Deep Spatial Transformation for Pose-Guided Person Image Generation and Animation

Yurui Ren, Ge Li, Shan Liu, and Thomas H. Li

***Abstract*—Pose-guided person image generation and animation aim to transform a source person image to target poses. These tasks require spatial manipulation 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. This framework first estimates global flow fields between sources and targets. Then, corresponding local source feature patches are sampled with content-aware local attention coefficients. We show that our framework can spatially transform the inputs in an efficient manner. Meanwhile, we further model the temporal consistency for the person image animation task to generate coherent videos. The experiment results of both image generation and animation tasks demonstrate the superiority of our model. Besides, additional results of novel view synthesis and face image animation show that our model is applicable to other tasks requiring spatial transformation. The source code of our project is available at https://github.com/RenYurui/Global-Flow-Local-Attention.**

***Index Terms*—Image Spatial Transformation, Image Animation, Pose-guided Image Generation.**

Fig. 1. Illustration of pose-guided person image generation and animation. We show the person image generation task in the first row. For each image pair, the left image is the generated result of our model, while the right image is the input source image. The arrows indicate the data spatial transformation. The second and third rows contain results of the person image animation task. The leftmost image of each row is the source image and the others are the generated results of our model.

I. Introduction

In this paper, we deal with the conditional generation task where the target images are the spatial deformation versions of the source images. Such deformation can be caused by object motions or viewpoint changes. This task is the core of many image/video generation problems. For example, pose-guided person image generation [1], [2], [3], [4] transforms a person image from a source pose to a target pose while retaining the source appearance details. The corresponding pose-guided image animation task [5], [6], [7], [8] further models the temporal consistency and generates a video from a still source image according to a driving target pose sequence. As illustrated in Figure 1, these tasks can be tackled by reasonably reassembling the input data in the spatial domain.

However, Convolutional Neural Networks (CNNs) lack the ability to spatially transform the input features in a parameter efficient manner. One important property of CNNs is the equivariance to transformation [9], which means that if the input spatially shifts, then the output shifts in the same way. This property can benefit tasks requiring reasoning about images such as segmentation [10], [11], detection [12], *etc*. However, it limits the networks by the lack of ability to deal with the deformable-object generation task which requires spatially rearranging the input data. In order to enable spatial transformation capabilities of CNNs, Spatial Transformer Networks (STN) [13] introduces a Spatial Transformer module to standard neural networks. This module regresses transformation parameters and warps the input features using a global affine transformation. However, the global affine transformation is not sufficient in representing the complex deformations of non-rigid objects.

The attention mechanism [14], [15] is able to transform information beyond local regions. It gives networks the ability to build long-term dependencies by allowing networks to use non-local features. It has emerged as an effective technique for many tasks such as natural language processing [14], image recognition [16], [17], and image generation [15]. 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 position. Therefore, each output feature is only related to a local region of the source features. In other words, 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. It predicts 2D coordinate offsets for

Y. Ren, and G. Li are with School of Electronic and Computer Engineering, Peking University, Shenzhen 518055, China (e-mail: yrren@pku.edu.cn; geli@ece.pku.edu.cn).

S. Liu is with Tencent America, Palo Alto, CA 94301 USA (e-mail: shanl@tencent.com).

T. H. Li is with Advanced Institute of Information Technology, Peking University, HangZhou 311215, China (e-mail: tli@aiit.org.cn).

the target features specifying the sampling source positions. However, networks struggle to find reasonable sampling locations when warping the inputs at the feature level [18], [19]. Possible explanations for this phenomenon are that: (1) The input features and flow fields change simultaneously during the training stage. Their parameter update processes are mutually constrained, which means that the input features cannot obtain reasonable gradients without correct flow fields and vice versa. (2) The commonly used Bilinear sampling method provides poor gradient propagation [20], [21]; In order to obtain meaningful flow fields, some flow-based methods [22], [23] warp input data at the pixel level. However, this operation limits the networks to be unable to generate new content. Meanwhile, large motions are difficult to be extracted due to the requirement of generating full-resolution flow fields [24]. Some methods warp the input at the feature level by pre-calculating the flow fields using additional 3D models [8] or generate dense flow fields from sparse key point representation [7]. However, they do not solve the problems in a straightforward manner, which leads to an insufficient transformation representation capability.

In this paper, we propose a differentiable *Global-Flow Local-Attention (GFLA) framework* to solve the problems. Our framework can enable CNNs to reasonably sample and reassemble source features without using any labeled flow fields. The architecture of our GFLA framework 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 Field Estimator is responsible for extracting the long-term dependencies between sources and targets. It estimates flow fields that assign a local source feature patch for each target position. The Local Neural Texture Renderer uses the extracted flow fields to sample the vivid source neural textures. In order to warp sources at the feature level, we propose several targeted solutions to deal with the analyzed problems. First, a Sampling Correctness loss is proposed to constrain flow fields to sample semantically similar regions. This loss helps with the convergence by providing flow fields with additional gradients that are not related to the input source features. Then, a content-aware sampling method is proposed to avoid the poor gradient propagation of the Bilinear sampling. Experiments show that our framework is able to spatially transform the information in an efficient manner. Ablation studies demonstrate that the proposed improvements are helpful for the convergence.

The image-based pose transformation can be further extended for the pose animation task by coherently rendering an input skeleton video. However, most existing models [7], [8], [6], [25] directly apply image transformation methods for this task and generate each video frame independently. This operation does not take the correlations of adjacent frames into consideration, which leads to temporally inconsistent results. In order to obtain coherent results, we make additional efforts to model the temporal dynamics. We notice that the input skeleton sequences extracted by popular pose estimation models [26], [27] are always inconsistent. Since these models predict result poses in an image-based manner and do not consider the temporal information of videos, obvious noise can be observed in their results. Therefore, we propose a Motion Extraction Network to extract clean skeleton sequences from the corresponding noise data. Meanwhile, we improve our GFLA model to generate video clips in a recurrent manner. It allows our model to explicitly extract the correlations between adjacent frames. Ablation studies show that these methods can efficiently improve the final results.

We compare our model with several state-of-the-art methods over both pose-guided image generation and animation tasks. The subjective and objective experiments demonstrate the superiority of our model. Besides, we show that our model is not limited to generating person images. Additional experiments are conducted over other tasks requiring spatial transformation manipulation including novel view synthesis and face image animation. The results show the versatility of our module. The main contributions of our paper can be summarized as:

- A GFLA model is proposed for deep spatial transformation. Experiments on the pose-guided person image generation task show that our model is able to spatially transform the source neural textures in an efficient manner.
- The temporal consistency is further modeled for the person image animation task. Experiments demonstrate that our simple yet efficient improvements can help the model in generating coherent results.
- We show the versatility of our model. Additional experiments on novel view synthesis and face image animation demonstrate that our model can be flexibly applied to other tasks requiring spatial transformation.

A preliminary version of our work has been presented in [19]. In this journal article, we improve our work from the following aspects: (1) We present more in-depth analyses of our GFLA model including a more extensive ablation study to evaluate the efficacy of the components and a more thorough analysis to explain the model performance. (2) We extent our model to tackle the person image animation task. A Motion Extraction Network is proposed to extract clean skeletons from noise inputs. Meanwhile, a sequential GFLA model is presented to model the correlations of the adjacent frames. (3) We provide comprehensive ablation studies and comparison experiments to evaluate the efficacy of our animation model.

II. Related Work

Pose-guided Person Image Generation. An overview of current monocular state-of-the-art pose-guided person image generation approaches is given in Table I. We discuss these methods in detail here. An early attempt [1] performs this task with a two-stage network. It first generates a coarse image with the target pose and then refines the result in an adversarial way. Esser *et al.* [28] propose to disentangle the appearance and pose of person images. However, they ignore the spatial distribution of the original appearance, which limits the network to generate complex textures. Siarohin *et al.* [2] propose that efficient deformation operations are essential for reconstructing realistic results. They assume that the complex deformation between sources and targets can be well approximated by a set of local affine transformations (*e.g.* arms and legs *etc.*). A deformable skip connection module is introduced in their

		Deformation Type	Flow Field Label	For Specific Subject	Temporal Coherence
Generation	Esser *et al.* [28]	-	-	N	N
	Siarohin *et al.* [2]	Multi-Affine Trans	-	N	N
	Zhu *et al.* [4]	Progressive Attn	-	N	N
	Han *et al.* [29]	Pixel Flow	N	N	N
	Li *et al.* [30]	Feature Flow	Y	N	N
	Ours	Feature Flow	N	N	N
Animation	Wang *et al.* [31]	Pixel Flow	Y	Y	Y
	Chan *et al.* [32]	-	-	Y	Y
	Liu *et al.* [8]	Feature Flow	Y	N	N
	Siarohin *et al.* [6]	Feature Flow	N	N	N
	Wang *et al.* [5]	Pixel Flow	Y	N	Y
	Ours-Animation	Feature Flow	N	N	Y

TABLE I. Comparison of the state-of-the-art pose-guided person image generation and animation methods. Methods are compared from four aspects. What deformation module does the method use; Whether the flow field labels are required for training or inference; Is the model trained for a specific subject; Whether temporal coherence is explicitly enforced during training. More details can be found in Section II

paper to spatially transform image textures. However, the predefined transformation components limit the application of this method. Zhu *et al.* [4] propose a more flexible method by using a progressive attention module. However, information may be lost during multiple transfers, which may result in blurry details. Han *et al.* [29] propose a method using flow-based operation for information transform. However, to ease the optimization, they warp the inputs at the pixel level, which means that further refinement networks are required to fill the holes of occlusion contents. Li *et al.* [30] warp the inputs at the feature level, which enables the network to generate occluded contents. But their method requires additional 3D human models to calculate the ground-truth flow field labels. Our model does not require any supplementary information and obtains accurate flow fields in a self-supervised manner.

Pose-guided Person Image Animation. Taking advantage of the generation capabilities of CNNs, many papers [31], [32], [33] deal with this task based on conditional generative adversarial networks (CGANs). Their key idea is to train a mapping function to generate realistic images by mimicking the distribution of training sets. However, as summarized in Table I, these methods are trained to generate specific subjects *i.e.* new models are required to be trained when animating new content. To deal with this problem, Liu *et al.* [8] propose to extract flow fields from predicted 3D body meshes and use a liquid warping module to transform source features. However, the performance of this model is limited by the accuracy of the 3D human mesh prediction. Paper [25] calculates the dense warp grids according to the UV coordinates extracted by Densepose [34]. Again, it relies on accurate UV coordinates to generate realistic images. Some methods predict the dense flow fields from sparse key points movements. Paper [7] and paper [6] use zeroth-order and first-order Taylor expansions to approximate the complex transformations using a set of sparse trajectories respectively. However, these methods do not explicitly model the video temporal coherence, which may lead to inconsistent movements. Wang *et al.* [5] propose a sequential generator to model the correlations between adjacent frames and generate coherent videos. Combining the advantages of the previous works, our animation model can efficiently transform the source neural textures and generate coherent results.

Sparse Attention in Image Generation. The attention mechanism [14] enables networks to model long-term spatial dependencies. It has emerged as a powerful technique to improve the performance of the image generation tasks [15], [35]. However, the standard dense attention module is computationally inefficient. Meanwhile, the dense connection affects networks to benefit from the image locality [36]. To mitigate these limitations, paper [37] introduces Sparse Transformers which separate the full attention operation across several steps of attention. For each step, only a subset of input positions is attended for calculation. Sparse Transformers attain better performance than dense attention with significantly fewer operations. Daras *et al.* [36] propose that local sparse attention kernels introduced in Sparse Transformers are mainly designed for one-dimensional data. They introduce a new local sparse attention layer that preserves two-dimensional image locality and achieves better performance. Instead of separating the dense attention, some methods [38], [39] achieve sparse attention by controlling the sharpness of the softmax function. Our GFLA model can be seen as a type of sparse attention module, where only the flowed local patches are used for the attention coefficient calculation.

