Learning to Forecast and Refine Residual Motion for Image-to-Video Generation

Long Zhao[1], Xi Peng[2], Yu Tian[1], Mubbasir Kapadia[1], and Dimitris Metaxas[1]

[1] Rutgers University, Piscataway, USA
{lz311,yt219,mk1353,dnm}@cs.rutgers.edu
[2] Binghamton University, Binghamton, USA
xpeng@binghamton.edu

Abstract. We consider the problem of image-to-video translation, where an input image is translated into an output video containing motions of a single object. Recent methods for such problems typically train transformation networks to generate future frames conditioned on the structure sequence. Parallel work has shown that short high-quality motions can be generated by spatiotemporal generative networks that leverage temporal knowledge from the training data. We combine the benefits of both approaches and propose a two-stage generation framework where videos are generated from structures and then refined by temporal signals. To model motions more efficiently, we train networks to learn residual motion between the current and future frames, which avoids learning motion-irrelevant details. We conduct extensive experiments on two image-to-video translation tasks: facial expression retargeting and human pose forecasting. Superior results over the state-of-the-art methods on both tasks demonstrate the effectiveness of our approach.[3]

Keywords: Video generation · Motion forecasting · Residual learning

1 Introduction

Recently, Generative Adversarial Networks (GANs) [7] have attracted a lot of research interests, as they can be utilized to synthesize realistic-looking images or videos for various vision applications [15,16,39,44,46,47]. Compared with image generation, synthesizing videos is more challenging, since the networks need to learn the appearance of objects as well as their motion models. In this paper, we study a form of classic problems in video generation that can be framed as image-to-video translation tasks, where a system receives one or more images as the input and translates it into a video containing realistic motions of a *single* object. Examples include facial expression retargeting [13,21,34], future prediction [37,38,40], and human pose forecasting [5,6,39].

One approach for the long-term future generation [39,41] is to train a transformation network that translates the input image into each future frame separately

[3] The project website is publicly available at https://garyzhao.github.io/FRGAN.

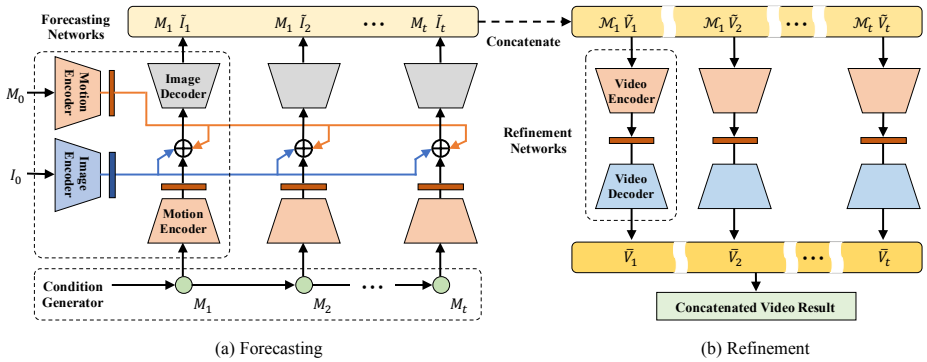

Fig. 1. Method overview. Videos are (a) generated from conditions and then (b) refined. Our framework consists of three components: a condition generator, motion forecasting networks and refinement networks. Each part is explained in the corresponding section.

conditioned by a sequence of structures. It suggests that it is beneficial to incorporate high-level structures during the generative process. In parallel, recent work [11,31,36,37,40] has shown that temporal visual features are important for video modeling. Such an approach produces temporally coherent motions with the help of spatiotemporal generative networks but is poor at long-term conditional motion generation since no high-level guidance is provided.

In this paper, we combine the benefits of these two methods. Our approach includes two motion transformation networks as shown in Figure 1, where the entire video is synthesized in a generation and then refinement manner. In the generation stage, the *motion forecasting networks* observe a single frame from the input and generate all future frames individually, which are conditioned by the structure sequence predicted by a *motion condition generator*. This stage aims to generate a coarse video where the spatial structures of the motions are preserved. In the refinement stage, spatiotemporal *motion refinement networks* are used for refining the output from the previous stage. It performs the generation guided by temporal signals, which targets at producing temporally coherent motions.

For more effective motion modeling, two transformation networks are trained in the *residual space*. Rather than learning the mapping from the structural conditions to motions directly, we force the networks to learn the differences between motions occurring in the current and future frames. The intuition is that learning only the residual motion avoids the redundant motion-irrelevant information, such as static backgrounds, which remains unchanged during the transformation. Moreover, we introduce a novel network architecture using *dense connections* for decoders. It encourages reusing spatially different features and thus yields realistic-looking results.

We experiment on two tasks: facial expression retargeting and human pose forecasting. Success in either task requires reasoning realistic spatial structures as well as temporal semantics of the motions. Strong performances on both tasks

demonstrate the effectiveness of our approach. In summary, our work makes the following *contributions*:

- We devise a novel two-stage generation framework for image-to-video translation, where the future frames are generated according to the spatial structure sequence and then refined with temporal signals;
- We investigate learning residual motion for video generation, which focuses on the motion-specific knowledge and avoids learning redundant or irrelevant details from the inputs;
- Dense connections between layers of decoders are introduced to encourage spatially different feature reuse during the generation process, which yields more realistic-looking results;
- We conduct extensive experimental validation on standard datasets which both quantitatively and subjectively compares our method with the state-of-the-arts to demonstrate the effectiveness of the proposed algorithm.

2 Related Work

Deep learning techniques have improved the accuracy of various vision systems [23,24,32,33]. Especially, a lot of generative problems [8,25,26,35] have been solved by GANs [7]. However, traditional frameworks fail to handle complicated tasks, e.g., to generate fine-grained images or videos with large motion changes. Recent approaches [16,42,43] prove that coarse-to-fine strategy can handle these cases. Our model also employs this strategy for video generation.

