal{L}_{\ell_1} + \lambda_a \mathcal{L}_{adv} + \lambda_p \mathcal{L}_{prec} + \lambda_s \mathcal{L}_{style} \quad (15)$$

4 . Experiments

4 . 1. Implementation Details

Datasets. Two datasets are used in our experiments: person re-identification dataset Market-1501 [40] and DeepFashion In-shop Clothes Retrieval Benchmark [19]. Market-1501 contains 32668 low-resolution images (128×64). The images vary in terms of the viewpoints, background, illumination etc. The DeepFashion dataset contains 52712 high-quality model images with clean backgrounds. We split the datasets with the same method as that of [42]. The personal identities of the training and testing sets do not overlap.

Metrics. We use Learned Perceptual Image Patch Similarity (LPIPS) proposed by [39] to calculate the reconstruction error. LPIPS computes the distance between the generated images and reference images at the perceptual domain. It indicates the perceptual difference between the inputs. Meanwhile, Fréchet Inception Distance [9] (FID) is employed to measure the realism of the generated images. It calculates the Wasserstein-2 distance between distributions of the generated images and ground-truth images. Besides,



Figure 4. The qualitative comparisons with several state-of-the-art models including Def-GAN [26], VU-Net [2], Pose-Attn[42], and Intr-Flow [16]. The left part shows the results of the Fashion dataset. The right part shows the results of the Market-1501 dataset.

we perform a Just Noticeable Difference (JND) test to evaluate the subjective quality. Volunteers are asked to choose the more realistic image from the data pair of ground-truth and generated images.

Network Implementation and Training Details. Basically, auto-encoder structures are employed to design our F and G . The residual block is used as the basic component of these models. We train our model using 256×256 images for the Fashion dataset. Two local attention blocks are used for feature maps with resolutions as 32×32 and 64×64 . The extracted local patch sizes are 3 and 5 respectively. For Market-1501, we use 128×64 images with a single local attention block at the feature maps with resolution as 32×16 . The extracted patch size is 3. We train our model in stages. The Flow Field Estimator is first trained to generate flow fields. Then we train the whole model in an end-to-end manner. We adopt the ADAM optimizer with the learning rate as 10^{-4} . The batch size is set to 8 for all experiments.

4 .2. Comparisons

We compare our method with several state-of-the-art methods including Def-GAN [26], VU-Net [2], Pose-Attn[42] and Intr-Flow [16]. The quantitative evaluation results are shown in Table 1. For the Market-1501 dataset, we

follow the previous work [20] to calculate the mask-LPIPS to alleviate the influence of the backgrounds. It can be seen that our model achieves competitive results in both datasets, which means that our model can generate realistic results with fewer perceptual reconstruction errors.

As the subjective metrics may not be sensitive to some artifacts, its results may mismatch with the actual subjective perceptions. Therefore, we implement a just noticeable difference test on Amazon Mechanical Turk (MTurk). This experiment requires volunteers to choose the more realistic image from image pairs of real and generated images. The test is performed over 800 images for each model and dataset. Each image is compared 5 times by different volunteers. The evaluation results are shown in Table 1. It can be seen that our model achieves the best result in the challenging Fashion dataset and competitive results in the Market-1501 dataset.

The typical results of different methods are provided in Figure 4. For the Fashion dataset, VU-Net and Pose-Attn struggle to generate complex textures since these models lack efficient spatial transformation blocks. Def-GAN defines local affine transformation components (*e.g.* arms and legs *etc.*). This model can generate correct textures. However, the pre-defined affine transformations are not sufficient to represent complex spatial variance, which limits the

	Flow-Based	Content-aware Sampling	FID	LPIPS
Baseline	N	-	16.008	0.2473
Global-Attn	N	-	18.616	0.2575
Bi-Sample	Y	N	12.143	0.2406
Full Model	Y	Y	10.573	0.2341

Table 2. The evaluation results of the ablation study.

performance of the model. Flow-based model Intr-Flow is able to generate vivid textures for front pose images. However, it may fail to generate realistic results for side pose images due to the requirement of generating full-resolution flow fields. Meanwhile, this model needs 3D human models to generate the ground-truth flow fields for training. Our model regresses flow fields in an unsupervised manner. It can generate realistic images with not only the correct global pattern but also the vivid details such as the lace of clothes and the shoelace. For the Market-1501 Dataset, our model can generate correct pose with vivid backgrounds. Artifacts can be found in the results of competitors, such as the sharp edges in Pose-Attn and the halo effects in Def-GAN. Please refer to Section C for more comparison results.

The numbers of model parameters are also provided to evaluate the computation complexity in Table 1. Thanks to our efficient attention blocks, our model does not require a large number of convolution layers. Thus, we can achieve high performance with less than half of the parameters of the competitors.

4 .3. Ablation Study

In this subsection, we train several ablation models to verify our assumptions and evaluate the contribution of each component.

Baseline. Our baseline model is an auto-encoder convolutional network. We do not use any attention blocks in this model. Images \mathbf{x}_s , \mathbf{p}_t , \mathbf{p}_s are directly concatenated as the model inputs.

Global Attention Model (Global-Attn). The Global-Attn model is designed to compare the global-attention block with our local-attention block. We use a similar network architecture as our Local Neural Texture Renderer G for this model. The local attention blocks are replaced by global attention blocks where the attention coefficients are calculated by the similarities between the source features \mathbf{f}_s and target features \mathbf{f}_t .