III. Global-Flow Local-Attention for Person Image Generation

For the pose-guided person image generation task, target images are the deformation versions of source images. Therefore, target images can be generated by spatially transforming the source images. In this section, we describe a GFLA model to efficiently warp and reassemble source neural textures. The architecture of our model can be found in Figure 2. It can be divided into two modules: *Global Flow Field Estimator* F and *Local Neural Texture Renderer* G. The Global Flow Field Estimator is responsible for estimating the global motions between sources and targets. Flow fields $\mathbf{w}$ and occlusion masks $\mathbf{m}$ are estimated by this module. 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 subsections. 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.

A. Global Flow Field Estimator

We use the 18-channel heat map that encodes the locations of 18 joints of a human body as the structure guidance. Following the previous works [1], [2], [4], the human body joints are detected by the Human Pose Estimator [40]. 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. The Global Flow Field Estimator F takes $\mathbf{x}_s$, $\mathbf{p}_s$, and $\mathbf{p}_t$ as inputs and generates the flow fields $\mathbf{w}$ and occlusion masks $\mathbf{m}$.

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

原始图+原始pose和目标pose得到光流场和遮挡mask

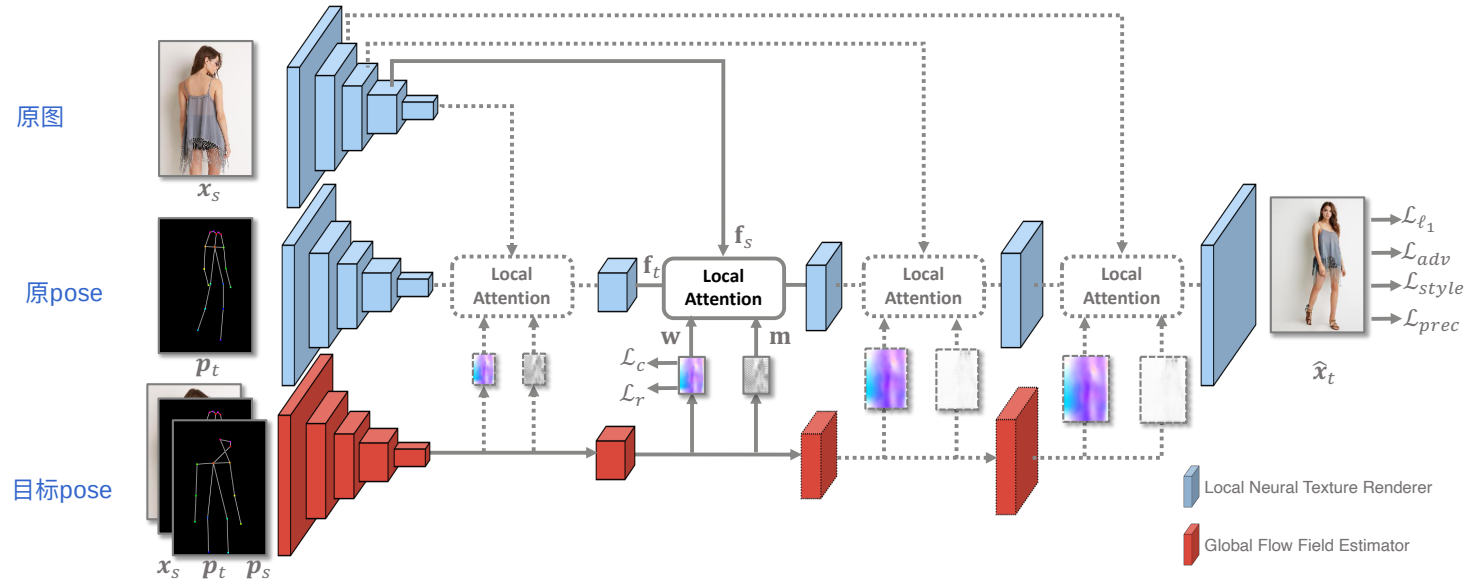


Fig. 2. Overview of our GFLA model. The Global Flow Field Estimator is used to generate flow fields. The Local Neural Texture Renderer yields results by spatially transforming the source features using local attention. Dotted lines indicate that our local attention module can be used at different scales.

where the flow fields $\mathbf{w}$ assign a source patch for each target location. The occlusion masks $\mathbf{m}$ with continuous values between 0 and 1 indicate whether the flowed source patches can be used to generate targets. We design F as a fully convolutional network. $\mathbf{w}$ and $\mathbf{m}$ share all weights of F other than their output layers.

Warping sources at the feature level can help models to be able to generate new content. Meanwhile, it relaxes the requirements of the flow field estimation since the resolutions of the generated flow fields are reduced. However, networks may struggle to find reasonable sampling positions. An important reason is that the gradient propagation of the input features and flow fields are mutually constrained during the warping operation. The input features cannot obtain correct gradients without reasonable flow fields and vice versa. Therefore, we use additional losses to help with the training. We propose a sampling correctness loss to constrain $\mathbf{w}$ in a self-supervised manner. The sampling correctness loss calculates the similarity between the warped source feature and the 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. $\mathbf{v}_{s,\mathbf{w}} = \mathbf{w}(\mathbf{v}_s)$ is the warped results of the source feature $\mathbf{v}_s$ using $\mathbf{w}$. The sampling correctness loss calculates the relative cosine similarity between $\mathbf{v}_{s,\mathbf{w}}$ and $\mathbf{v}_t$.

平移前后余弦相似度越大损失越小

$$\mathcal{L}_c = \frac{1}{N}\sum_{l\in\Omega} exp(-\frac{\mu(\mathbf{v}^l_{s,\mathbf{w}}, \mathbf{v}^l_t)}{\mu^l_{max}}) \tag{2}$$

where $\mu(*)$ denotes the cosine similarity. Coordinate set Ω contains all N positions in the feature maps. $\mathbf{v}^l_{s,\mathbf{w}}$ and $\mathbf{v}^l_t$ denote the features of $\mathbf{v}_{s,\mathbf{w}}$ and $\mathbf{v}_t$ located at the coordinate $l = (x, y)$. The normalization term μ^l_{max} is used to avoid the bias brought by occlusion. It represents the similarity between $\mathbf{v}^l_t$ and its most similar feature in the source feature map $\mathbf{v}_s$. It is calculated as

$$\mu^l_{max} = \max_{l'\in\Omega} \mu(\mathbf{v}^{l'}_s, \mathbf{v}^l_t) \tag{3}$$

where $\mathbf{v}^{l'}_s$ is the feature of $\mathbf{v}_s$ located at the coordinate l'.

Our sampling correctness loss calculates the element-wise similarities. It cannot model the correlation of adjacent features. However, the deformations of image neighborhoods are highly correlated. To model these correlations, we further propose a regularization term. This regularization term is designed 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.

局部仿射变换

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

旋转+平移

where $\mathbf{T}_l = \begin{bmatrix} x_1 & x_2 & ... & x_{n\times n} \\ y_1 & y_2 & ... & y_{n\times n} \end{bmatrix}$ with each coordinate $(x_i, y_i) \in \mathcal{N}_n(\mathbf{c}_t, l)$ and $\mathbf{S}_l = \begin{bmatrix} x_1 & x_2 & ... & x_{n\times n} \\ y_1 & y_2 & ... & y_{n\times n} \\ 1 & 1 & ... & 1 \end{bmatrix}$ 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

最小二乘法

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

where $\mathbf{S}_l^H$ is the transpose matrix of $\mathbf{S}_l$. Our regularization is calculated as the ℓ_2 distance of the error.

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

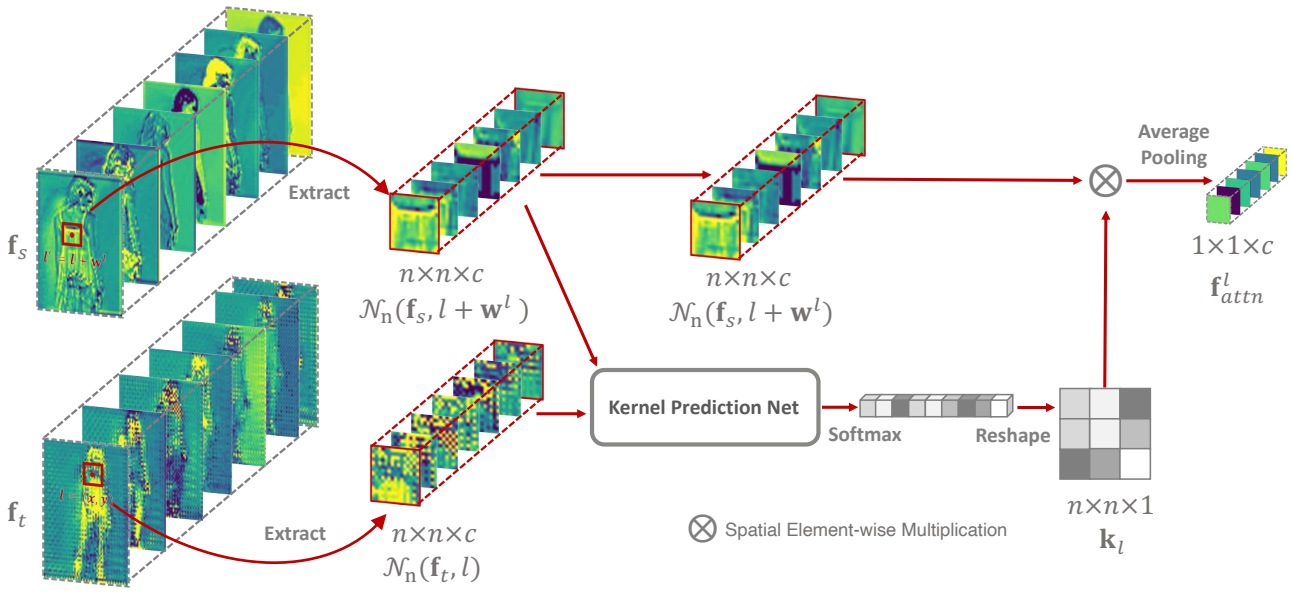


Fig. 3. Overview of our Local Attention. We first extract the feature patch pair from the source and target according to the flow fields. Then the context-aware sampling kernel is calculated by the kernel prediction net. Finally, we sample the source feature and obtain the warped result located at l.

根据KPN生成attention进行加权

B. Local Neural Texture Renderer

The Local Neural Texture Renderer G is responsible for generating the result images by rendering target poses with the source neural textures. It takes $\mathbf{x}_s$, $\mathbf{p}_t$, $\mathbf{w}$, and $\mathbf{m}$ as inputs and generates the result image $\hat{\mathbf{x}}_t$.

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

To avoid the poor gradient propagation of the Bilinear sampling, we propose a local attention operation to sample the source features with a content-aware manner. Our local attention works as a neural renderer where the source neural textures are sampled to render the target bones. We illustrate the processing details in Figure 3. Let $\mathbf{f}_t$ and $\mathbf{f}_s$ represent the extracted features of target bones $\mathbf{p}_t$ and source images $\mathbf{x}_s$ respectively. For each location l, local patches $\mathcal{N}_n(\mathbf{f}_t, l)$ and $\mathcal{N}_n(\mathbf{f}_s, l + \mathbf{w}^l)$ are first extracted from $\mathbf{f}_t$ and $\mathbf{f}_s$.[1] Then, we predict the local $n \times n$ kernel $\mathbf{k}_l$ as the attention coefficients from the extracted local feature patch pair using a kernel prediction network M.