Xiong et al. [42] proposed an algorithm to generate video in two stages, but there are important differences between their work and ours. First, [42] is proposed for time-lapse videos while we can generate general videos. Second, we make use of structure conditions to guide the generation in the first stage but [42] models this stage with 3D convolutional networks. Finally, we can make long-term predictions while [42] only generates videos with fixed length.

Video Generation. Recent methods [17,38,39,41] solve image-to-video generation problem by training transformation networks that translate the input image into each future frame separately, together with a generator predicting the structure sequence which conditions the future frames. However, due to the absence of pixel-level temporal knowledge during the training process, motion artifacts can be observed from the results of these methods.

Other approaches explore learning temporal visual features from video with spatiotemporal networks. Ji et al. [11] showed how 3D convolutional networks could be applied to human action recognition. Tran et al. [36] employed spatiotemporal 3D convolutions to model features encoded in videos. Vondrick et al. [40] built a model to generate scene dynamics with 3D generative adversarial networks. Our method differs from the two-stream model of [40] in two aspects. First, our residual motion map disentangles motion from the input: the generated frame is conditioned on the current and future motion structures. Second, we can control object motions in future frames efficiently by using structure conditions. Therefore, our method can be applied to motion manipulation problems.

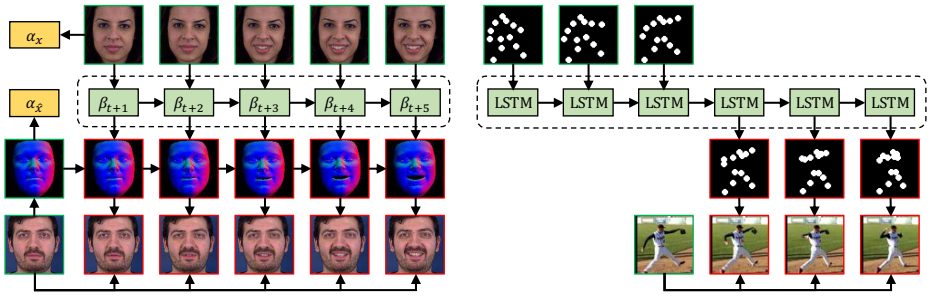

Fig. 2. Illustration of our motion condition generators designed for two tasks. *Left*: For facial expression retargeting, 3D Morphable Model [4] is utilized as domain knowledge to produce expression conditional sequence. *Right*: For human pose forecasting, the pose is represented by the 2D positions of joints. The LSTM [6] observes a sequence of human pose inputs and predicts the next several timesteps.

Dense Connections. Recent studies [9,10] used dense connections for image classification. They have proven that dense connections for encoders strengthen feature propagation and also encourage feature reuse. Instead, we introduce dense connections for decoders. Compared with multi-scale feature fusion in [14] where feature maps are only concatenated to the last layer of the network, our dense connections upsample and concatenate feature maps with different scales to all intermediate layers. Our approach is more efficient at feature re-use when utilized for generation, which yields sharper and more realistic-looking results.

3 Method

As shown in Figure 1, our framework consists of three components: a motion condition generator G_C, an image-to-image transformation network G_M for forecasting motion conditioned by G_C to each future frame individually, and a video-to-video transformation network G_R which aims to refine the video clips concatenated from the output of G_M. G_C is a task-specific generator that produces a sequence of structures to condition the motion of each future frame. Two discriminators are utilized for adversarial learning, where D_I differentiates real frames from generated ones and D_V is employed for video clips. In the following sections, we explain how each component is designed respectively.

3.1 Motion Condition Generators

In this section, we illustrate how the motion condition generators G_C are implemented for two image-to-video translation tasks: facial expression retargeting and human pose forecasting. One superiority of G_C is that domain-specific knowledge can be leveraged to help the prediction of motion structures.

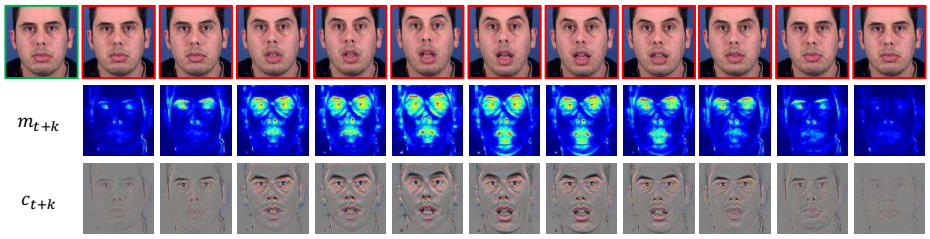

Fig. 3. Illustration of our residual formulation. We disentangle the motion differences between the input and future frames into a residual motion map m_{t+k} and a residual content map c_{t+k}. Compared with the difference map directly computed from them, our formulation makes the learning task much easier.

Facial Expression Retargeting. As shown in Figure 2, we utilize 3D Morphable Model (3DMM) [4] to model the sequence of expression motions. Given a video containing expression changes of an actor x, it can be parameterized with α_x and $(\beta_t, \beta_{t+1}, \ldots, \beta_{t+k})$ using 3DMM, where α_x represents the facial identity and β_t is the expression coefficients in the frame t. In order to retarget the sequence of expressions to another actor $\hat{x}$, we compute the facial identity vector $\alpha_{\hat{x}}$ and combine it with $(\beta_t, \beta_{t+1}, \ldots, \beta_{t+k})$ to reconstruct a new sequence of 3D face models with corresponding facial expressions. The conditional motion maps are the normal maps calculated from the 3D models respectively.

Human Pose Forecasting. We follow [39] to implement an LSTM architecture [6] as the human pose predictor. The human pose of each frame is represented by the 2D coordinate positions of joints. The LSTM observes consecutive pose inputs to identify the type of motion, and then predicts the pose for the next period of time. An example is shown in Figure 2. Note that the motion map is calculated by mapping the output 2D coordinates from the LSTM to heatmaps and concatenating them on depth.