Bilinear Sampling Model (Bi-Sample). The Bi-Sample model is designed to evaluate the contribution of our content-aware sampling method described in Section 3 .2. Both the Global Flow Field Estimator F and Local Neural Texture Renderer G are employed in this model. However, we use the Bilinear sampling as the sampling method in model G .

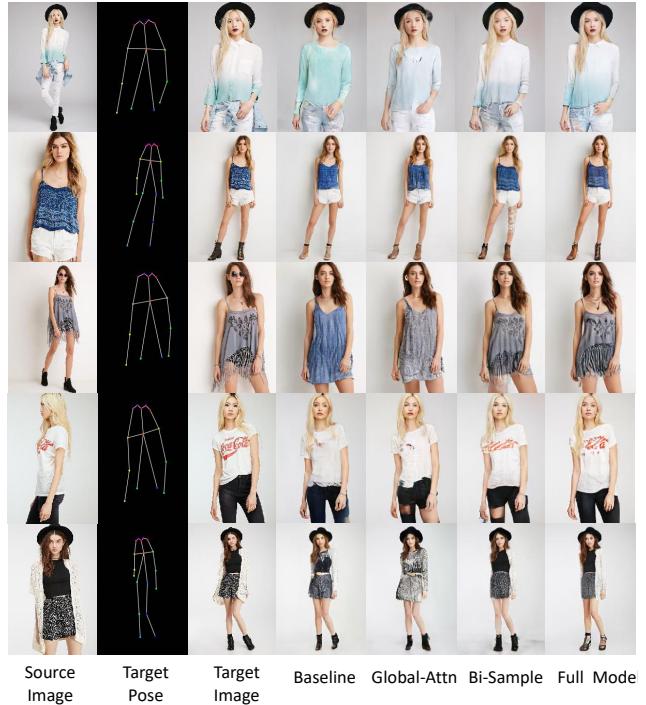


Figure 5. Qualitative results of the ablation study.



Figure 6. The visualization results of different attention modules. The red rectangles indicate the target locations. The heat maps show the attention coefficients. Blue represents low weights.

Full Model (Ours). We use our proposed global-flow local-attention framework in this model.

The evaluation results of the ablation study are shown in Table 2. Compared with the Baseline, the performance of the Global-Attn model is degraded, which means that unreasonable attention block cannot efficiently transform the information. Improvements can be obtained by using flow-

based methods such as the Bi-Sample model and our Full model which force the attention coefficient matrix to be a sparse matrix. However, the Bi-Sample model uses a pre-defined sampling method with a limited sampling receptive field, which may lead to unstable training. Our full model uses a content-aware sampling operation with an adjustable receptive field, which brings further performance gain.

Subjective comparison of these ablation models can be found in Figure 5. It can be seen that the Baseline and Global-Attn model generate correct structures. However, the textures of the source images are not well-maintained. The possible explanation is that these models generate images by first extracting global features and then propagating the information to specific locations. This process leads to the loss of details. The flow-based methods spatially transform the features. They are able to reconstruct vivid details. However, the Bi-Sample model uses the pre-defined Bilinear sampling method. It cannot find the exact sampling locations, which leads to artifacts in the final results.

We further provide the visualization of the attention maps in Figure 6. It can be seen that the Global-Attn model struggles to exclude irrelevant information. Therefore, the extracted features are hard to be used to generate specific textures. The Bi-Sample model assigns a local patch for each generated location. However, incorrect features are often flowed due to the limited sampling receptive field. Our Full model using the content-aware sampling method can flexibly change the sampling weights and avoid the artifacts.

5 . Application on Other Tasks

In this section, we demonstrate the versatility of our global-flow local-attention module. Since our model does not require any additional information other than images and structure guidance, it can be flexibly applied to tasks requiring spatial transformation. Two example tasks are shown as follows.

View Synthesis. View synthesis requires generating novel views of objects or scenes based on arbitrary input views. Since the appearance of different views is highly correlated, the existing information can be reassembled to generate the targets. The ShapeNet dataset [1] is used for training. We generate novel target views using single view input. The results can be found in Figure 7. We provide the results of appearance flow as a comparison. It can be seen that appearance flow struggles to generate occluded contents as they warp image pixels instead of features. Our model generates reasonable results.

Image Animation. Given an input image and a driving video sequence depicting the structure movements, the image animation task requires generating a video containing the specific movements. This task can be solved by spatially moving the appearance of the sources. We train our model with the real videos in the FaceForensics dataset [24],

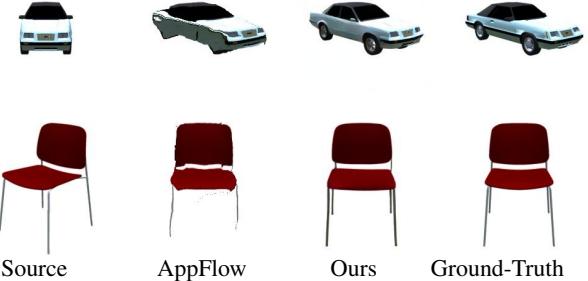


Figure 7. Qualitative results of the view synthesis task. We show the results of our model and appearance flow [41] model. *Click on the image to start the animation in a browser.*



Figure 8. Qualitative results of the image animation task. Our model generates the results video using reference image and edge guidance. *Click on the image to play the video clip in a browser.*

which contains 1000 videos of news briefings from different reporters. The face regions are cropped for this task. We use the edge maps as the structure guidance. For each frame, the input source frame and the previous generated n frames are used as the references. The flow fields are calculated for each reference. The results can be found in Figure 8. It can be seen that our model generates realistic results with vivid movements. More applications can be found in Section C .