直接cat输入

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

We design M as a fully connected network. The local patch pair $\mathcal{N}_n(\mathbf{f}_s, l + \mathbf{w}^l)$ and $\mathcal{N}_n(\mathbf{f}_t, l)$ are directly concatenated as the network inputs. We use the softmax function as the non-linear activation function of the output layer of model M. This operation forces the sum of $\mathbf{k}_l$ to 1, which enables the stability of gradient backward. Finally, the attention result localed at coordinate $l = (x, y)$ is calculated as

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

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

Furthermore, in order to enable the network to generate occluded contents, we use the mask $\mathbf{m}$ with continuous values between 0 and 1 to select features between the warped result $\mathbf{f}_{attn}$ and the target feature $\mathbf{f}_t$. The final output feature map $\mathbf{f}_{out}$ is calculated as

mask决定遮挡

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

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

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

The generative adversarial loss [41] is used to mimic the distributions of the ground-truth $\mathbf{x}_t$.

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

where D is the discriminator of the Local Neural Texture Renderer G. The perceptual loss and style loss introduced by [42] are used to reduce the reconstruction errors. The perceptual loss calculates ℓ_1 distance between activation maps of a pre-trained network. It can be written as

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

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

$$\mathcal{L}_{style} = \sum_j \left\| G^\phi_j(\mathbf{x}_t) - G^\phi_j(\hat{\mathbf{x}}_t) \right\|_1 \tag{14}$$

[1]The patch $\mathcal{N}_n(\mathbf{f}_s, l + \mathbf{w}^l)$ is extracted using the Bilinear sampling as the coordinates may not be integers.

where G_j^ϕ is the Gram matrix constructed from activation maps ϕ_j. Our GFLA model is trained using the overall loss as

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

IV. Modeling The Temporal Consistency for Person Image Animation

The pose-guided person image animation task refers to generating videos by rendering continuous skeletons using the neural textures of source images. Different from the generation task, it requires not only generating realistic textures for each frame but also modeling the temporal consistency between adjacent frames. Therefore, we further improve our model to generate coherent results. First, a Motion Extraction Network is proposed to extract accurate movements from the noisy input skeletons. Then we improve our GFLA model to generate sequences in a recurrent manner. We describe the details in this section.

A. Motion Extraction Network

One of the major problems is that the input skeleton sequences extracted by the popular algorithms [26], [27] are not temporally consistent. As shown in Figure 8, the predicted locations vibrate around the ground-truth values. Our Motion Extraction Network works as a denoise model extracting accurate movements from noisy skeleton sequences. The architecture of the Motion Extraction Network is shown in Figure 4. Inspired by the paper [43], we design the network using 1D convolutional layers. The input layer of this network takes the concatenated (x, y) coordinates of the N joints for each skeleton frame instead of the 2D heat maps. Let $\mathbf{J}_t^{[1,K]} \in \mathbb{R}^{2N \times K}$ denotes the joints of K input skeletons. The output joints $\hat{\mathbf{J}}_t^{[1,K]}$ contains the coordinates of skeletons with accurate movements. We use Adaptive layer normalization (ADALN) in this network. It has a similar architecture to that of ANAIN [44] but using layer normalization [45] as the normalization function. Layer normalization calculates the statistics for each single training case and normalizes the activities in a batch-wise manner. The effect of this normalization operation can be explained as to removing the unrelated factors such as global locations and scales, thereby making the network focus on motion extraction. As our task is to reconstruct the coherent skeletons, we need to recover the statistics of the original sequences after reasoning about the motions. Therefore, we enable the network to recover the original statistics by calculating the affine parameters of the normalization layers from the input skeletons. The network is trained with ground-truth joints $\mathbf{J}_{gt}^{[1,K]}$. The commonly used mean per-joint position error (MPJPE) is employed as the loss function.

$$\mathcal{L}_{mpjpe} = \left\| \hat{\mathbf{J}}_t^{[1,K]}, \mathbf{J}_{gt}^{[1,K]}, \right\|_1 \quad (16)$$

Since most person animation datasets do not provide the required ground-truth skeleton labels, we train this network separately using the Human3.6M dataset [46]. This dataset contains accurate 3D human skeleton sequences acquired by recording the performance of 11 subjects under 4 different viewpoints. We extract the noise skeleton inputs from the videos of the Human3.6M dataset by using the pose extractor [27]. The ground-truth labels $\mathbf{J}_{gt}^{[1,K]}$ are obtained by projecting the 3D skeletons to the corresponding viewpoints. After training the Motion Extraction Network, we can obtain the clean skeletons $\hat{\mathbf{J}}_t^{[1,K]}$ by performing inference on the animation datasets.

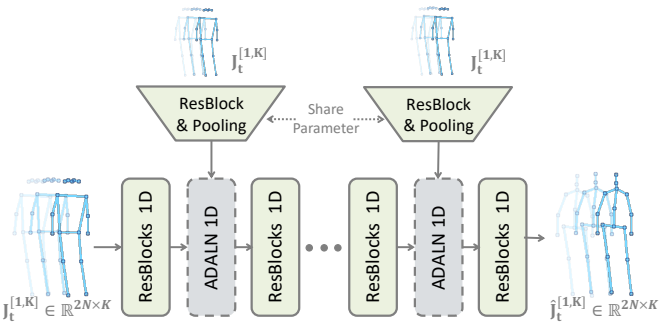


Fig. 4. The architecture of our Motion Extraction Network.

B. Sequential Global-Flow Local-Attention Model

We design a sequential GFLA model to generate result videos from the extracted accurate movements. Let $\hat{\mathbf{p}}_t^{[1,K]} \equiv \{\hat{\mathbf{p}}_t^1, \hat{\mathbf{p}}_t^2, ..., \hat{\mathbf{p}}_t^K\}$ denotes the 2D heat map sequences obtained from the extracted joints $\hat{\mathbf{J}}_t^{[1,K]}$. Our model generates video clips $\hat{\mathbf{x}}_t^{[1,K]} \equiv \{\hat{\mathbf{x}}_t^1, \hat{\mathbf{x}}_t^2, ..., \hat{\mathbf{x}}_t^K\}$ by rendering skeletons $\hat{\mathbf{p}}_t^{[1,K]}$ using the appearance of the source image $\mathbf{x}_s$. We explicitly build the correlations between adjacent frames. Video clips are generated in a recurrent manner: the previously generated frames are used as the inputs of the current generation step. Specifically, Figure 5 shows the generation process of frame $\hat{\mathbf{x}}_t^k$. It can be seen that we have added an additional spatial transformation module responsible for transforming the information of the previously generated frame $\hat{\mathbf{x}}_t^{k-1}$ to the sequential GFLA model. Our model first extracts flow fields $\mathbf{w}_s^k$ and $\mathbf{w}_p^k$ using the Global Flow Field Estimators F_s and F_p respectively.

$$\mathbf{w}_s^k, \mathbf{m}_s^k = F_s(\mathbf{x}_s, \mathbf{p}_s, \hat{\mathbf{p}}_t^k) \quad (17)$$

$$\mathbf{w}_p^k, \mathbf{m}_p^k = F_p(\hat{\mathbf{x}}_t^{k-1}, \hat{\mathbf{p}}_t^{k-1}, \hat{\mathbf{p}}_t^k) \quad (18)$$

where the $\mathbf{m}_s^k$ and $\mathbf{m}_p^k$ are the occlusion masks. The Local Neural Texture Renderer G is then used to generate the result image by spatially transforming the information of $\mathbf{x}_s$ and $\hat{\mathbf{x}}_t^{k-1}$.

$$\hat{\mathbf{x}}_t^k = G(\mathbf{x}_s, \mathbf{p}_s, \mathbf{w}_s^k, \mathbf{m}_s^k, \hat{\mathbf{x}}_t^{k-1}, \hat{\mathbf{p}}_t^{k-1}, \mathbf{w}_p^k, \mathbf{m}_p^k, \hat{\mathbf{p}}_t^k) \quad (19)$$

Two local attention modules are used to warp the features of the source image $\mathbf{x}_s$ and previously generated image $\hat{\mathbf{x}}_t^{k-1}$. The processing operation is the same as that described in Section III-B. The output features $\mathbf{f}_{out\_s}^k$ and $\mathbf{f}_{out\_p}^k$ are generated by these local attention modules. The final output

feature $\mathbf{f}_{out}^k$ is calculated by fusing the outputs of the two branches

$$\mathbf{f}_{out}^k = \mathbf{f}_{out\_s}^k + \mathbf{f}_{out\_p}^k \tag{20}$$

We train the animation model using both spatial and temporal losses. The spatial losses can constrain the model to generate realistic frames. We use the same joint loss (Equation 15) as that of our image generation model for each result frame. The temporal loss is used to model the correlations between different frames. We use a temporal discriminator D_v to calculate this loss. The temporal discriminator D_v takes image sequences as inputs and estimates the probabilities that the inputs are sampled from real video clips.

$$\begin{aligned}\mathcal{L}_{adv\_v} = &\mathbb{E}[\log(1 - D_v(\hat{\mathbf{x}}_t^{[1,K]}))] \\ &+ \mathbb{E}[\log D_v(\mathbf{x}_t^{[1,K]})]\end{aligned} \tag{21}$$

Therefore, the overall loss function of our animation model can be written as

$$\mathcal{L}_A = \frac{1}{K}\sum_{k=1}^{K}\mathcal{L}_G^k + \lambda_v \mathcal{L}_{adv\_v} \tag{22}$$

where $\mathcal{L}_G^k$ represents the spatial loss of frame $\hat{\mathbf{x}}_t^k$.

V. Experiments

In this section, we evaluate the performance of the proposed method. The network structures and training details are first provided in Section V-A. Then, we verify the impact of the proposed modules. The ablation studies are divided into two parts. In Section V-B, we show that our GFLA framework can efficiently spatially transform source feature maps. In Section V-C, we verify the efficacy of our sequential GFLA model. Finally, we compare our method with several state-of-the-art algorithms over both generation and animation tasks in Section V-D.

A. Implementation Details

Network Architecture and Training Details. Auto-encoder structures are employed to design our networks. We use the residual block [47] as the basic component of our model. Unless otherwise specified, we train our models using 256×256 images. We use local attention modules for feature maps with resolutions of 32×32 and 64×64. The extracted local patch sizes are 3 and 5 respectively. For the person image generation task, we train our GFLA 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. For the image animation task, we first train the Motion Extraction Network using the Human3.6M dataset [46] as described in Section IV-A. Then we train our sequential GFLA model using the predicted clean skeletons. We adopt the ADAM optimizer with the learning rate as 10^{-4}.

Metrics. We employ both image-based metrics and video-based metrics to evaluate our results. Learned Perceptual Image Patch Similarity [48] (LPIPS) is used to calculate the reconstruction errors of generated images. This metric computes perceptual distances between input image pairs.

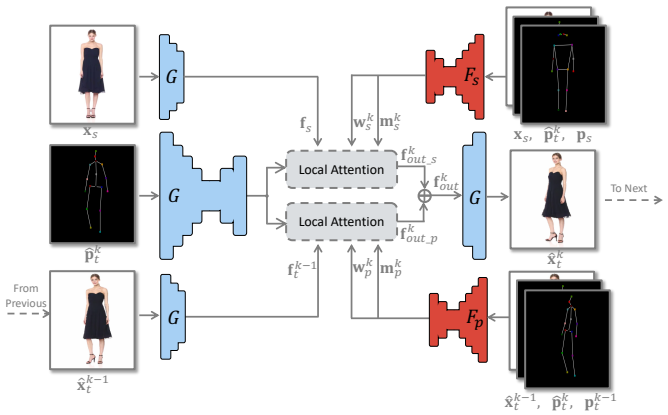


Fig. 5. The generation process of video frame $\hat{\mathbf{x}}_t^k$. Our sequential GFLA model spatially transforms the information of the source image $\mathbf{x}_s$ and previously generated image $\hat{\mathbf{x}}_t^{k-1}$ to generate the result.