3.2 Motion Forecasting Networks

Starting from the frame I_t at time t, our network synthesizes the future frame I_{t+k} by predicting the residual motion between them. Previous work [16,28] implemented this idea by letting the network estimate a difference map between the input and output, which can be denoted as:

$$I_{t+k} = I_t + G_M(I_t|M_t, M_{t+k}), \tag{1}$$

where M_t is the motion map which conditions I_t. However, this straightforward formulation easily fails when employed to handle videos including large motions, since learning to generate a combination of residual changes from both dynamic and static contents in a single map is quite difficult. Therefore, we introduce an enhanced formulation where the transformation is disentangled into a residual

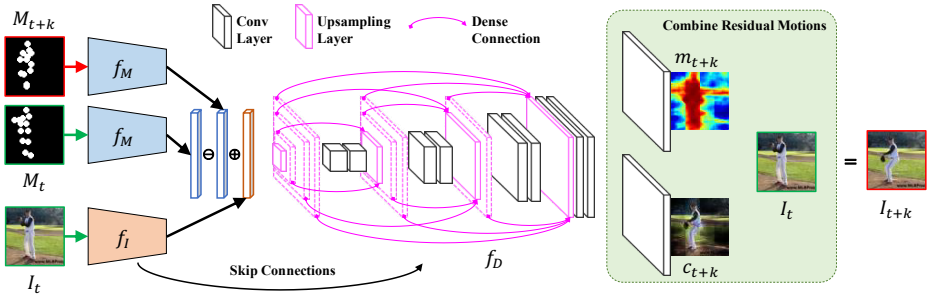

Fig. 4. Architecture of our motion forecasting network G_M. The network observes the input frame I_t with its corresponding motion map M_t, and the motion map of the future frame M_{t+k}. Through analogy learning, the network estimates the *residual motion* between the current frame I_t and future frame I_{t+k}. Note that the *dashed* layers upsample the inputs and connect them to the subsequent dense blocks.

motion map m_{t+k} and a residual content map c_{t+k} with the following definition:

$$I_{t+k} = \underbrace{m_{t+k} \odot c_{t+k}}_{\text{residual motion}} + \underbrace{(1 - m_{t+k}) \odot I_t}_{\text{static content}}, \tag{2}$$

where both m_{t+k} and c_{t+k} are predicted by G_M, and $\odot$ is element-wise multiplication. Intuitively, $m_{t+k} \in [0, 1]$ can be viewed as a spatial mask that highlights where the motion occurs. c_{t+k} is the content of the residual motions. By summing the residual motion with the static content, we can obtain the final result. Note that as visualized in Figure 3, m_{t+k} forces G_M to reuse the static part from the input and concentrate on inferring dynamic motions.

Architecture. Figure 4 shows the architecture of G_M, which is inspired by the visual-structure analogy learning [27]. The future frame I_{t+k} can be generated by transferring the structure differences from M_t to M_{t+k} to the input frame I_t. We use a motion encoder f_M, an image encoder f_I and a residual content decoder f_D to model this concept. And the residual motion is learned by:

$$\Delta(I_{t+k}, I_t) = f_D(f_M(M_{t+k}) - f_M(M_t) + f_I(I_t)). \tag{3}$$

Intuitively, f_M aims to identify key motion features from the motion map containing high-level structural information; f_I learns to map the appearance model of the input into an embedding space, where the motion feature transformations can be easily imposed to generate the residual motion; f_D learns to decode the embedding. Note that we add skip connections [20] between f_I and f_D, which makes it easier for f_D to reuse features of static objects learned from f_I.

Huang et al. [9,10] introduce dense connections to enhance feature propagation and reuse in the network. We argue that this is an appealing property for motion transformation networks as well, since in most cases the output frame shares similar high-level structure with the input frame. Especially, dense connections make it easy for the network to reuse features of different spatial positions

when large motions are involved in the image. The decoder of our network thus consists of multiple *dense connections*, each of which connects different dense blocks. A dense block contains two 3×3 convolutional layers. The output of a dense block is connected to the first convolutional layers located in *all* subsequent blocks in the network. As dense blocks have different feature resolutions, we upsample feature maps with lower resolutions when we use them as inputs into higher resolution layers.

Training Details. Given a video clip, we train our network to perform random jumps in time to learn forecasting motion changes. To be specific, for every iteration at training time, we sample a frame I_t and its corresponding motion map M_t given by G_C at time t, and then force it to generate frame I_{t+k} given motion map M_{t+k}. Note that in order to let our network perform learning in the entire residual motion space, k is also randomly defined for each iteration. On the other hand, learning with jumps in time can prevent the network from falling into suboptimal parameters as well [39]. Our network is trained to minimize the following objective function:

$$\mathcal{L}_{G_M} = \mathcal{L}_{rec}(I_{t+k}, \tilde{I}_{t+k}) + \mathcal{L}_r(m_{t+k}) + \mathcal{L}_{gen}. \tag{4}$$

$\mathcal{L}_{rec}$ is the reconstruction loss defined in the image space which measures the pixel-wise differences between the predicted and target frames:

$$\mathcal{L}_{rec}(I_{t+k}, \tilde{I}_{t+k}) = \|I_{t+k} - \tilde{I}_{t+k}\|_1, \tag{5}$$

where $\tilde{I}_{t+k}$ denotes the frame predicted by G_M. The reconstruction loss intuitively offers guidance for our network in making a rough prediction that preserves most content information of the target image. More importantly, it leads the result to share similar structure information with the input image. $\mathcal{L}_r$ is an L-1 norm regularization term defined as:

$$\mathcal{L}_r(m_{t+k}) = \|m_{t+k}\|_1, \tag{6}$$

where m_{t+k} is the residual motion map predicted by G_M. It forces the predicted motion changes to be sparse, since dynamic motions always occur in local positions of each frame while the static parts (e.g., background objects) should be unchanged. $\mathcal{L}_{gen}$ is the adversarial loss that enables our model to generate realistic frames and reduce blurs, and it is defined as:

$$\mathcal{L}_{gen} = -D_I([\tilde{I}_{t+k}, M_{t+k}]), \tag{7}$$

where D_I is the discriminator for images in adversarial learning. We concatenate the output of G_M and motion map M_{t+k} as the input of D_I and make the discriminator conditioned on the motion [18].