6 . Conclusion

In this paper, we solve the person image generation task with deep spatial transformation. We analyze the specific reasons causing instable training when warping and transforming sources at the feature level. Targeted solution global-flow local-attention framework is proposed to enable our model to reasonably reassemble the source neural textures. Experiments show that our model can generate target images with correct poses while maintaining vivid details. In addition, the ablation study shows that our improvements help the network find reasonable sampling positions. Finally, we show that our model can be easily extended to address other spatial deformation tasks such as view synthesis and video animation.

References

- [1] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet:

- An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015. 8
- [2] Patrick Esser, Ekaterina Sutter, and Björn Ommer. A variational u-net for conditional appearance and shape generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8857–8866, 2018. 2, 5, 6, 13
- [3] Philipp Fischer, Alexey Dosovitskiy, Eddy Ilg, Philip Häusser, Caner Hazirbaş, Vladimir Golkov, Patrick Van der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. *arXiv preprint arXiv:1504.06852*, 2015. 2, 3
- [4] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014. 1
- [5] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016. 1
- [6] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014. 5
- [7] Xintong Han, Xiaojun Hu, Weilin Huang, and Matthew R Scott. Clothflow: A flow-based model for clothed person generation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 10471–10480, 2019. 2
- [8] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 1
- [9] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems*, pages 6626–6637, 2017. 5
- [10] Han Hu, Zheng Zhang, Zhenda Xie, and Stephen Lin. Local relation networks for image recognition. *arXiv preprint arXiv:1904.11491*, 2019. 1
- [11] Jingjia Huang, Nannan Li, Thomas Li, Shan Liu, and Ge Li. Spatial-temporal context-aware online action detection and prediction. *IEEE Transactions on Circuits and Systems for Video Technology*, 2019. 1
- [12] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017. 1
- [13] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. In *Advances in neural information processing systems*, pages 2017–2025, 2015. 1, 2
- [14] Wei Jiang, Weiwei Sun, Andrea Tagliasacchi, Eduard Trulls, and Kwang Moo Yi. Linearized multi-sampling for differentiable image transformation. *Proceedings of the IEEE International Conference on Computer Vision*, 2019. 2
- [15] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*, pages 694–711. Springer, 2016. 5
- [16] Yining Li, Chen Huang, and Chen Change Loy. Dense intrinsic appearance flow for human pose transfer. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 2, 3, 5, 6, 13
- [17] Chen-Hsuan Lin and Simon Lucey. Inverse compositional spatial transformer networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2568–2576, 2017. 2
- [18] Wen Liu, Zhixin Piao, Jie Min, Wenhan Luo, Lin Ma, and Shenghua Gao. Liquid warping gan: A unified framework for human motion imitation, appearance transfer and novel view synthesis. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5904–5913, 2019. 2, 3
- [19] Ziwei Liu, Ping Luo, Shi Qiu, Xiaogang Wang, and Xiaoou Tang. Deepfashion: Powering robust clothes recognition and retrieval with rich annotations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1096–1104, 2016. 5, 13
- [20] Liqian Ma, Xu Jia, Qianru Sun, Bernt Schiele, Tinne Tuytelaars, and Luc Van Gool. Pose guided person image generation. In *Advances in Neural Information Processing Systems*, pages 406–416, 2017. 1, 2, 6
- [21] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. *arXiv preprint arXiv:1802.05957*, 2018. 16
- [22] Anurag Ranjan and Michael J Black. Optical flow estimation using a spatial pyramid network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4161–4170, 2017. 2
- [23] Yurui Ren, Xiaoming Yu, Ruonan Zhang, Thomas H. Li, Shan Liu, and Ge Li. Structureflow: Image inpainting via structure-aware appearance flow. In *IEEE International Conference on Computer Vision (ICCV)*, 2019. 2, 3
- [24] Andreas Rössler, Davide Cozzolino, Luisa Verdoliva, Christian Riess, Justus Thies, and Matthias Nießner. Faceforensics: A large-scale video dataset for forgery detection in human faces. *arXiv preprint arXiv:1803.09179*, 2018. 8
- [25] Aliaksandr Siarohin, Stéphane Lathuilière, Sergey Tulyakov, Elisa Ricci, and Nicu Sebe. Animating arbitrary objects via deep motion transfer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2377–2386, 2019. 3
- [26] Aliaksandr Siarohin, Enver Sangineto, Stéphane Lathuilière, and Nicu Sebe. Deformable gans for pose-based human image generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3408–3416, 2018. 1, 2, 5, 6, 13
- [27] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in neural information processing systems*, pages 568–576, 2014. 1
- [28] Sijie Song, Wei Zhang, Jiaying Liu, and Tao Mei. Unsupervised person image generation with semantic parsing transformation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2357–2366, 2019. 1