Meanwhile, we employ Fréchet Inception Distance [49] (FID) to measure the realism of the generated images. It calculates the Wasserstein-2 distance between distributions of the generated data and real data. For video results, in order to model the temporal consistency errors, we use the I3D model [50] to extract the video features. Average Euclidean Distance [7] (AED) is used as the perceptual reconstruction error indicator. It calculates the Euclidean distance between features of generated videos and ground-truth videos. FID-Video takes the extracted video features as inputs and evaluates the realism of generated videos. Besides, 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. We provide the fooling rate as the evaluation result.

Datasets. For the person image generation task, we use two public datasets: Market-1501 [51] and DeepFashion In-shop Clothes Retrieval Benchmark [52]. 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 these datasets with the same method as that of [4]. The personal identities of the training and testing sets do not overlap. Two video datasets are used for animation tasks: FashionVideo [25] and iPER [8]. The FashionVideo dataset contains 500 training and 100 test videos, each containing roughly 350 frames. Videos have static viewpoints and clean backgrounds. The iPER dataset contains 206 high-resolution videos. Human subjects in this dataset have different conditions of shape, height, and gender.

B. Efficacy of the GFLA Framework

We evaluate the components of our GFLA framework by comparing our model with the following variants.

Baseline. An auto-encoder convolutional network is used as the Baseline model. We do not use any attention blocks in this model. Source images $\mathbf{x}_s$ and guidance poses $\mathbf{p}_t$, $\mathbf{p}_s$ are directly concatenated as the model inputs.

	Flow-Based Method	Content-aware Sampling	FID	LPIPS
Baseline	N	-	16.008	0.2473
Global-Attn	N	-	18.616	0.2575
Local-Attn	Y	Y	12.943	**0.2339**
Bi-Sample	Y	N	12.143	0.2406
Full Model	Y	Y	**10.573**	0.2341

TABLE II. The ablation study results of our GFLA model.

Global Attention Model (Global-Attn). The Global-Attn model is used to compare global attention with our local attention. This model has a similar architecture to the Local Neural Texture Renderer G in Section III-B. The local attention modules 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 $\mathbf{f}_t$.

Local Attention Model (Local-Attn). The Local-Attn model is used to evaluate the efficacy of our sampling correctness loss and regularization loss described in Section III-A. We use the same network architecture as the GFLA model. However, the sampling correctness loss $\mathcal{L}_c$ and regularization loss $\mathcal{L}_r$ are not employed for training.

Bilinear Sampling Model (Bi-Sample). The Bi-Sample model is used to evaluate the efficacy of our local-attention module described in Section III-B. We use both Global Flow Field Estimator F and Local Neural Texture Renderer G in this model. However, the local-attention module is replaced by the Bilinear sampling method.

Full Generation Model (Ours). The proposed GFLA framework is used in this model.

The evaluation results of the ablation study are shown in Table II. Compared with the Baseline, the performance of the Global-Attn model is degraded. It means that the global attention cannot efficiently transform the source information in this task. Flow-based models (Local-Attn, Bi-Sample, and our Full model) improve the generation results, since they force the attention coefficient matrix to be a sparse matrix. The Local-Attn model achieves a good LPIPS result. However, the poor FID score indicates that the realism of its results is degraded since it cannot find reasonable sampling positions for target outputs. The Bi-Sample model is able to obtain relatively accurate flow fields. However, the pre-defined sampling method with limited receptive fields leads to performance degradation. Our full model improves the performance by using the content-aware sampling operation with adjustable receptive fields.

The subjective comparison of these ablation models can be found in Figure 6. The Baseline and Global-Attn model are able to generate images with correct poses. However, the source appearances 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. However, the global features only characterize the global style of the sources, regardless of spatial information. Thus, it causes the vivid local texture details "wash away" in the ultimate image. The flow-based methods spatially transform the features. They are able to reconstruct image details. The Local-Attn model generates textures with similar styles of that of sources. However,

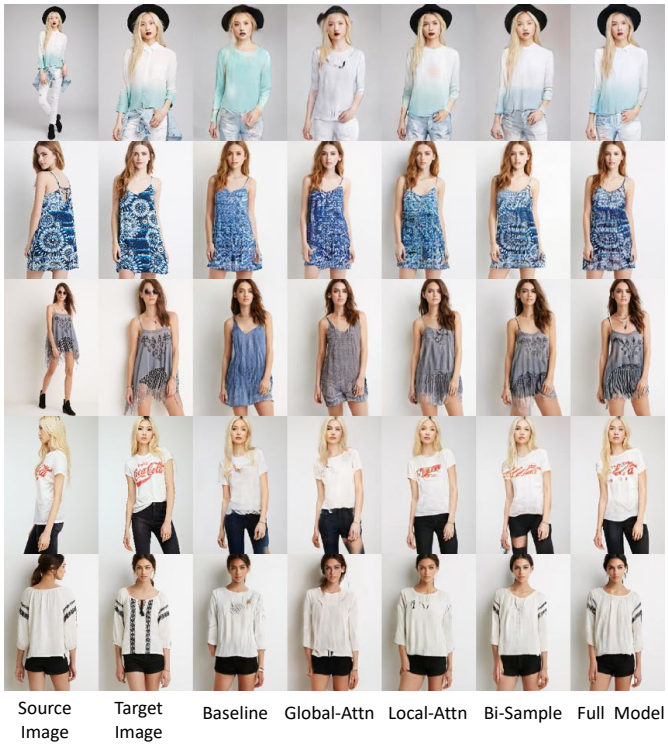


Fig. 6. Qualitative results of our GFLA model and its variants.

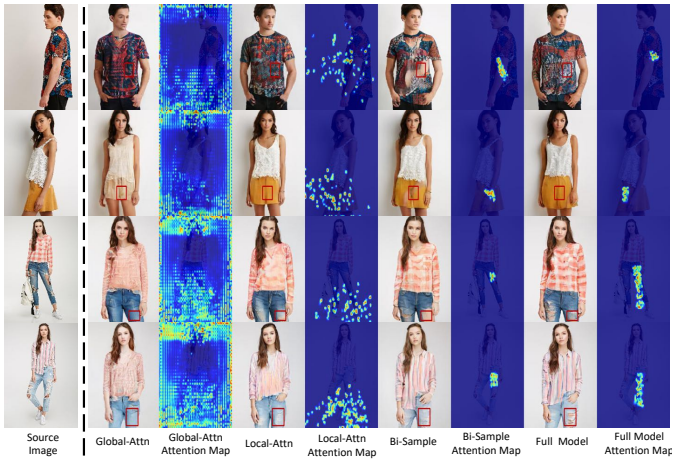


Fig. 7. The visualization results of different attention modules. The red rectangles in the generated images indicate the query locations. The heat maps show the visualization of the corresponding attention coefficients of these query locations. Blue represents low weights.

specific texture patterns (*e.g.* logos) are not reconstructed, since it cannot extract accurate movements between sources and targets. The Bi-Sample model can generate vivid textures. However, artifacts can be observed in its results. Our full model is able to generate realistic images. We further provide the visualization of the attention coefficients in Figure 7. For each attention module, we provide the generated target images and the corresponding attention coefficient heat maps. In order to visualize the attention coefficients, we first calculate attention maps of all query locations in the red rectangle. Then the visualization heat map is calculated by summing the

	DeepFashion			Market-1501				Number of
	FID	LPIPS	JND	FID	LPIPS	Mask-LPIPS	JND	Parameters
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 III. Quantitative comparisons over dataset DeepFashion [52] and Market-1501 [51] with state-of-the-art person image generation methods including Def-GAN [2], VU-Net [28], Pose-Attn [4], and Intr-Flow [30]. FID [49] and LPIPS [48] are objective metrics. JND is obtained by human subjective studies.

obtained attention maps. It can be seen that the Global-Attn model struggles to exclude irrelevant information. Therefore, it is hard to reconstruct accurate textures using the attention results. The Local-Attn model casts a wide net to sample textures for a local target patch. It seems that this model tries to sample all possible positions and omitted the irrelevant information using occlusion masks. However, texture patterns are destroyed during this operation. The Bi-Sample model is able to sample local regions. However, incorrect regions are often sampled due to the poor gradient propagation. Our Full model using the content-aware sampling method can flexibly change the sampling weights and avoid artifacts.

C. Efficacy of the temporal-consistency modeling

We prove that our sequential GFLA model and Motion Extraction Network can help with modeling the temporal-consistency. We compare our model with the following variants.

Naive Animation Model (Naive-Animation). We directly use our GFLA model described in Section III as the Naive-Animation model. It is trained to generate a single target image from a source input. In the inference phase, video frames are generated independently.

Sequential Animation Model (Seq-Animation). The Seq-Animation model is used to evaluate the efficacy of our sequential GFLA model. We use the architecture described in Section IV-B for this model. However, the Motion Extraction Network is not used to preprocess the noisy skeleton sequences.

Full Animation Model (Ours-Animation). We use the sequential GFLA model with clean skeleton sequences obtained by the Motion Extraction Network.

The evaluation results are shown in Table IV. It can be seen that obvious performance gain is obtained by using the sequential GFLA model which explicitly models the correlations between adjacent frames. However, the noisy input skeletons still cause inconsistency which leads to performance degradation. Our full model further improves the performance by extracting clean movements from the noisy input sequences. The subjective results are shown in Figure 8. It can be seen that Naive-Animation can generate realistic video frames. However, it struggles to maintain temporal consistency. The Seq-Animation model solves this problem to a certain extent. However, the noisy input sequences still cause incoherent results. By preprocessing the input skeletons using the Motion Extraction Network, our animation model is able to generate coherent videos with vivid textures.

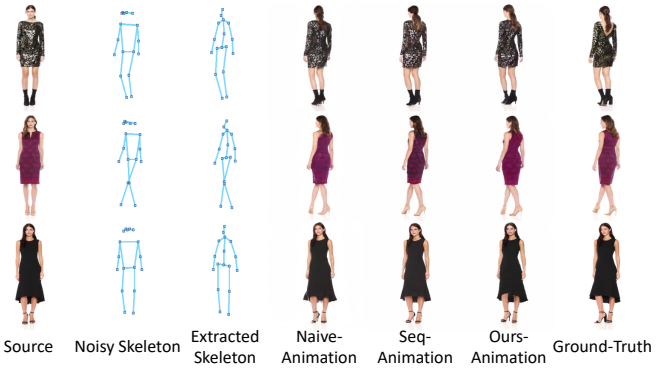


Fig. 8. Qualitative results of our person image animation model and its variants. *Click on the image to play the video in a browser.* The red rectangles in the video highlight the temporal inconsistent clips.

	Sequential Generation	Motion Denoise	FID-Video	AED
Naive-Animation	N	N	4.176	0.0141
Seq-Animation	Y	N	3.685	0.0128
Ours-Animation	Y	Y	**3.426**	**0.0126**

TABLE IV. The ablation study results of our image animation model.

D. Comparisons

In this section, we compare our method to several state-of-the-art models on both generation and animation tasks. For the person image generation task, popular methods Def-GAN [2], VU-Net [28], Pose-Attn[4] and Intr-Flow [30] are selected as the competitors. The quantitative evaluation results are shown in Table III. Please note that we train the Market-1501 dataset using their original 128×64 images. To alleviate the influence of the backgrounds on the reconstruction errors, we follow the previous work [1] to provide the mask-LPIPS. It can be seen that our model achieves competitive evaluation results, which means that our model can generate realistic results with fewer perceptual reconstruction errors. Since subjective metrics have their own limitations, their results may mismatch with the actual subjective perceptions [48]. Therefore, a human objective evaluation test is performed. A JND test is implemented on Amazon Mechanical Turk (MTurk). Volunteers are asked to choose the more realistic image from image pairs of real and generated images. The test is performed over 800 image pairs for each model and dataset. To avoid individual bias, each image pair is compared 5 times by different volunteers. The results can be found in Table III. It can be seen that our model achieves the best result in the challenging Fashion dataset and competitive results in the Market-1501 dataset. Besides, we provide the numbers of model parameters to

Fig. 9. Qualitative comparisons with several state-of-the-art person image generation models including Def-GAN [2], VU-Net [28], Pose-Attn[4], and Intr-Flow [30]. The left part shows the results of the DeepFashion dataset. The right part shows the results of the Market-1501 dataset.

evaluate the computation complexity. Thanks to our efficient spatial transformation 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.