Note that we follow WGAN [3,8] to train D_I to measure the Wasserstein distance between distributions of the real images and results generated from G_M. During the optimization of D_I, the following loss function is minimized:

$$\mathcal{L}_{D_I} = D_I([\tilde{I}_{t+k}, M_{t+k}]) - D_I([I_{t+k}, M_{t+k}]) + \lambda \cdot \mathcal{L}_{gp}, \tag{8}$$

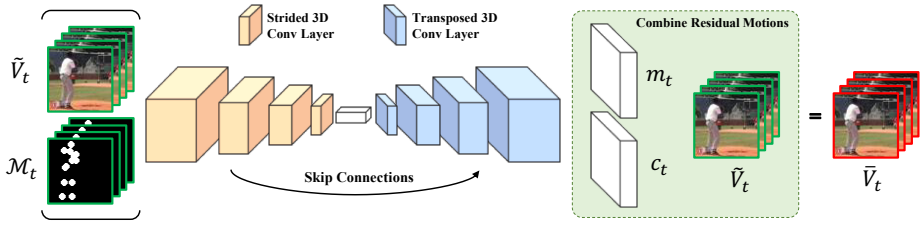

Fig. 5. Architecture of our motion refinement network G_R. The network receives temporally concatenated frames generated by G_M together with their corresponding conditional motion map as the input and aims to refine the video clip to be more temporally coherent. It performs learning in the residual motion space as well.

$$\mathcal{L}_{gp} = (\|\nabla_{[\hat{I}_{t+k}, M_{t+k}]} D_I([\hat{I}_{t+k}, M_{t+k}])\|_2 - 1)^2, \tag{9}$$

where λ is experimentally set to 10. $\mathcal{L}_{gp}$ is the gradient penalty term proposed by [8] where $\hat{I}_{t+k}$ is sampled from the interpolation of I_{t+k} and $\tilde{I}_{t+k}$, and we extend it to be conditioned on the motion M_{t+k} as well. The adversarial loss in combination with the rest of loss terms allows our network to generate high-quality frames given the motion conditions.

3.3 Motion Refinement Networks

Let $\tilde{V}_t = [\tilde{I}_{t+1}, \tilde{I}_{t+2}, \ldots, \tilde{I}_{t+K}]$ be the video clip with length K temporally concatenated from the outputs of G_M. The goal of the motion refinement network G_R is to refine $\tilde{V}_t$ to be more temporally coherent, which is achieved by performing pixel-level refinement with the help of spatiotemporal generative networks. We extends Equation 2 by adding one additional temporal dimension to let G_R estimate the residual between the real video clip V_t and $\tilde{V}_t$, which is defined as:

$$V_t = m_t \odot c_t + (1 - m_t) \odot \tilde{V}_t, \tag{10}$$

where m_t is a spatiotemporal mask which selects either to be refined for each pixel location and timestep, while c_t produces a spatiotemporal cuboid which stands for the refined motion content masked by m_t.

Architecture. Our motion refinement network roughly follows the architectural guidelines of [40]. As shown in Figure 5, we do not use pooling layers, instead strided and fractionally strided convolutions are utilized for in-network downsampling and upsampling. We also add skip connections to encourage feature reuse. Note that we concatenate the frames with their corresponding conditional motion maps as the inputs to guide the refinement.

Training Details. The key requirement for G_R is that the refined video should be temporal coherent in motion while preserving the annotation information

from the input. To this end, we propose to train the network by minimizing a combination of three losses which is similar to Equation 4:

$$\mathcal{L}_{G_R} = \mathcal{L}_{rec}(V_t, \bar{V}_t) + \mathcal{L}_r(m_t) + \bar{\mathcal{L}}_{gen}, \tag{11}$$

where $\bar{V}_t$ is the output of G_R. $\mathcal{L}_{rec}$ and $\mathcal{L}_r$ share the same definition with Equation 5 and 6 respectively. $\mathcal{L}_{rec}$ is the reconstruction loss that aims at refining the synthesized video towards the ground truth with minimal error. Compared with the self-regularization loss proposed by [29], we argue that the sparse regularization term $\mathcal{L}_r$ is also efficient to preserve the annotation information (e.g., the facial identity and the type of pose) during the refinement, since it force the network to only modify the essential pixels. $\bar{\mathcal{L}}_{gen}$ is the adversarial loss:

$$\bar{\mathcal{L}}_{gen} = -D_V([\bar{V}_t, \mathcal{M}_t]) - \frac{1}{K}\sum_{i=1}^{K} D_I([\bar{I}_{t+i}, M_{t+i}]), \tag{12}$$

where $\mathcal{M}_t = [M_{t+1}, M_{t+2}, \ldots, M_{t+K}]$ is the temporally concatenated condition motion maps, and $\bar{I}_{t+i}$ is the i-th frame of $\bar{V}_t$. In the adversarial learning term $\bar{\mathcal{L}}_{gen}$, both D_I and D_V play the role to judge whether the input is a real video clip or not, providing criticisms to G_R. The image discriminator D_I criticizes G_R based on individual frames, which is trained to determine if each frame is sampled from a real video clip. At the same time, D_V provides criticisms to G_R based on the whole video clip, which takes a fixed length video clip as the input and judges if a video clip is sampled from a real video as well as evaluates the motions contained. As suggested by [37], although D_V alone should be sufficient, D_I significantly improves the convergence and the final results of G_R.