- [29] Hao Tang, Dan Xu, Gaowen Liu, Wei Wang, Nicu Sebe, and Yan Yan. Cycle in cycle generative adversarial networks for keypoint-guided image generation. In *Proceedings of the 27th ACM International Conference on Multimedia*, pages 2052–2060, 2019. 1
- [30] Hao Tang, Dan Xu, Yan Yan, Jason J Corso, Philip HS Torr, and Nicu Sebe. Multi-channel attention selection gans for guided image-to-image translation. *arXiv preprint arXiv:2002.01048*, 2020. 1
- [31] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*, 2016. 16
- [32] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017. 1
- [33] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Guilin Liu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. Video-to-video synthesis. *arXiv preprint arXiv:1808.06601*, 2018. 2
- [34] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7794–7803, 2018. 1
- [35] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Generative image inpainting with contextual attention. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5505–5514, 2018. 2
- [36] Xiaoming Yu, Yuanqi Chen, Shan Liu, Thomas Li, and Ge Li. Multi-mapping image-to-image translation via learning disentanglement. In *Advances in Neural Information Processing Systems*, pages 2990–2999, 2019. 1
- [37] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks. *arXiv preprint arXiv:1805.08318*, 2018. 1
- [38] Haoyang Zhang and Xuming He. Deep free-form deformation network for object-mask registration. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4251–4259, 2017. 2
- [39] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 586–595, 2018. 5
- [40] Liang Zheng, Liyue Shen, Lu Tian, Shengjin Wang, Jingdong Wang, and Qi Tian. Scalable person re-identification: A benchmark. In *Proceedings of the IEEE international conference on computer vision*, pages 1116–1124, 2015. 5, 13
- [41] Tinghui Zhou, Shubham Tulsiani, Weilun Sun, Jitendra Malik, and Alexei A Efros. View synthesis by appearance flow. In *European conference on computer vision*, pages 286–301. Springer, 2016. 2, 8, 14, 15
- [42] Zhen Zhu, Tengteng Huang, Baoguang Shi, Miao Yu, Bofei Wang, and Xiang Bai. Progressive pose attention transfer for person image generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2347–2356, 2019. 1, 2, 5, 6, 13
- [43] Xiaoou Tang Yiming Liu Ziwei Liu, Raymond Yeh and Aseem Agarwala. Video frame synthesis using deep voxel flow. In *Proceedings of International Conference on Computer Vision (ICCV)*, October 2017. 2

A . Warping at the Feature Level

The networks are easy to be stuck within bad local minima when warping the inputs at the feature level by using flow-based methods with traditional warping operation. Two main problems are pointed out in our paper. In this section, we further explain these reasons.

Figure A.9 shows the warping process of the traditional Bilinear sampling. For each location $l = (x, y)$ in the output features \mathbf{f}_{out} , a sampling position is assigned by the offsets in flow fields $\mathbf{w}^l = (\Delta x, \Delta y)$. Then, the output feature $\mathbf{f}_{out}^{(x,y)}$ is obtained by sampling the local regions of the input features \mathbf{f}_{in} .

$$\begin{aligned}\mathbf{f}_{out}^{x,y} = & (\lceil \Delta y \rceil - \Delta y)(\lceil \Delta x \rceil - \Delta x)\mathbf{f}_{in}^{x',y'} + (\Delta y - \lfloor \Delta y \rfloor)(\Delta x - \lfloor \Delta x \rfloor)\mathbf{f}_{in}^{x'+1,y'+1} \\ & + (\Delta y - \lfloor \Delta y \rfloor)(\lceil \Delta x \rceil - \Delta x)\mathbf{f}_{in}^{x',y'+1} + (\lceil \Delta y \rceil - \Delta y)(\Delta x - \lfloor \Delta x \rfloor)\mathbf{f}_{in}^{x'+1,y'}\end{aligned}\quad (\text{A.16})$$

where $\lceil \cdot \rceil$ and $\lfloor \cdot \rfloor$ represent round up and round down respectively. Location $(x', y') = (x + \lfloor x \rfloor, y + \lfloor y \rfloor)$. Then, we can obtain the backward gradients. The gradients of flow fields is

$$\frac{\partial \mathbf{f}_{out}^{x,y}}{\partial \Delta x} = (\lceil \Delta y \rceil - \Delta y)(\mathbf{f}_{in}^{x'+1,y'} - \mathbf{f}_{in}^{x',y'}) + (\Delta y - \lfloor \Delta y \rfloor)(\mathbf{f}_{in}^{x'+1,y'+1} - \mathbf{f}_{in}^{x',y'+1}) \quad (\text{A.17})$$

the $\frac{\partial \mathbf{f}_{out}^{x,y}}{\partial \Delta y}$ can be obtained in a similar way. The gradients of the input features can be written as

$$\frac{\partial \mathbf{f}_{out}^{x,y}}{\partial \mathbf{f}_{in}^{x',y'}} = (\lceil \Delta y \rceil - \Delta y)(\lceil \Delta x \rceil - \Delta x) \quad (\text{A.18})$$

The other items can be obtained in a similar way.

Let us first explain the first problem. Different from image pixels, the image features are changing during the training process. The flow fields need reasonable input features to obtain correct gradients. As shown in Equation 17, their gradients are calculated by the difference between adjacent features. If the input features are meaningless, the network cannot obtain correct flow fields. Meanwhile, according to Equation 18, the gradients of the input features are calculated by the offsets. They cannot obtain reasonable gradients without correct flow fields. Imagine the worst case that the model only uses the warped results as output and does not use any skip connections, the warp operation stops the gradient from back-propagating.

Another problem is caused by the limited receptive field of Bilinear sampling. Suppose we have got meaningful input features \mathbf{f}_{in} by pre-training, we still cannot obtain stable gradient propagation. According to Equation 17, the gradient of flow fields is calculated by the difference between adjacent features. However, since the adjacent features are extracted from adjacent image patches, they often have strong correlations (*i.e.* $\mathbf{f}_{in}^{x',y'} \approx \mathbf{f}_{in}^{x',y'+1}$). Therefore, the gradients may be small at most positions. Meanwhile, large motions are hard to be captured.