We provide the typical results of different methods in Figure 9. For the Fashion dataset, VU-Net and Pose-Attn struggle to generate complex textures since these models lack efficient spatial transformation blocks. Def-GAN generates correct appearances by transforming neural textures with pre-defined local affine transformation components (eg. arms and legs *etc*.). However, the affine transformation sets are not sufficient to represent the complex spatial variance, which limits the model performance. The flow-based model Intr-Flow is able to generate vivid textures for front pose images. However, it fails to generate realistic results for side pose images. The possible explanation is that this model requires predicting 3D human meshes from 2D images to generate the training flow field labels. Its performance is vulnerable to 3D meshes estimation errors. Our model does not require supplementary labels and obtains accurate flow fields in a self-supervised manner. Thanks to our efficient deep spatial transformation module, we can well preserve the complex textures of the source images. It can be seen that our model generates realistic results with not only correct global patterns but also the vivid details such as the lace of clothes and the shoelace. For the Market-1501 Dataset, artifacts are observed in the results of competitors, such as the sharp edges in Pose-Attn and the halo effects in Def-GAN. Our model is able to generate realistic images. However, it is worth noting that our model does not achieve significant advantages over the competitors. The main reason is that the low-resolution images in this dataset do not contain complex textures, which prevents our advantages from being fully utilized.

For the pose-guided animation task, we compare our model with FewShot-V2V [5] and LiquidNet [8]. The comparison

	FashionVideo				iPER				Number of Parameters
	FID	LPIPS	FID-Video	AED	FID	LPIPS	FID-Video	AED	
LiquidNet	17.681	0.0897	5.174	0.0184	29.97	0.1096	9.212	0.0251	97.45M
FewShot-V2V	27.803	0.0816	5.096	0.0188	75.42	0.2524	8.213	0.0232	97.96M
Ours-Animation	**14.95**	**0.0651**	**3.426**	**0.0126**	**20.53**	**0.0735**	**4.616**	**0.0183**	**23.51M**

TABLE V. Quantitative comparisons over dataset FashionVideo [25] and iPER [8] with state-of-the-art person image animation methods including LiquidNet [8] and FewShot-V2V [5]. FID and LPIPS are image-based metrics. Their results indicate the quality of video frames. FID-Video and AED are calculated using video features. These metrics take the temporal distortions into consideration.

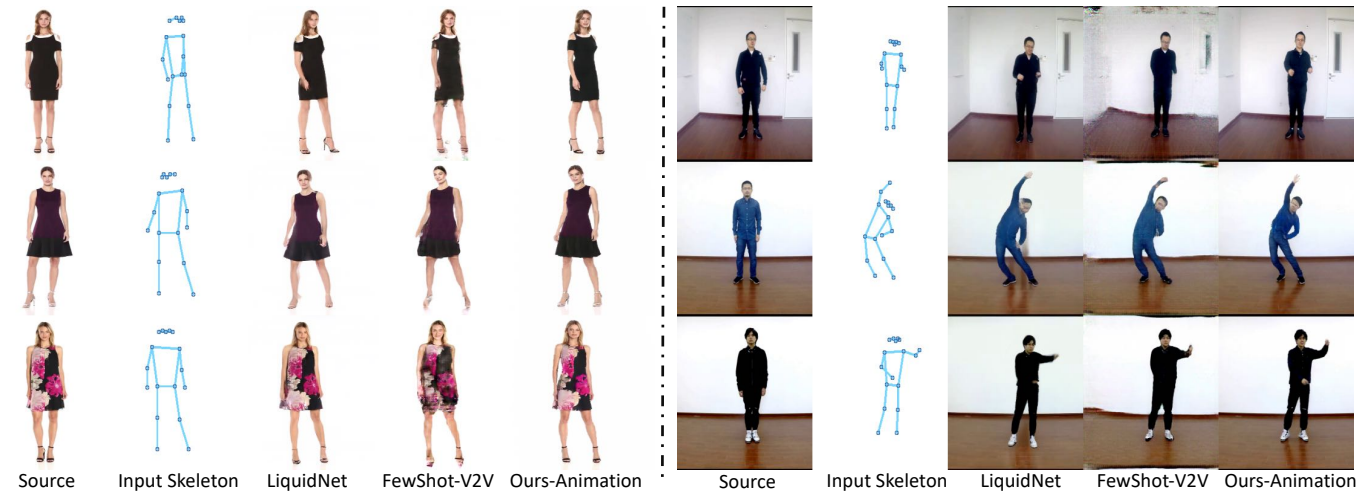


Fig. 10. Qualitative comparisons with person image animation models LiquidNet [8] and FewShot-V2V [5]. The left part contains the results of the FashionVideo dataset. The right part contains the results of the iPER dataset. *Click on the image to play the video in a browser.*

Fig. 11. We show more animation results of our model. The first and third rows contain the source images. The second and fourth rows contain the driven skeleton sequences and the generated results. *Click on the image to play the video in a browser.*

results are shown in Table V. Different from the competitors which employ either face refine models or background inpainting models to improve their results, we do not use any post-processing methods. It can be seen that our model achieves the best results on both datasets. LiquidNet achieves good FID and LPIPS scores, which means that it can generate realistic video frames. However, the relatively poor FID-Video and AED scores indicate that the temporal consistency is not well-maintained. Although FewShot-V2V achieve good results on the video-based metrics, it may suffer from some image-based artifacts, which leads to poor FID scores. Our model can generate coherent results with realistic frames. Meanwhile, we use significantly fewer model weights than competitors.

The subjective results are provided in Figure 10. It can be seen that the LiquidNet model struggles to maintain temporal consistency. This is because this model generates each frame independently. The FewShot-V2V model solves this problem by modeling the correlations between adjacent frames. Although this model can generate coherent results, it suffers from artifacts when generating images with complex textures or backgrounds. Our sequential GFLA model efficiently builds temporal dynamics. Meanwhile, the accurate neural texture transformation module helps with preserving the realistic details. Therefore, our model can generate results with not only correct textures but also vivid temporal details such as the folds of clothes and the movements of hemlines. We provide more results of our model in Figure 11. It can be seen that our model is able to generate realistic videos even for source images with complex textures.

VI. Application on Other Tasks

In this section, we show that our model is not limited to generating person images. It can be flexibly applied to other tasks requiring spatial transformation. Additional experiments

are shown on two typical example tasks: novel view synthesis and face image animation.

Novel view synthesis requires generating new images of an object observed from arbitrary viewpoints. It can be solved by spatially transforming the source information. The car and chair categories of the ShapeNet dataset [53] are used in this experiment. We train the GFLA model described in Section III. The results can be found in Figure 12. We provide the results of appearance flow [22] which warps the source images at the pixel level as a comparison. It can be seen that appearance flow is able to transform the contents in the source images. However, it struggles to reconstruct the occluded details. Our model generates realistic images.

Face image animation is to generate a coherent face video clip according to a source image and a driven structure sequence. Similar to the person image animation task, this task also requires spatial manipulation of source data. We employ the real videos in the FaceForensics dataset [54] This dataset contains 1000 videos of news briefings from different reporters. We follow the previous papers [31], [5] to use the edge maps as the structure guidance. Our sequential generator described in Section IV-B is employed to tackle this task. We show the qualitative results in Figure 13. It can be seen that our model can generate temporally consistent results with realistic textures.

VII. Conclusion and Future Work

In this paper, we tackle the person image generation and animation tasks using deep spatial transformation. We analyze the possible reasons causing poor gradient propagation when warping sources at the feature level. Targeted solution GFLA framework is proposed to first estimate flow fields between sources and targets and then sample the source features in a content-aware manner. We have demonstrated empirically that the GFLA model can provide improved gradients, leading to accurate spatial transformations. Meanwhile, we further propose a sequential GFLA model to extract the correlations between adjacent frames for the animation task. Experiments show that our model can efficiently build temporal dynamics and generate coherent videos. Finally, we demonstrate that our model is versatile on other tasks requiring spatial transformation such as face image animation and novel view synthesis.

Although our model generates impressive results, we also observe some failure cases as shown in Figure 14. These typical failure cases are due to the severe occlusions of source images, which misleads the model to sample incorrect neural textures. We provide possible solutions for this open issue to inspire future works in this problem. One way is to add additional constraints to flow fields. For example, loss functions can be designed to penalize sampling between different semantic regions. Another solution is to perform multi-step warping operations to gradually warp source images to targets by using additional video datasets.

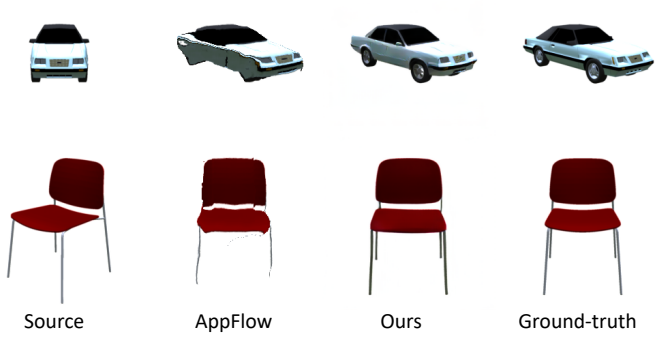


Fig. 12. Qualitative results of the view synthesis task. We show the results of our model and appearance flow [22] (AppFlow) model. *Click on the image to play the video in a browser.*

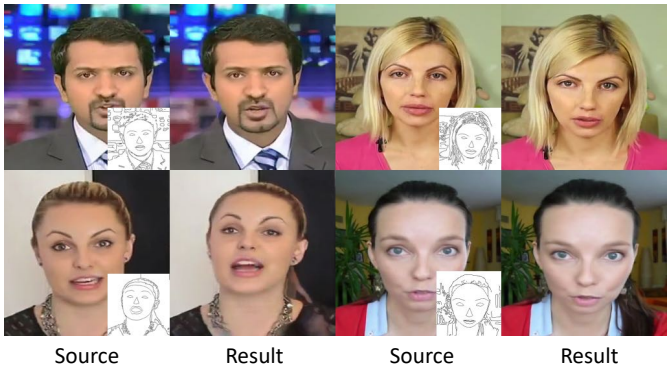


Fig. 13. Qualitative results of the face image animation task. *Click on the image to play the video in a browser.*

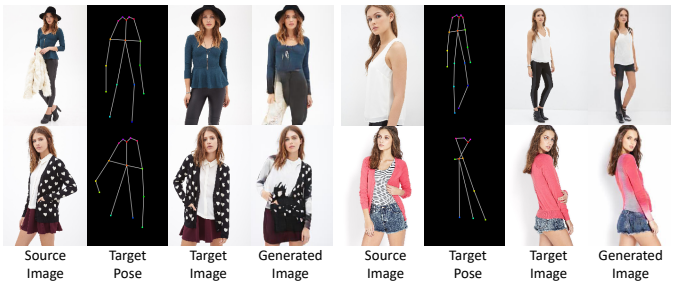


Fig. 14. Some failure cases of our GFLA model.