We follow the same strategy as introduced in Equation 8 to optimize D_I. Note that in each iteration, one pair of real and generated frames is randomly sampled from V_t and $\bar{V}_t$ to train D_I. On the other hand, training D_V is also based on the WGAN framework, where we extend it to spatiotemporal inputs. Therefore, D_V is optimized by minimizing the following loss function:

$$\mathcal{L}_{D_V} = D_V([\bar{V}_t, \mathcal{M}_t]) - D_V([V_t, \mathcal{M}_t]) + \lambda \cdot \mathcal{L}_{gp}, \tag{13}$$

$$\mathcal{L}_{gp} = (\|\nabla_{[\hat{V}_t, \mathcal{M}_t]} D_V([\hat{V}_t, \mathcal{M}_t])\|_2 - 1)^2, \tag{14}$$

where $\hat{V}_t$ is sampled from the interpolation of V_t and $\bar{V}_t$. Note that G_R, D_I and D_V are trained alternatively. To be specific, we update D_I and D_V in one step while fixing G_R; in the alternating step, we fix D_I and D_V while updating G_R.

4 Experiments

We perform experiments on two image-to-video translation tasks: facial expression retargeting and human pose forecasting. For facial expression retargeting, we demonstrate that our method is able to combine domain-specific knowledge, such as 3DMM, to generate realistic-looking results. For human pose forecasting, experimental results show that our method yields high-quality videos when applied for video generation tasks containing complex motion changes.

4.1 Settings and Databases

To train our networks, we use Adam [12] for optimization with a learning rate of 0.0001 and momentums of 0.0 and 0.9. We first train the forecasting networks, and then train the refinement networks using the generated coarse frames. The batch size is set to 32 for all networks. Due to space constraints, we ask the reader to refer to the project website for the details of the network designs.

We use the *MUG Facial Expression Database* [1] to evaluate our approach on facial expression retargeting. This dataset is composed of 86 subjects (35 women and 51 men). We crop the face regions with regards to the landmark ground truth and scale them to 96×96. To train our networks, we use only the sequences representing one of the six facial expressions: anger, fear, disgust, happiness, sadness, and surprise. We evenly split the database into three groups according to the subjects. Two groups are used for training G_M and G_R respectively, and the results are evaluated on the last one. The 3D Basel Face Model [22] serves as the morphable model to fit the facial identities and expressions for the condition generator G_C. We use [48] to compute the 3DMM parameters for each frame. Note that we train G_R to refine the video clips every 32 frames.

The *Penn Action Dataset* [45] consists of 2326 video sequences of 15 different human actions, which is used for evaluating our method on human pose forecasting. For each action sequence in the dataset, 13 human joint annotations are provided as the ground truth. To remove very noisy joint ground-truth in the dataset, we follow the setting of [39] to sub-sample the actions. Therefore, 8 actions including baseball pitch, baseball swing, clean and jerk, golf swing, jumping jacks, jump rope, tennis forehand, and tennis serve are used for training our networks. We crop video frames based on temporal tubes to remove as much background as possible while ensuring the human actions are in all frames, and then scale each cropped frame to 64×64. We evenly split the standard dataset into three sets. G_M and G_R are trained in the first two sets respectively, while we evaluate our models in the last set. We employ the same strategy as [39] to train the LSTM pose generator. It is trained to observe 10 inputs and predict 32 steps. Note that G_R is trained to refine the video clips with the length of 16.

4.2 Evaluation on Facial Expression Retargeting

We compare our method to MCNet [38], MoCoGAN [37] and Villegas et al. [39] on the MUG Database. For each facial expression, we randomly select one video as the reference, and retarget it to all the subjects in the testing set with different methods. Each method only observes the input frame of the target subject, and performs the generation based on it. Our method and [39] share the same 3DMM-based condition generator as introduced in Section 3.1.

Quantitative Comparison. The quality of a generated video are measured by the Average Content Distance (ACD) as introduced in [37]. For each generated video, we make use of OpenFace [2], which outperforms human performance in the face recognition task, to measure the video quality. OpenFace produces a

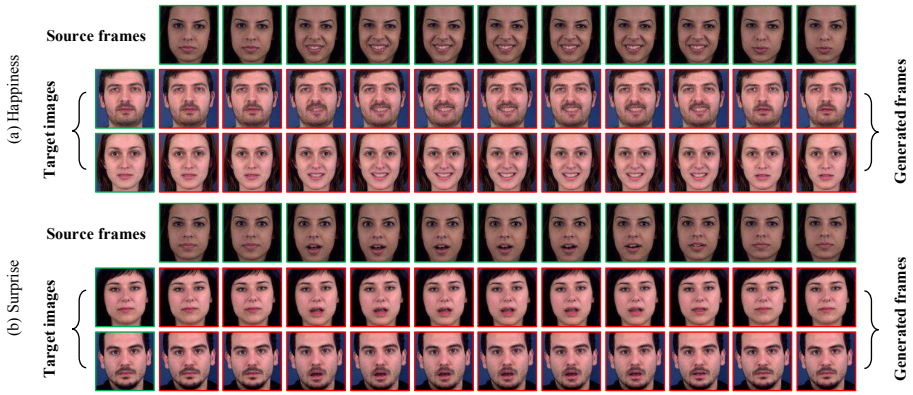

Fig. 6. Examples of facial expression retargeting using our algorithm on the MUG Database [1]. We show two expressions as an illustration: (a) happiness and (b) surprise. The reference video and the input target images are highlighted in *green*, while the generated frames are highlighted in *red*. The results are sampled every 8 frames.