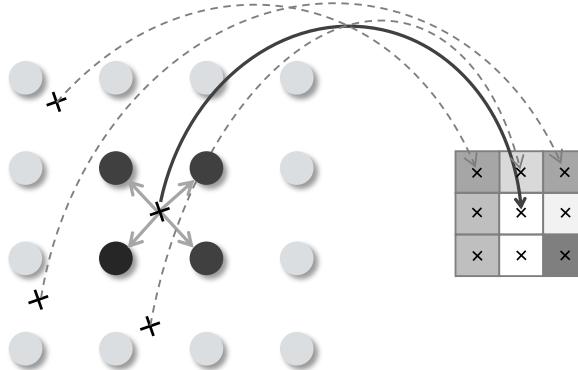


Figure A.9. Bilinear sampling. For each point of the right feature map (target feature map), the flow field assigns a sampling location in the left feature map (source feature map). Then, the sampled feature is calculated through the nearest neighbors by Bilinear interpolation.

B . Analysis of the Regularization loss

In our paper, we mentioned that the deformations of image neighborhoods are highly correlated and proposed a regularization loss to extract this relationship. In this section, we further discuss this regularization loss. Our loss is based on an assumption: although the deformations of the whole images are complex, the deformations of local regions such as arms, clothes are always simple. These deformations can be modeled using affine transformation. Based on this assumption, our regularization loss is proposed to punish local regions where the transformation is not an affine transformation. Figure B.10 gives an example. Our regularization loss first extracts local 2D coordinate matrix $\mathcal{N}(\mathbf{c}_t, l)$ and $\mathcal{N}(\mathbf{c}_s, l)$. Then we estimate the affine transformation parameters $\hat{\mathbf{A}}_l$. Finally, the error is calculated as the regularization loss. Our loss assigns large errors to local regions that do not conform to the affine transformation assumption thereby forcing the network to change the sampling regions.

We provide an ablation study to show the effect of the regularization loss. A model is trained without using the regularization term. The results are shown in Figure B.11. It can be seen that by using our regularization loss the flow fields are more smooth. Unnecessary jumps are avoided. We use the obtained flow fields to warp the source images directly to show the sampling correctness. It can be seen that without using the regularization loss incorrect sampling regions will be assigned due to the sharp jumps of the flow fields. Relatively good results can be obtained by using the affine assumption prior.

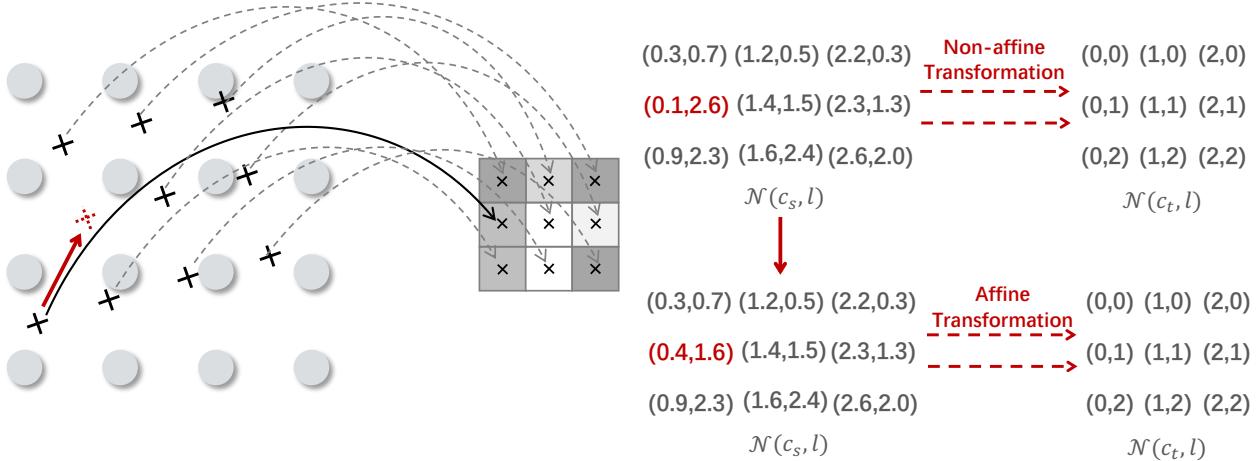


Figure B.10. Our regularization loss assigns large errors to local regions where the transformation is not an affine transformation thereby forcing the network to change the sampling locations.

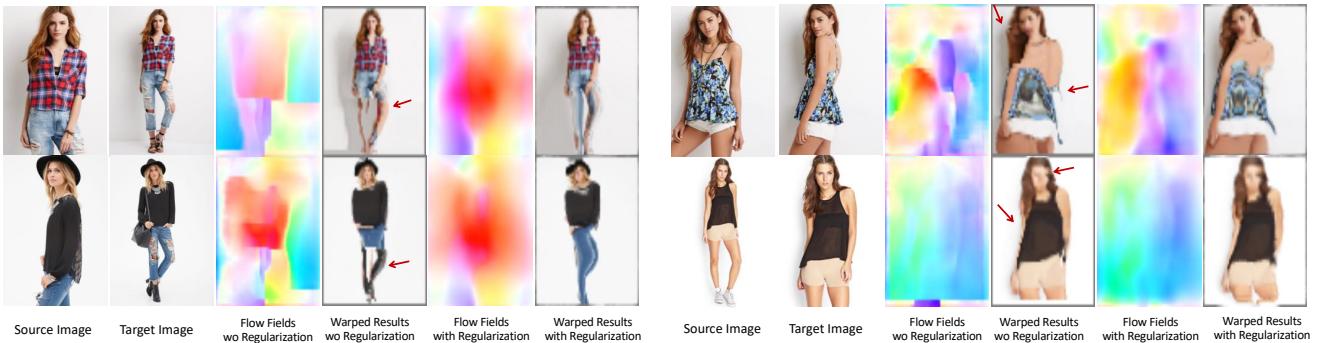


Figure B.11. The flow fields of models trained with and without the regularization loss.