References

[1] L. Ma, X. Jia, Q. Sun, B. Schiele, T. Tuytelaars, and L. Van Gool, "Pose guided person image generation," in *Advances in Neural Information Processing Systems*, 2017, pp. 406–416. 1, 2, 3, 9

[2] A. Siarohin, E. Sangineto, S. Lathuilière, and N. Sebe, "Deformable gans for pose-based human image generation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3408–3416. 1, 2, 3, 9, 10, 15

[3] S. Song, W. Zhang, J. Liu, and T. Mei, "Unsupervised person image generation with semantic parsing transformation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2357–2366. 1

[4] Z. Zhu, T. Huang, B. Shi, M. Yu, B. Wang, and X. Bai, "Progressive pose attention transfer for person image generation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2347–2356. 1, 3, 7, 9, 10, 15

[5] T.-C. Wang, M.-Y. Liu, A. Tao, G. Liu, J. Kautz, and B. Catanzaro, "Few-shot video-to-video synthesis," *arXiv preprint arXiv:1910.12713*, 2019. 1, 3, 10, 11, 12, 16

[6] A. Siarohin, S. Lathuilière, S. Tulyakov, E. Ricci, and N. Sebe, "First order motion model for image animation," in *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., 2019, pp. 7137–7147. 1, 2, 3

[7] A. Siarohin, S. Lathuilire, S. Tulyakov, E. Ricci, and N. Sebe, "Animat-

ing arbitrary objects via deep motion transfer," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 1, 2, 3, 7

[8] W. Liu, Z. Piao, J. Min, W. Luo, L. Ma, and S. 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*, 2019, pp. 5904–5913. 1, 2, 3, 7, 10, 11, 16, 19

[9] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016. 1

[10] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580–587. 1

[11] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969. 1

[12] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in *Advances in neural information processing systems*, 2014, pp. 568–576. 1

[13] M. Jaderberg, K. Simonyan, A. Zisserman *et al.*, "Spatial transformer networks," in *Advances in neural information processing systems*, 2015, pp. 2017–2025. 1

[14] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008. 1, 3

[15] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena, "Self-attention generative adversarial networks," *arXiv preprint arXiv:1805.08318*, 2018. 1, 3

[16] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7794–7803. 1

[17] H. Hu, Z. Zhang, Z. Xie, and S. Lin, "Local relation networks for image recognition," *arXiv preprint arXiv:1904.11491*, 2019. 1

[18] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. S. Huang, "Generative image inpainting with contextual attention," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5505–5514. 2

[19] Y. Ren, X. Yu, J. Chen, T. H. Li, and G. Li, "Deep image spatial transformation for person image generation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 7690–7699. 2

[20] W. Jiang, W. Sun, A. Tagliasacchi, E. Trulls, and K. M. Yi, "Linearized multi-sampling for differentiable image transformation," *Proceedings of the IEEE International Conference on Computer Vision*, 2019. 2

[21] Y. Ren, X. Yu, R. Zhang, T. H. Li, S. Liu, and G. Li, "Structureflow: Image inpainting via structure-aware appearance flow," in *IEEE International Conference on Computer Vision (ICCV)*, 2019. 2

[22] T. Zhou, S. Tulsiani, W. Sun, J. Malik, and A. A. Efros, "View synthesis by appearance flow," in *European conference on computer vision*. Springer, 2016, pp. 286–301. 2, 12, 17, 18

[23] P. Fischer, A. Dosovitskiy, E. Ilg, P. Häusser, C. Hazırbaş, V. Golkov, P. Van der Smagt, D. Cremers, and T. Brox, "Flownet: Learning optical flow with convolutional networks," *arXiv preprint arXiv:1504.06852*, 2015. 2

[24] A. Ranjan and M. J. Black, "Optical flow estimation using a spatial pyramid network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 4161–4170. 2

[25] P. Zablotskaia, A. Siarohin, B. Zhao, and L. Sigal, "Dwnet: Dense warp-based network for pose-guided human video generation," *arXiv preprint arXiv:1910.09139*, 2019. 2, 3, 7, 11

[26] Z. Cao, G. Hidalgo, T. Simon, S.-E. Wei, and Y. Sheikh, "Openpose: realtime multi-person 2d pose estimation using part affinity fields," *arXiv preprint arXiv:1812.08008*, 2018. 2, 6

[27] J. Li, C. Wang, H. Zhu, Y. Mao, H.-S. Fang, and C. Lu, "Crowdpose: Efficient crowded scenes pose estimation and a new benchmark," *arXiv preprint arXiv:1812.00324*, 2018. 2, 6, 19

[28] P. Esser, E. Sutter, and B. Ommer, "A variational u-net for conditional appearance and shape generation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8857–8866. 2, 3, 9, 10, 15

[29] X. Han, X. Hu, W. Huang, and M. R. Scott, "Clothflow: A flow-based model for clothed person generation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 10 471–10 480. 3

[30] Y. Li, C. Huang, and C. C. Loy, "Dense intrinsic appearance flow for human pose transfer," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3693–3702. 3, 9, 10, 15

[31] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, G. Liu, A. Tao, J. Kautz, and B. Catanzaro, "Video-to-video synthesis," *arXiv preprint arXiv:1808.06601*, 2018. 3, 12

[32] C. Chan, S. Ginosar, T. Zhou, and A. A. Efros, "Everybody dance now," *arXiv preprint arXiv:1808.07371*, 2018. 3

[33] K. Aberman, M. Shi, J. Liao, D. Lischinski, B. Chen, and D. Cohen-Or, "Deep video-based performance cloning," in *Computer Graphics Forum*, vol. 38, no. 2. Wiley Online Library, 2019, pp. 219–233. 3

[34] R. Alp Güler, N. Neverova, and I. Kokkinos, "Densepose: Dense human pose estimation in the wild," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7297–7306. 3, 19

[35] A. Brock, J. Donahue, and K. Simonyan, "Large scale gan training for high fidelity natural image synthesis," *arXiv preprint arXiv:1809.11096*, 2018. 3

[36] G. Daras, A. Odena, H. Zhang, and A. G. Dimakis, "Your local gan: Designing two dimensional local attention mechanisms for generative models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 14 531–14 539. 3

[37] R. Child, S. Gray, A. Radford, and I. Sutskever, "Generating long sequences with sparse transformers," *arXiv preprint arXiv:1904.10509*, 2019. 3

[38] P. Zhang, B. Zhang, D. Chen, L. Yuan, and F. Wen, "Cross-domain correspondence learning for exemplar-based image translation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 5143–5153. 3

[39] W. Jiang, S. Liu, C. Gao, J. Cao, R. He, J. Feng, and S. Yan, "Psgan: Pose and expression robust spatial-aware gan for customizable makeup transfer," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 5194–5202. 3

[40] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, "Realtime multi-person 2d pose estimation using part affinity fields," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 7291–7299. 3

[41] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680. 5

[42] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *European conference on computer vision*. Springer, 2016, pp. 694–711. 5

[43] D. Pavllo, C. Feichtenhofer, D. Grangier, and M. Auli, "3d human pose estimation in video with temporal convolutions and semi-supervised training," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 7753–7762. 6, 19

[44] X. Huang and S. Belongie, "Arbitrary style transfer in real-time with adaptive instance normalization," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 1501–1510. 6, 19

[45] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," *arXiv preprint arXiv:1607.06450*, 2016. 6

[46] C. Ionescu, D. Papava, V. Olaru, and C. Sminchisescu, "Human3. 6m: Large scale datasets and predictive methods for 3d human sensing in natural environments," *IEEE transactions on pattern analysis and machine intelligence*, vol. 36, no. 7, pp. 1325–1339, 2013. 6, 7

[47] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778. 7

[48] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 586–595. 7, 9

[49] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "Gans trained by a two time-scale update rule converge to a local nash equilibrium," in *Advances in Neural Information Processing Systems*, 2017, pp. 6626–6637. 7, 9

[50] J. Carreira and A. Zisserman, "Quo vadis, action recognition? a new model and the kinetics dataset," in *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 6299–6308. 7

[51] L. Zheng, L. Shen, L. Tian, S. Wang, J. Wang, and Q. Tian, "Scalable person re-identification: A benchmark," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1116–1124. 7, 9, 15

[52] Z. Liu, P. Luo, S. Qiu, X. Wang, and X. Tang, "Deepfashion: Powering robust clothes recognition and retrieval with rich annotations," in

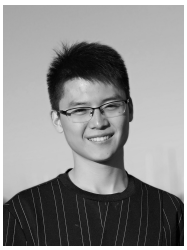
Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 1096–1104. 7, 9, 15

[53] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su *et al.*, "Shapenet: An information-rich 3d model repository," *arXiv preprint arXiv:1512.03012*, 2015. 12

[54] A. Rössler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Nießner, "Faceforensics: A large-scale video dataset for forgery detection in human faces," *arXiv preprint arXiv:1803.09179*, 2018. 12

[55] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Instance normalization: The missing ingredient for fast stylization," *arXiv preprint arXiv:1607.08022*, 2016. 19

[56] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, "Spectral normalization for generative adversarial networks," *arXiv preprint arXiv:1802.05957*, 2018. 19



Yurui Ren received the B.S. degree form School of Electronic Information and Electrical Engineering, Dalian University of Technology, Dalian, China, in 2017. He is currently pursuing the Ph.D. degree in School of Electronic and Computer Engineering, Peking University, Shenzhen, China. His research interests include image generation and image enhancement.

Ge Li is a professor at the School of Electronic and Computer Engineering in Peking University Shenzhen Graduate School, China. His research interests include image/video process and analysis, machine learning, digital communications and signal processing.

Shan Liu received the B.Eng. degree in electronic engineering from Tsinghua University, the M.S. and Ph.D. degrees in electrical engineering from the University of Southern California, respectively.

She is a Tencent Distinguished Scientist and General Manager of Tencent Media Lab. She was formerly Director of Media Technology Division at MediaTek USA. She was also formerly with MERL, Sony and IBM. She has been actively contributing to international standards since the last decade and served co-Editor of HEVC SCC and the emerging VVC. She has numerous technical contributions adopted into various standards, such as HEVC, VVC, OMAF, DASH and PCC, etc. At the same time, technologies and products developed by her and under her leadership have served several hundred million users. Dr. Liu holds more than 150 granted US and global patents and has published more than 80 journal and conference papers. She was in the committee of Industrial Relationship of IEEE Signal Processing Society (2014-2015) and is on the Editorial Board of IEEE Transactions on Circuits and Systems for Video Technology (2018-2021). She was the VP of Industrial Relations and Development of Asia-Pacific Signal and Information Processing Association (2016-2017) and was named APSIPA Industrial Distinguished Leader in 2018. She was appointed Vice Chair of IEEE Data Compression Standards Committee in 2019. Her research interests include audio-visual, high volume, immersive and emerging media compression, intelligence, transport, and systems.

Thomas H. Li received the BE in electronics engineering from Beijing Institute of Technology, Beijing, China, in 1982, and the MSEE and Ph.D. in electrical engineering from Purdue University, West Lafayette, Indiana, USA in 1991 and 1999, respectively. After working in industry for over 20 years, he joined Gpower Semiconductor, Inc., in Suzhou, China, as the Chief Strategist. His research interests include communication, signal processing and machine learning.

A . Additional Results of Person Image Generation

We provide additional comparisons with state-of-the-art person image generation models in this section. The qualitative results is shown in Figure H.15.