Table 1. Video generation quality comparison on the MUG Dataset [1]. We also compute the ACD-* score for the training set, which is the reference.

Methods	ACD-I	ACD-C
MCNet [38]	0.545	0.322
Villegas et al. [39]	0.683	0.130
MoCoGAN [37]	0.291	0.205
Ours	**0.184**	**0.107**
Reference	0.109	0.098

Table 2. Average user preference score (the average number of times, a user prefers our result to the competing one) on the MUG Dataset [1]. Our results own higher preference scores compared with the others.

Methods	Preference (%)
Ours / MCNet [38]	**84.2** / 15.8
Ours / Villegas et al. [39]	**74.6** / 25.4
Ours / MoCoGAN [37]	**62.5** / 37.5

feature vector for each frame, and then the ACD is calculated by measuring the L-2 distance of these vectors. We introduce two variants of the ACD in this experiment. The ACD-I is the average distance between each generated frame and the original input frame. It aims to judge if the facial identity is well-preserved in the generated video. The ACD-C is the average pairwise distance of the per-frame feature vectors in the generated video. It measures the content consistency of the generated video.

Table 1 summarizes the comparison results. From the table, we find that our method achieves ACD-* scores both lower than 0.2, which is substantially better than the baselines. One interesting observation is that [39] has the worst ACD-I but its ACD-C is the second best. We argue that this is due to the high-level information offered by our 3DMM-based condition generator, which plays a vital role for producing content consistency results. Our method outperforms other state-of-the-arts, since we utilize both domain knowledge (3DMM) and temporal signals for video generation. We show that it is greatly beneficial to incorporate both factors into the generative process.

We also conduct a user study to quantitatively compare these methods. For each method, we randomly select 10 videos for each expression. We then randomly pair the videos generated by ours with the videos from one of the competing methods to form 54 questions. For each question, 3 users are asked to select the video which is more realistic. To be fair, the videos from different methods are shown in random orders. We report the average user preference scores (the average number of times, a user prefers our result to the competing one) in Table 2. We find that the users consider the videos generated by ours more realistic most of the time. This is consistent with the ACD results in Table 1, in which our method substantially outperforms the baselines.

Visual Results. In Figure 6, we show the visual results (the expressions of happiness and surprise) generated by our method. We observe that our method is able to generate realistic motions while the facial identities are well-preserved. We hypothesize that the domain knowledge (3DMM) employed serves as a good prior which improves the generation. More visual results of different expressions and subjects are given on the project website.

4.3 Evaluation on Human Pose Forecasting

We compare our approach with VGAN [40], Mathieu et al. [17] and Villegas et al. [39] on the Penn Action Dataset. We produce the results of their models according to their papers or reference codes. For fair comparison, we generate videos with 32 generated frames using each method, and evaluate them starting from the first frame. Note that we train an individual VGAN for different action categories with randomly picked video clips from the dataset, while one network among all categories are trained for every other method. Both [39] and our method perform the generation based on the pre-trained LSTM provided by [39], and we train [39] through the same strategy of our motion forecasting network G_M.

Implementation. Following the settings of [39], we engage the feature similarity loss term $\mathcal{L}_{feat}$ for our motion forecasting network G_M to capture the appearance (C_1) and structure (C_2) of the human action. This loss term is added to Equation 4, which is defined as:

$$\mathcal{L}_{feat} = \|C_1(I_{t+k}) - C_1(\tilde{I}_{t+k})\|_2^2 + \|C_2(I_{t+k}) - C_2(\tilde{I}_{t+k})\|_2^2, \tag{15}$$

where we use the last convolutional layer of the VGG16 Network [30] as C_1, and the last layer of the Hourglass Network [19] as C_2. Note that we compute the bounding box according to the group truth to crop the human of interest for each frame, and then scale it to 224×224 as the input of the VGG16.

Results. We evaluate the predictions using Peak Signal-to-Noise Ratio (PSNR) and Mean Square Error (MSE). Both metrics perform pixel-level analysis between the ground truth frames and the generated videos. We also report the

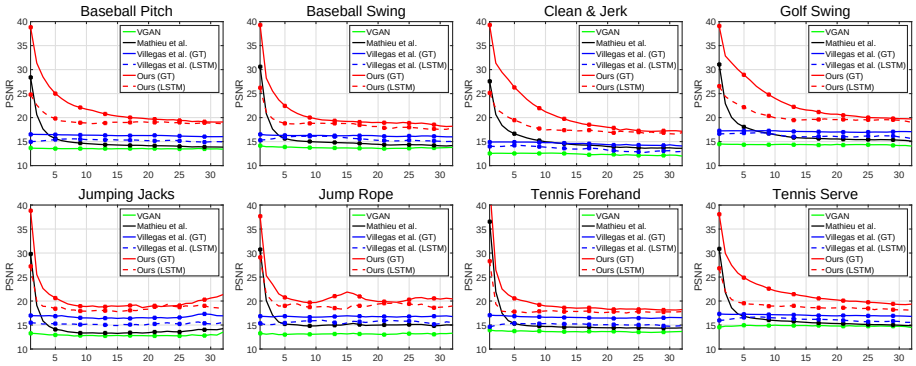

Fig. 7. Comparison of state-of-the-arts using Peak Signal-to-Noise Ratio (PSNR) on different human action categories from the Penn Action Dataset [45].

Table 3. Comparison of state-of-the-art algorithms on the Penn Action Database [45]. A smaller MSE score means better performance.

Methods	MSE	MSE (GT)
VGAN [40]	0.047	-
Mathieu et al. [17]	0.041	-
Villegas et al. [39]	0.030	0.025
Ours	**0.023**	**0.011**

Table 4. Quantitative results of ablation study. We report the ACD-* scores on the MUG Database [1] and MSE scores on the Penn Action Dataset [45].