C . Additional Results

C .1. Additional Comparisons with Existing Works

We provide additional comparison results in this section. The qualitative results is shown in Figure C.12.

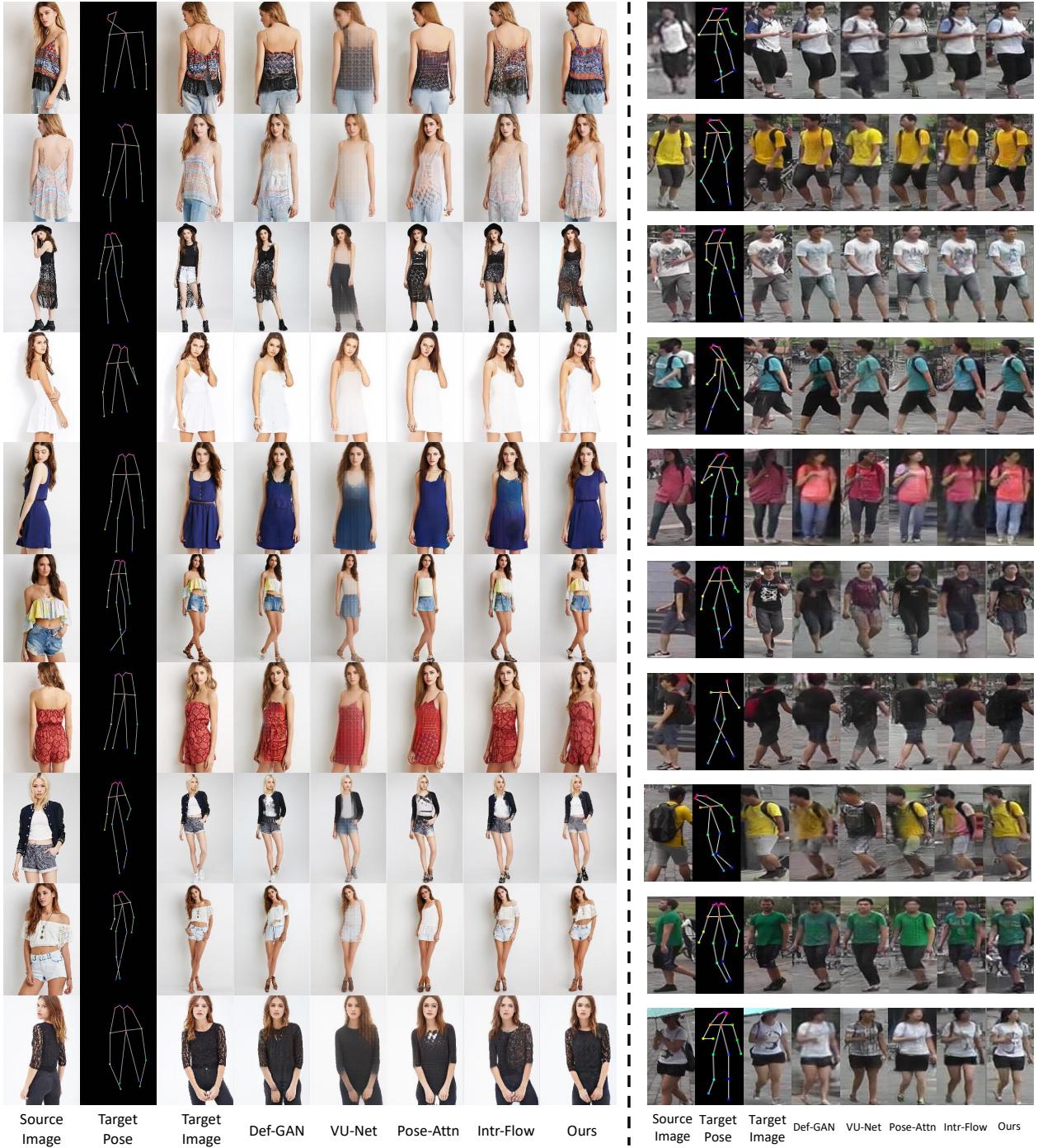


Figure C.12. The qualitative comparisons with several state-of-the-art models including Def-GAN [26], VU-Net [2], Pose-Attn[42], and Intr-Flow [16] over dataset DeepFashion [19] and Market-1501 [40].

C .2. Additional Results of View Synthesis

We provide additional results of the view synthesis task in Figure C.13 and Figure C.14.