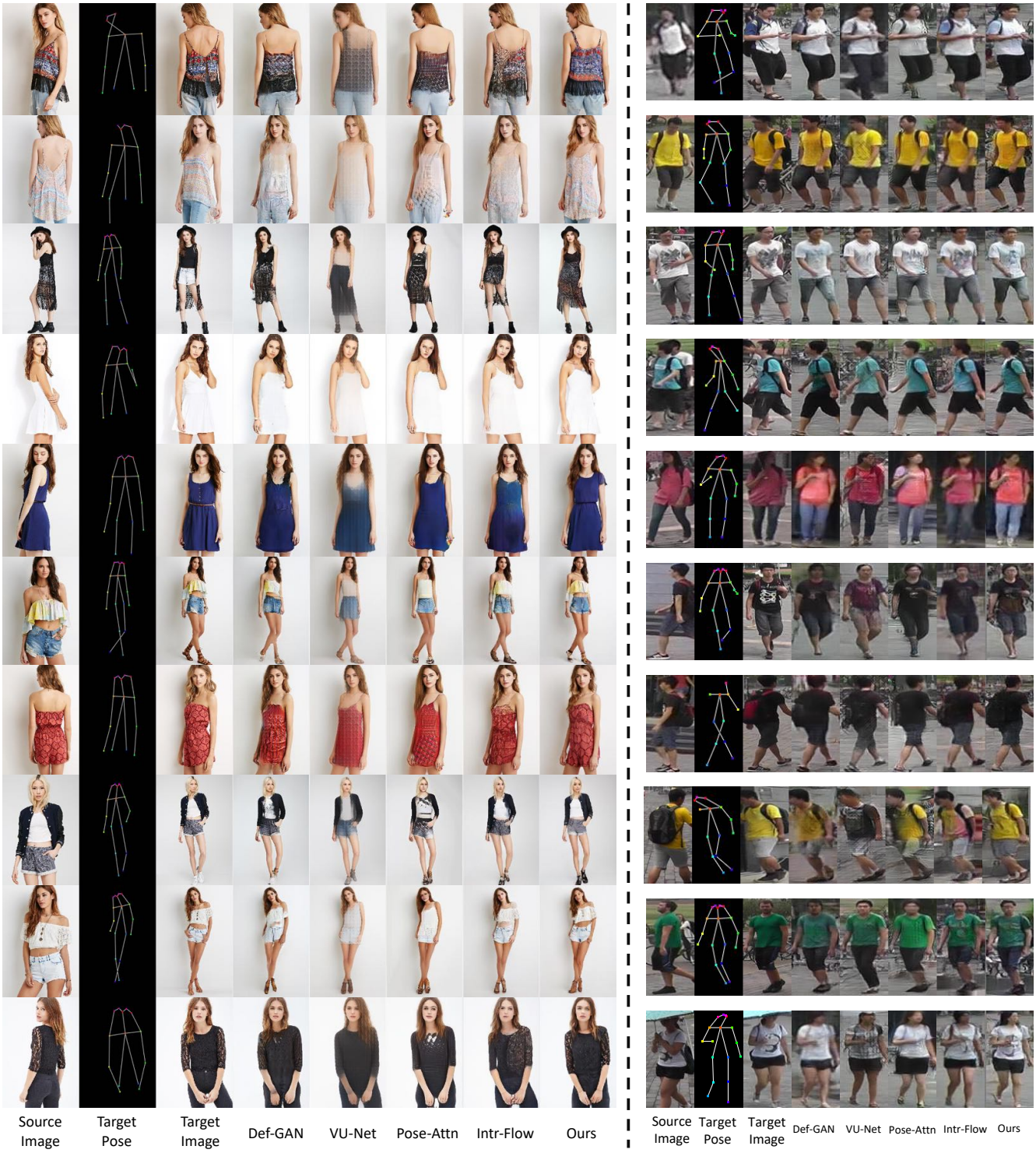


Fig. H.15. The qualitative comparisons with several state-of-the-art models including Def-GAN [2], VU-Net [28], Pose-Attn[4], and Intr-Flow [30] over dataset DeepFashion [52] and Market-1501 [51].

B . Additional Results of Person Image Animation

We provide additional results of the person image animation task in Figure B.16.

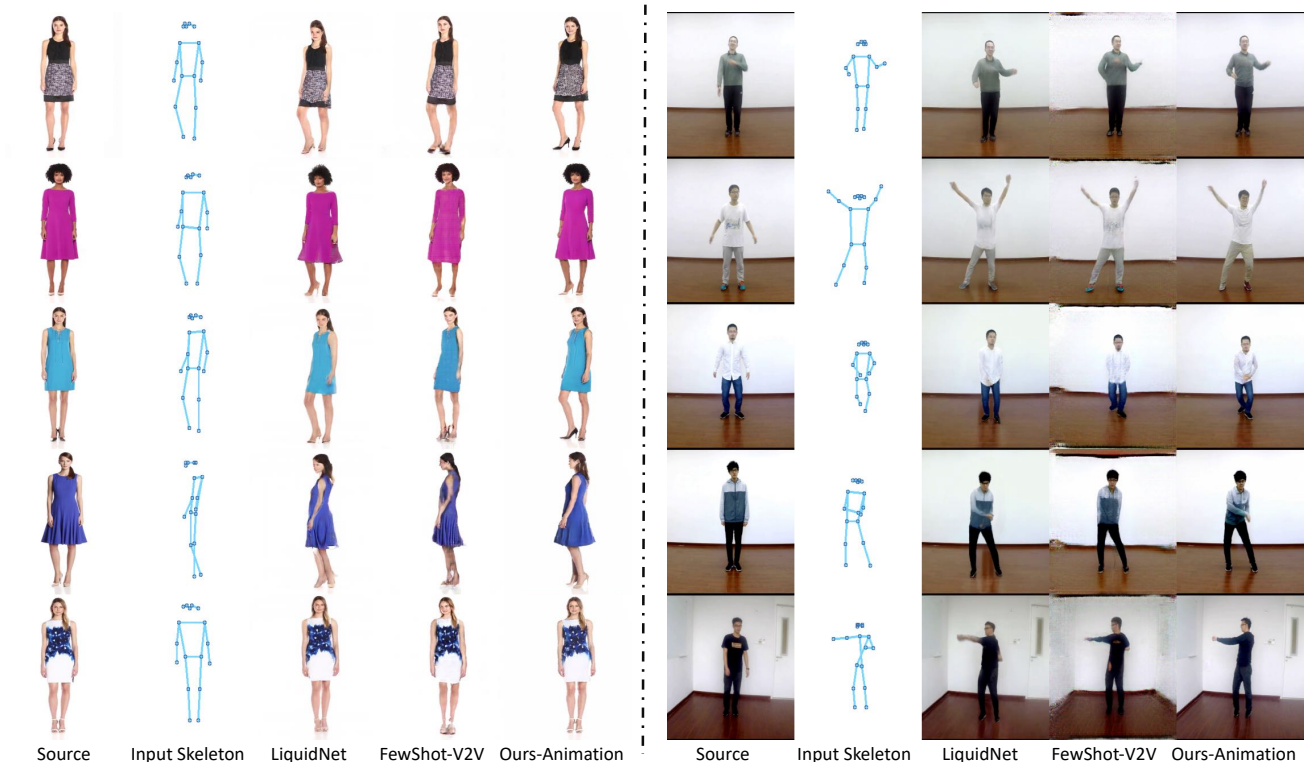


Fig. B.16. Qualitative results of the person image animation task. We compare our model with stare-of-the-art person image animation models including LiquidNet [8] and FewShot-V2V [5]. *Click on the image to start the animation in a browser.*

C . Additional Results of Face Image Animation

We provide additional results of the face image animation task in Figure C.17.

Fig. C.17. 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 . Additional Results of View Synthesis

We provide additional results of the view synthesis task in Figure D.18 and Figure D.19.

Fig. D.18. Qualitative results of the view synthesis task. For each group, we show the results of Appearance Flow [22], 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.*

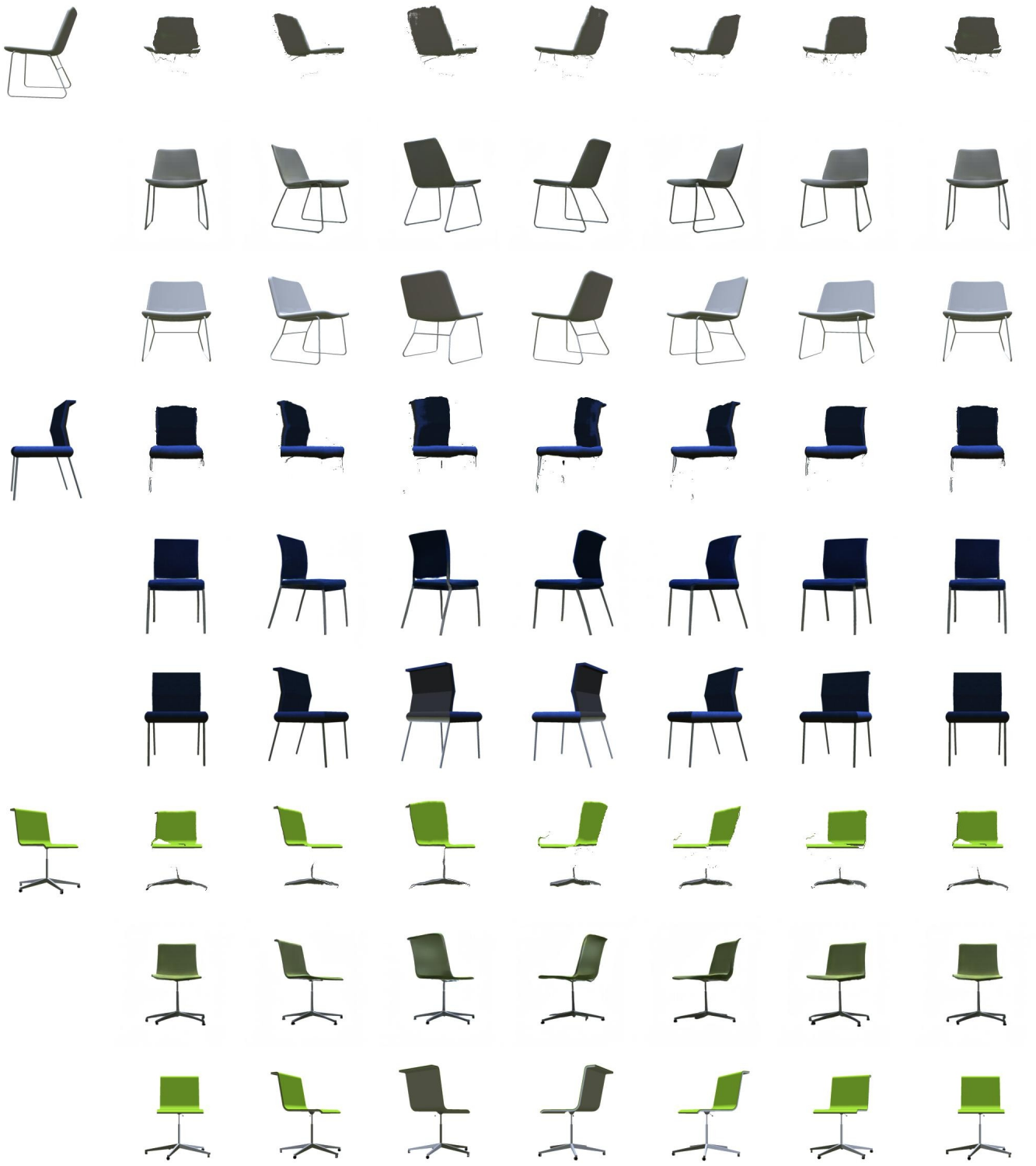


Fig. D.19. Qualitative results of the view synthesis task. For each group, we show the results of Appearance Flow [22], the results of our model, and ground-truth images, respectively. The top left image is the input source image. The other images are the generated results and ground-truth images. *Click on the image to start the animation in a browser.*

E . Implementation Details

Basically, the auto-encoder structure is employed to design our networks. We use the residual blocks as shown in Figure E.20 to build our model. Each convolutional layer is followed by instance normalization [55]. We use Leaky-ReLU as the activation function in our model. Spectral normalization [56] is employed in the discriminator to solve the notorious problem of instability training of generative adversarial networks.

For our GFLA model, we show the architecture in Figure E.21. 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 this 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 $\lambda_s = 500$.