Settings	ACD-I	ACD-C	MSE
G_M ~~(Dense)~~, ~~G_R~~	0.459	0.155	0.027
G_M (Dense), ~~G_R~~	0.252	0.140	0.014
G_M (Dense), G_R	**0.184**	**0.107**	**0.011**

results of our method and [39] using the condition motion maps computed from the ground truth joints (GT). The results are shown in Figure 7 and Table 3 respectively. From these two scores, we discover that the proposed method achieves better quantitative results which demonstrates the effectiveness of our algorithm.

Figure 8 shows visual comparison of our method with [39]. We can find that the predicted future of our method is closer to the ground-true future. To be speclfic, our method yields more consistent motions and keeps human appearances as well. Due to space constraints, we ask the reader to refer to the project website for more side by side visual results.

4.4 Ablation Study

Our method consists of three main modules: residual learning, dense connections for the decoder and the two-stage generation schema. Without residual learning, our network decays to [39]. As shown in Section 4.2 and 4.3, ours outperforms [39] which demonstrates the effectiveness of residual learning. To verify the rest modules, we train one partial variant of G_M, where the dense connections are not employed in the decoder f_D. Then we evaluate three different settings of our method on both tasks: G_M without dense connections, using only G_M for gener-

Fig. 8. Visual comparison of our method with Villegas et al. [39] on the Penn Action Dataset [45]. Examples are sampled from the action of baseball (*top*) and tennis (*bottom*) respectively. The results are taken every 5 frames.

ation and our full model. Note that in order to get rid of the influence from the LSTM, we report the results using the conditional motion maps calculated from the ground truth on the Penn Action Dataset. Results are shown in Table 4. Our approach with more modules performs better than those with less components, which suggests the effectiveness of each part of our algorithm.

5 Conclusions

In this paper, we combine the benefits of high-level structural conditions and spatiotemporal generative networks for image-to-video translation by synthesizing videos in a generation and then refinement manner. We have applied this method to facial expression retargeting where we show that our method is able to engage domain knowledge for realistic video generation, and to human pose forecasting where we demonstrate that our method achieves higher performance than state-of-the-arts when generating videos involving large motion changes. We also incorporate residual learning and dense connections to produce high-quality results. In the future, we plan to further explore the use of our framework for other image or video generation tasks.

Acknowledgment. This work is partly supported by the Air Force Office of Scientific Research (AFOSR) under the Dynamic Data-Driven Application Systems Program, NSF 1763523, 1747778, 1733843 and 1703883 Awards. Mubbasir Kapadia has been funded in part by NSF IIS-1703883, NSF S&AS-1723869, and DARPA SocialSim-W911NF-17-C-0098.

References

1. Aifanti, N., Papachristou, C., Delopoulos, A.: The MUG facial expression database. In: International Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS) (2010)
2. Amos, B., Ludwiczuk, B., Satyanarayanan, M.: OpenFace: A general-purpose face recognition library with mobile applications. Tech. rep., CMU-CS-16-118, CMU School of Computer Science (2016)
3. Arjovsky, M., Chintala, S., Bottou, L.: Wasserstein Generative Adversarial Networks. In: International Conference on Machine Learning (ICML) (2017)
4. Blanz, V., Vetter, T.: Face Recognition Based on Fitting a 3D Morphable Model. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) **25**(9), 1063–1074 (2003)
5. Chao, Y.W., Yang, J., Price, B., Cohen, S., Deng, J.: Forecasting human dynamics from static images. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017)
6. Fragkiadaki, K., Levine, S., Felsen, P., Malik, J.: Recurrent network models for human dynamics. In: IEEE International Conference on Computer Vision (ICCV). pp. 4346–4354 (2015)
7. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Annual Conference on Neural Information Processing Systems (NIPS). pp. 2672–2680 (2014)
8. Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., Courville, A.: Improved Training of Wasserstein GANs. In: Annual Conference on Neural Information Processing Systems (NIPS) (2017)
9. Huang, G., Liu, S., van der Maaten, L., Weinberger, K.Q.: Condensenet: An efficient densenet using learned group convolutions. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2018)
10. Huang, G., Liu, Z., van der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017)
11. Ji, S., Xu, W., Yang, M., Yu, K.: 3D convolutional neural networks for human action recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) **35**(1), 221–231 (2013)
12. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: International Conference on Learning Representations (ICLR) (2014)
13. Laine, S., Karras, T., Aila, T., Herva, A., Saito, S., Yu, R., Li, H., Lehtinen, J.: Production-level Facial Performance Capture Using Deep Convolutional Neural Networks. In: Proceedings of the ACM SIGGRAPH / Eurographics Symposium on Computer Animation (2017)
14. Liu, Z., Yeh, R.A., Tang, X., Liu, Y., Agarwala, A.: Video frame synthesis using deep voxel flow. In: IEEE International Conference on Computer Vision (ICCV) (2017)
15. Lu, J., Issaranon, T., Forsyth, D.: SafetyNet: Detecting and Rejecting Adversarial Examples Robustly. In: IEEE International Conference on Computer Vision (ICCV) (2017)
16. Ma, L., Jia, X., Sun, Q., Schiele, B., Tuytelaars, T., Van Gool, L.: Pose guided person image generation. In: Annual Conference on Neural Information Processing Systems (NIPS). pp. 405–415 (2017)