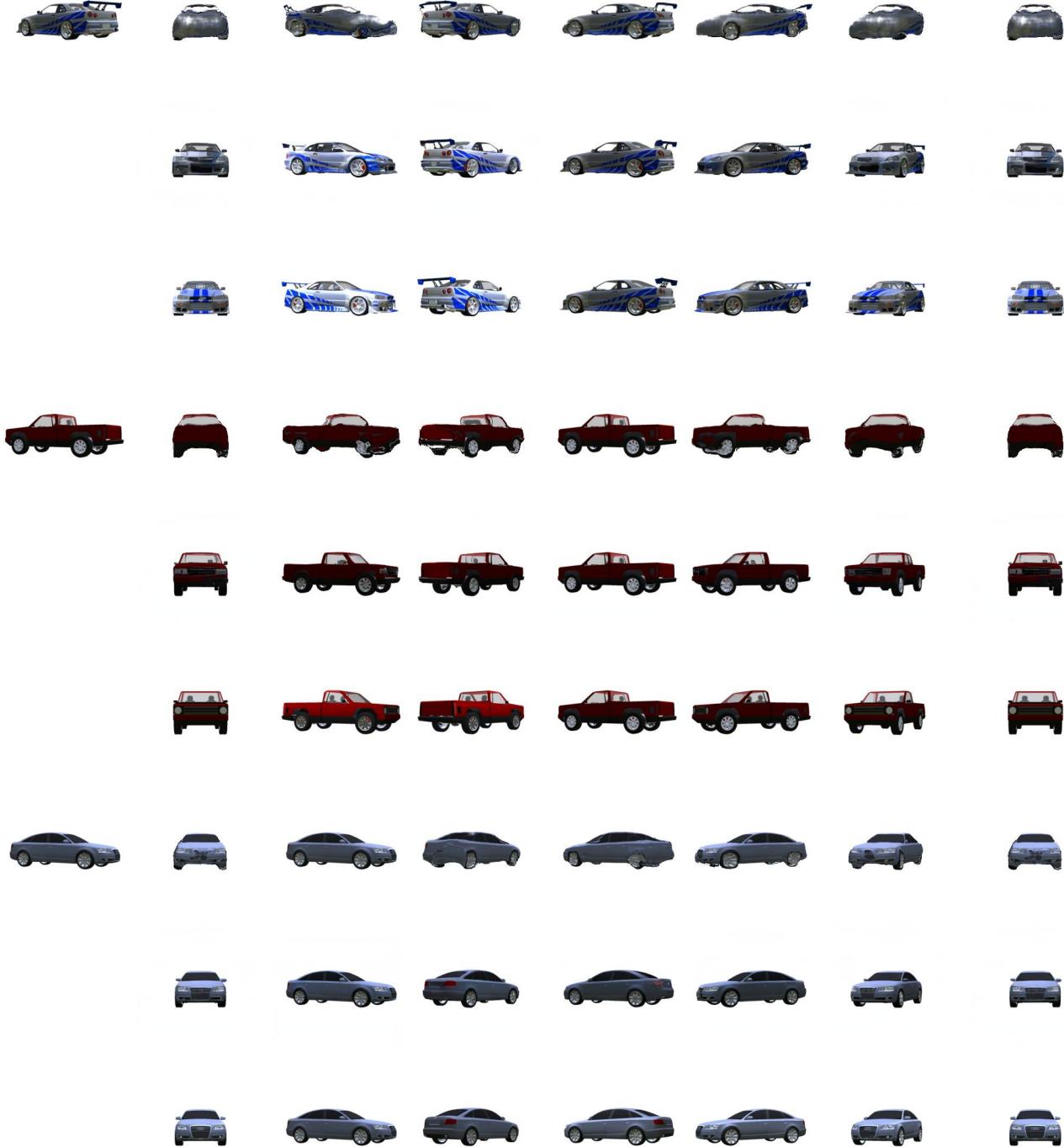


Figure C.13. Qualitative results of the view synthesis task. For each group, we show the results of Appearance Flow [41], the results of our model, and ground-truth images, respectively. The top left image is the input source image. The other images are the generated results and ground-truth images. *Click on the image to start the animation in a browser.*



Figure C.14. Qualitative results of the view synthesis task. For each group, we show the results of Appearance Flow [41], the results of our model, and ground-truth images, respectively. The top left image is the input source image. The other images are the generated results and ground-truth images. *Click on the image to start the animation in a browser.*

C .3. Additional Results of Image Animation

We provide additional results of the image animation task in Figure C.15.



Figure C.15. Qualitative results of the image animation task. For each row, the leftmost image is the source image. The others are generated images. *Click on the image to start the animation in a browser.*

D . Implementation Details

Basically, the auto-encoder structure is employed to design our network. We use the residual blocks as shown in Figure D.16 to build our model. Each convolutional layer is followed by instance normalization [31]. We use Leaky-ReLU as the activation function in our model. Spectral normalization [21] is employed in the discriminator to solve the notorious problem of instability training of generative adversarial networks. The architecture of our model is shown in Figure D.17. We note that since the images of the Market-1501 dataset are low-resolution images (128×64), we only use one local attention block at the feature maps with resolution as 32×16 . We design the kernel prediction net M in the local attention block as a fully connected network. The extracted local patch $\mathcal{N}_n(\mathbf{f}_s, l + \mathbf{w}^l)$ and $\mathcal{N}_n(\mathbf{f}_t, l)$ are concatenated as the input. The output of the network is \mathbf{k}_l . Since it needs to predict attention kernels \mathbf{k}_l for all location l in the feature maps, we use a convolutional layer to implement this network, which can take advantage of the parallel computing power of GPUs.

We train our model in stages. The Flow Field Estimator is first trained to generate flow fields. Then we train the whole model in an end-to-end manner. We adopt the ADAM optimizer. The learning rate of the generator is set to 10^{-4} . The discriminator is trained with a learning rate of one-tenth of that of the generator. The batch size is set to 8 for all experiments. The loss weights are set to $\lambda_c = 5$, $\lambda_r = 0.0025$, $\lambda_{\ell_1} = 5$, $\lambda_a = 2$, $\lambda_p = 0.5$, and $\lambda_s = 500$.