For our person image animation model, we show the architecture of the Motion Extraction Network E.22 and sequential GFLA network E.23. The Motion Extraction Network is designed using a similar structure as that of the paper [43]. We use the 1D convolutional layers as the basic component. The ADALN is used as the normalization layer. Let $\mathbf{f} \in \mathbb{R}^{N \times C \times L}$ denotes the activations of a 1D convolution layer. The ADALN normalize the inputs as

$$ADALN(\mathbf{f}) = \gamma(\frac{\mathbf{f} - \mu(\mathbf{f})}{\sigma(\mathbf{f})}) + \beta \tag{23}$$

where $\mu(\mathbf{f})$ and $\sigma(\mathbf{f})$ are computed across spatial and channel dimensions for each training case

$$\mu_b(\mathbf{f}) = \frac{1}{CL} \sum_{c=1}^{C} \sum_{l=1}^{L} \mathbf{f}_{bcl} \tag{24}$$

$$\sigma_b(\mathbf{f}) = \sqrt{\frac{1}{CL} \sum_{c=1}^{C} \sum_{l=1}^{L} (\mathbf{f}_{bcl} - \mu_b(\mathbf{f}))^2} \tag{25}$$

Instead of learning a single set of affine parameters γ and β, we follow previous methods [44] to calculate them for each training case using the input joints. This operation allows the network to recover the input statistics (*i.e.* locations, scales).

$$\beta, \gamma = E(\mathbf{J}_t^{[1,K]}) \tag{26}$$

where E is the statistic extraction module. As shown in Figure E.23, the sequential GFLA model has a similar architecture with that of the GFLA model. We add another path to transform the information of the previously generated images. We first train the Motion Extraction Network using the Human3.6M dataset. We use the Alphapose model [34] extract the noisy input skeletons. The corresponding ground-truth skeletons are provided by the dataset. After training the Motion Extraction Network, we can perprocess the skeletons of the person image animation datasets FashionVideo [27] and iPER [8]. Finally, we train the sequential GFLA model in an end-to-end manner. For the first frame, we use the source image as the previously generated image. 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 2 for all experiments.

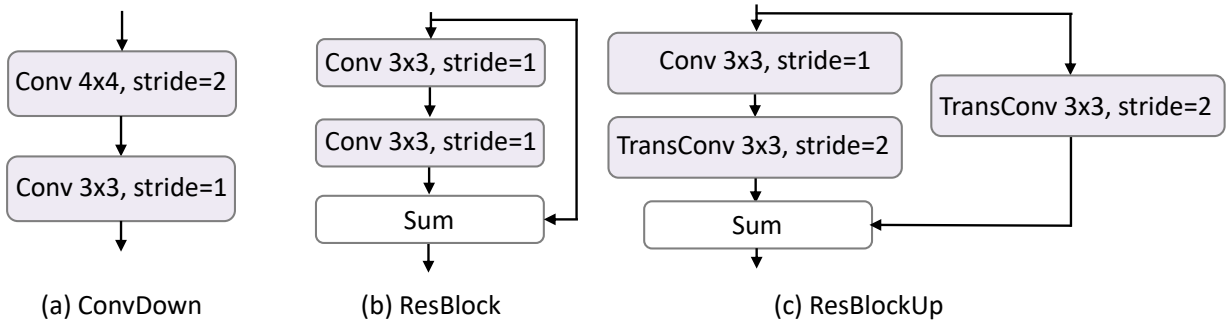


Fig. E.20. The components used in our networks.

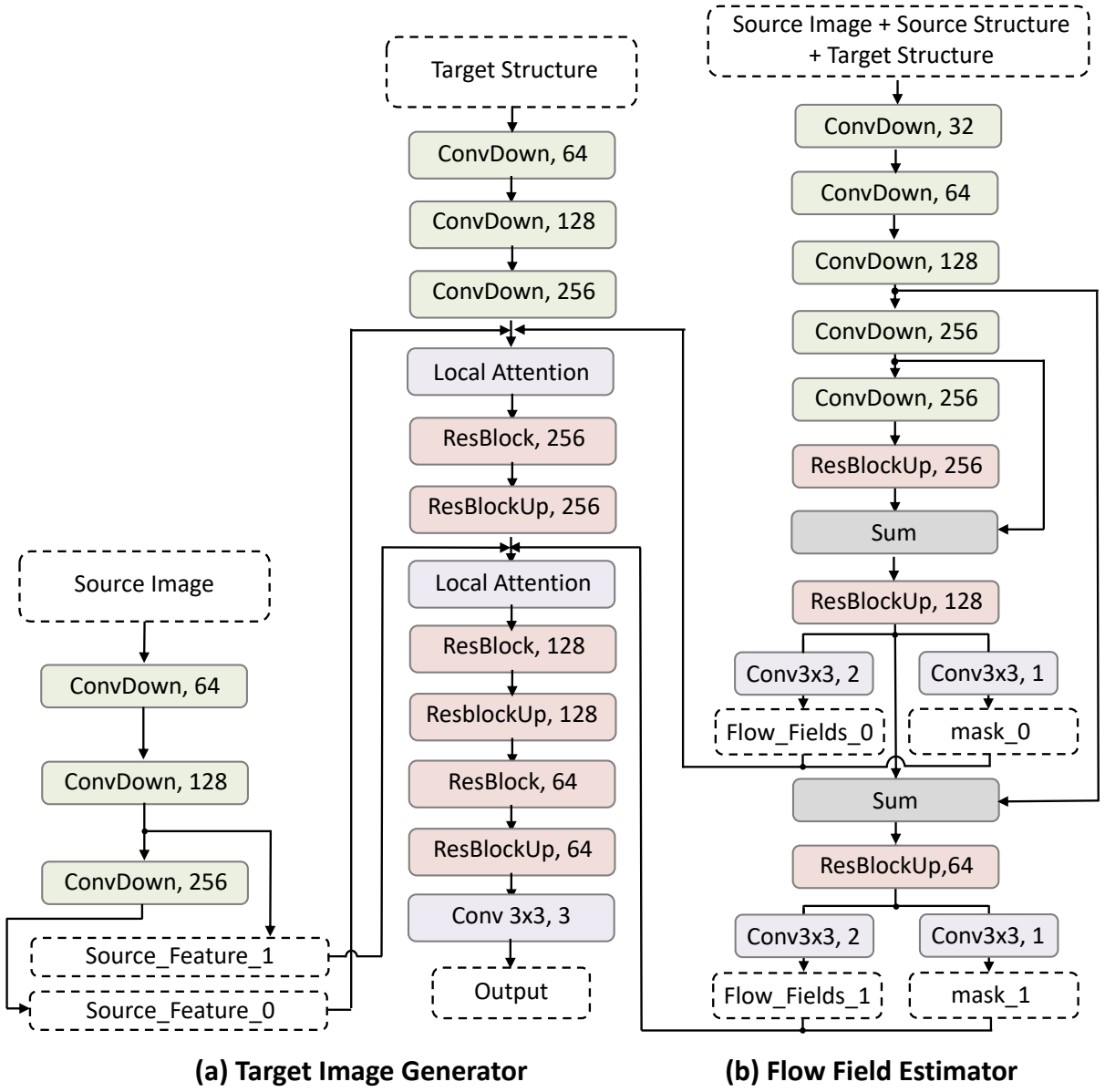


Fig. E.21. The network architecture of our GFLA model.

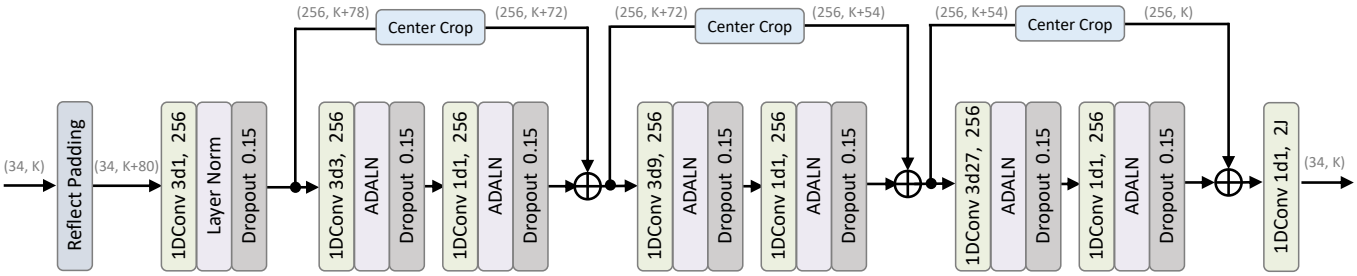


Fig. E.22. The architecture of our Motion Extraction Network.

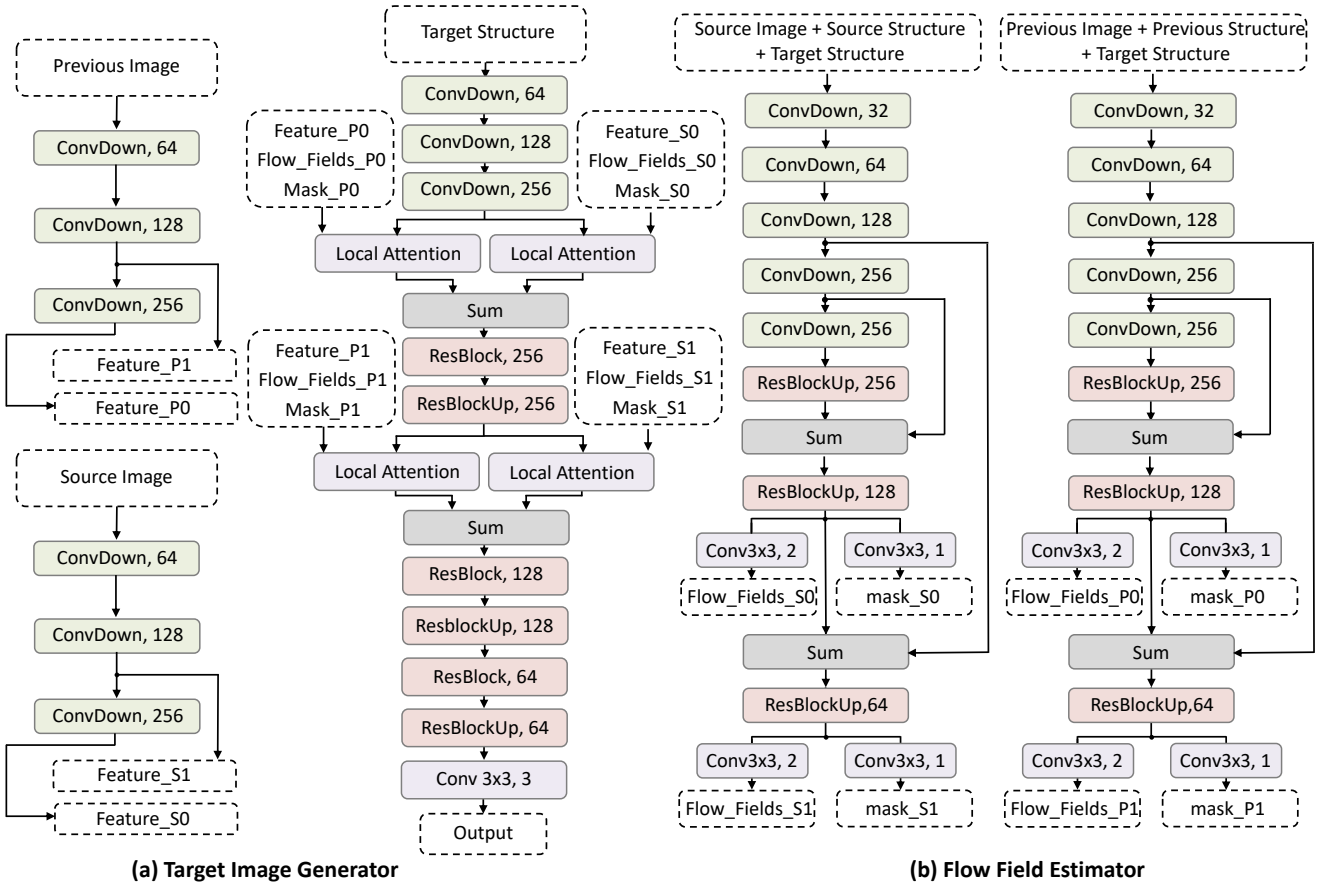


Fig. E.23. The network architecture of our sequential GFLA model.