17. Mathieu, M., Couprie, C., LeCun, Y.: Deep multi-scale video prediction beyond mean square error. In: International Conference on Learning Representations (ICLR) (2016)
18. Mirza, M., Osindero, S.: Conditional generative adversarial nets. arXiv preprint arXiv:1411.1784 (2014)
19. Newell, A., Yang, K., Deng, J.: Stacked hourglass networks for human pose estimation. In: European Conference on Computer Vision (ECCV). pp. 483–499 (2016)
20. Olaf Ronneberger, Philipp Fischer, T.B.: U-Net: Convolutional Networks for Biomedical Image Segmentation. In: Medical Image Computing and Computer-Assisted Intervention (MICCAI) (2015)
21. Olszewski, K., Li, Z., Yang, C., Zhou, Y., Yu, R., Huang, Z., Xiang, S., Saito, S., Kohli, P., Li, H.: Realistic Dynamic Facial Textures from a Single Image Using GANs. In: IEEE International Conference on Computer Vision (ICCV) (2017)
22. Paysan, P., Knothe, R., Amberg, B., Romdhani, S., Vetter, T.: A 3D Face Model for Pose and Illumination Invariant Face Recognition. In: IEEE International Conference on Advanced Video and Signal based Surveillance (AVSS) for Security, Safety and Monitoring in Smart Environments (2009)
23. Peng, X., Feris, R.S., Wang, X., Metaxas, D.N.: A recurrent encoder-decoder network for sequential face alignment. In: European Conference on Computer Vision (ECCV). pp. 38–56 (2016)
24. Peng, X., Huang, J., Hu, Q., Zhang, S., Elgammal, A., Metaxas, D.: From Circle to 3-Sphere: Head Pose Estimation by Instance Parameterization. Computer Vision and Image Understanding (CVIU) **136**, 92–102 (2015)
25. Peng, X., Tang, Z., Yang, F., Feris, R.S., Metaxas, D.: Jointly optimize data augmentation and network training: Adversarial data augmentation in human pose estimation. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2226–2234 (2018)
26. Perarnau, G., van de Weijer, J., Raducanu, B., Álvarez, J.M.: Invertible Conditional GANs for image editing. In: NIPS Workshop on Adversarial Training (2016)
27. Reed, S.E., Zhang, Y., Zhang, Y., Lee, H.: Deep visual analogy-making. In: Annual Conference on Neural Information Processing Systems (NIPS) (2015)
28. Shen, W., Liu, R.: Learning residual images for face attribute manipulation. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017)
29. Shrivastava, A., Pfister, T., Tuzel, O., Susskind, J., Wang, W., Webb, R.: Learning from simulated and unsupervised images through adversarial training. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017)
30. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: International Conference on Learning Representations (ICLR) (2015)
31. Simonyan, K., Zisserman, A.: Two-stream convolutional networks for action recognition in videos. In: Annual Conference on Neural Information Processing Systems (NIPS). pp. 568–576 (2014)
32. Tang, Z., Peng, X., Geng, S., Wu, L., Zhang, S., Metaxas, D.N.: Quantized Densely Connected U-Nets for Efficient Landmark Localization. In: European Conference on Computer Vision (ECCV) (2018)
33. Tang, Z., Peng, X., Geng, S., Zhu, Y., Metaxas, D.: CU-Net: Coupled U-Nets. In: British Machine Vision Conference (BMVC) (2018)
34. Thies, J., Zollhöfer, M., Stamminger, M., Theobalt, C., Nießner, M.: Face2Face: Real-time Face Capture and Reenactment of RGB Videos. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016)

35. Tian, Y., Peng, X., Zhao, L., Zhang, S., Metaxas, D.N.: CR-GAN: Learning Complete Representations for Multi-view Generation. In: International Joint Conference on Artificial Intelligence (IJCAI). pp. 942–948 (2018)
36. Tran, D., Bourdev, L., Fergus, R., Torresani, L., Paluri, M.: Learning spatiotemporal features with 3D convolutional networks. In: IEEE International Conference on Computer Vision (ICCV). pp. 4489–4497 (2015)
37. Tulyakov, S., Liu, M.Y., Yang, X., Kautz, J.: MoCoGAN: Decomposing Motion and Content for Video Generation. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2018)
38. Villegas, R., Yang, J., Hong, S., Lin, X., Lee, H.: Decomposing motion and content for natural video sequence prediction. In: International Conference on Learning Representations (ICLR) (2017)
39. Villegas, R., Yang, J., Zou, Y., Sohn, S., Lin, X., Lee, H.: Learning to generate long-term future via hierarchical prediction. In: International Conference on Machine Learning (ICML) (2017)
40. Vondrick, C., Pirsiavash, H., Torralba, A.: Generating videos with scene dynamics. In: Annual Conference on Neural Information Processing Systems (NIPS) (2016)
41. Xingjian, S., Chen, Z., Wang, H., Yeung, D.Y., Wong, W.K., Woo, W.c.: Convolutional LSTM network: A machine learning approach for precipitation nowcasting. In: Annual Conference on Neural Information Processing Systems (NIPS). pp. 802–810 (2015)
42. Xiong, W., Luo, W., Ma, L., Liu, W., Luo, J.: Learning to generate time-lapse videos using multi-stage dynamic generative adversarial networks. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2018)
43. Zhang, H., Xu, T., Li, H., Zhang, S., Wang, X., Huang, X., Metaxas, D.: StackGAN: Text to Photo-realistic Image Synthesis with Stacked Generative Adversarial Networks. In: IEEE International Conference on Computer Vision (ICCV) (2017)
44. Zhang, H., Sindagi, V., Patel, V.M.: Image De-raining Using a Conditional Generative Adversarial Network. arXiv preprint arXiv:1701.05957 (2017)
45. Zhang, W., Zhu, M., Derpanis, K.: From Actemes to Action: A Strongly-supervised Representation for Detailed Action Understanding. In: IEEE International Conference on Computer Vision (ICCV) (2013)
46. Zhang, Z., Xie, Y., Yang, L.: Photographic text-to-image synthesis with a hierarchically-nested adversarial network. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2018)
47. Zhang, Z., Yang, L., Zheng, Y.: Translating and segmenting multimodal medical volumes with cycle-and shapeconsistency generative adversarial network. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2018)
48. Zhu, X., Lei, Z., Liu, X., Shi, H., Li, S.: Face Alignment Across Large Poses: A 3D Solution. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016)