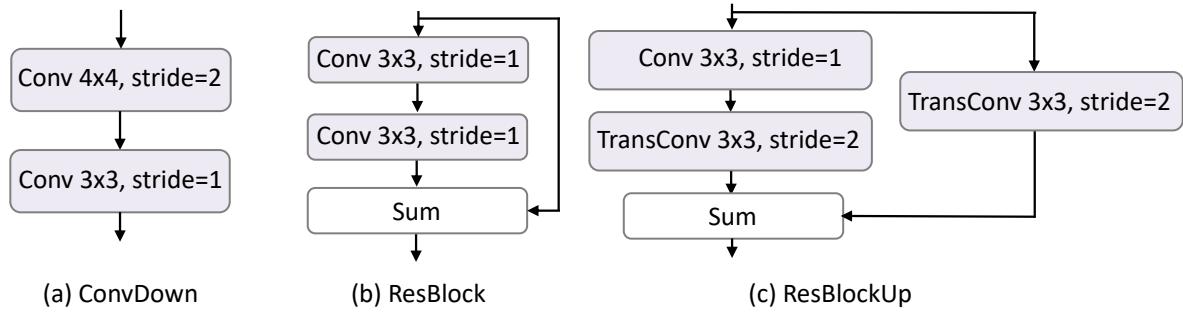


Figure D.16. The components used in our networks.

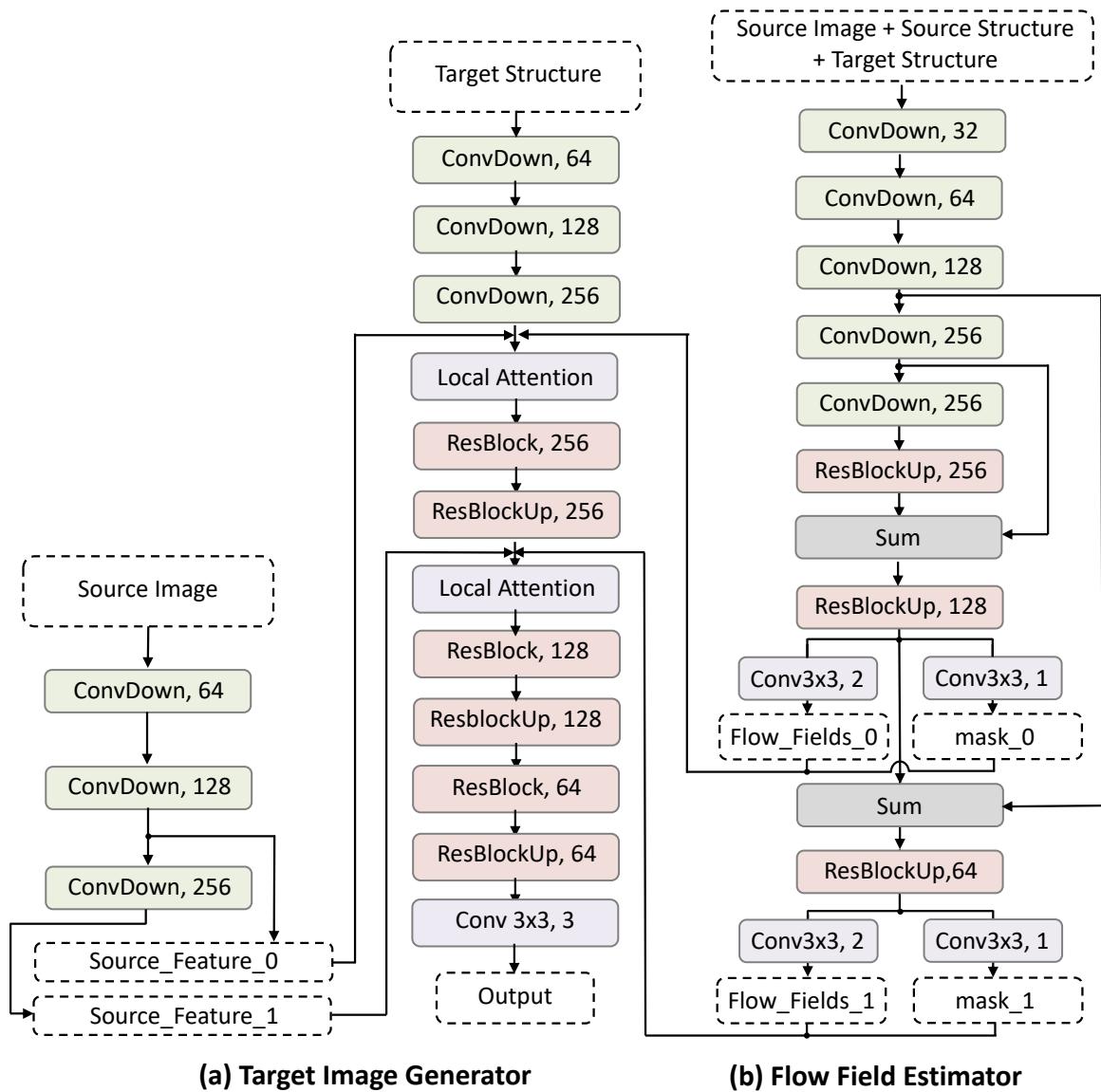


Figure D.17. The network architecture of our network.