Flow-Grounded Spatial-Temporal Video Prediction from Still Images

Yijun Li[1(✉)], Chen Fang[2], Jimei Yang[2], Zhaowen Wang[2], Xin Lu[2],
and Ming-Hsuan Yang[1,3]

[1] University of California, Merced, Merced, USA
{yli62,mhyang}@ucmerced.edu
[2] Adobe Research, San Jose, USA
{cfang,jimyang,zhawang,xinl}@adobe.com
[3] Google, Mountain View, USA

Abstract. Existing video prediction methods mainly rely on observing multiple historical frames or focus on predicting the next one-frame. In this work, we study the problem of generating consecutive multiple future frames by observing one single still image only. We formulate the multi-frame prediction task as a multiple time step flow (multi-flow) prediction phase followed by a flow-to-frame synthesis phase. The multi-flow prediction is modeled in a variational probabilistic manner with spatial-temporal relationships learned through 3D convolutions. The flow-to-frame synthesis is modeled as a generative process in order to keep the predicted results lying closer to the manifold shape of real video sequence. Such a two-phase design prevents the model from directly looking at the high-dimensional pixel space of the frame sequence and is demonstrated to be more effective in predicting better and diverse results. Extensive experimental results on videos with different types of motion show that the proposed algorithm performs favorably against existing methods in terms of quality, diversity and human perceptual evaluation.

Keywords: Future prediction · Conditional variational autoencoder
3D convolutions

1 Introduction

Part of our visual world constantly experiences situations that require us to forecast what will happen over time by observing one still image from a single moment. Studies in neuroscience show that this *preplay* activity might constitute an automatic prediction mechanism in human visual cortex [1]. Given the great progress in artificial intelligence, researchers also begin to let machines learn to perform such a predictive activity for various applications. For example in Fig. 1(top), from a snapshot by the surveillance camera, the system is expected to predict the man's next action which could be used for safety precautions. Another application in computational photography is turning still images into vivid cinemagraphs for aesthetic effects, as shown in Fig. 1(bottom).

In this work, we mainly study how to generate pixel-level future frames in multiple time steps given one still image. A number of existing prediction models [2–5] are under the assumption of observing a short video sequence (>1 frame). Since multiple historical frames explicitly exhibit obvious motion cues, most of them use deterministic models to render a fixed future sequence. In contrast, our single-image based prediction task, without any motion information provided, implies that there are obvious uncertainties existed in both spatial and temporal domains. Therefore we propose a probabilistic model based on a conditional variational autoencoder (cVAE) to model the uncertainty. Our probabilistic model has two unique features. First, it is a 3D-cVAE model, i.e., the autoencoder is designed in a spatial-temporal architecture with 3D convolution layers. The 3D convolutional layer [6], which takes a volume as input, is able to capture correlations between the spatial and temporal dimension of signals, thereby rendering distinctive spatial-temporal features for better predictions. Second, the output of our model is optical flows which characterize the spatial layout of how pixels are going to move step by step. Different from other methods that predict trajectories [7], frame differences [8] or frame pixels [5], the flow is a more natural and general representation of motions. It serves as a relatively low-dimensional reflection of high-level structures and can be obtained in an unsupervised manner.
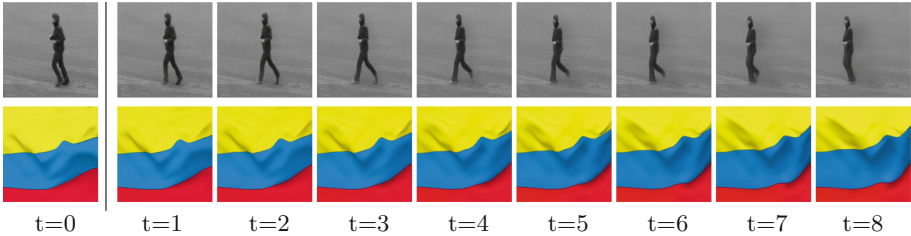

Fig. 1. Multi-step future sequences generated by our algorithm (t = 1 − 8) conditioned on one single still image (t = 0). Images are of size 128 × 128.

With the predicted flows, we next formulate the full frame synthesis as a generation problem. Due to the existence of occlusions, flow-based pixel-copying operations (e.g., warping) are obviously ineffective here. The model should be capable of "imagining" the appearance of future frames and removing the unnecessary parts in the previous frame at the same time. Therefore we propose a generative model *Flow2rgb* to generate pixel-level future frames. Such a model is non-trivial and is demonstrated to be effective in keeping the generated sequence staying close to the manifold of real sequences (Fig. 5). Overall, we formulate the multi-frame prediction task as a multiple time step flow prediction phase followed by a flow-to-frame generation phase. Such a two-phase design prevents the model from directly looking at the high-dimensional pixel space of the frame sequence and is demonstrated to be more effective in predicting better results. During the

testing, by drawing different samples from the learned latent distribution, our approach can also predict diverse future sequences.

The main contributions of this work are summarized as follows:

– We propose a spatial-temporal conditional VAE model (3D-cVAE) to predict future flows in multiple time steps. The diversity in predictions is realized by drawing different samples from the learned distribution.
– We present a generative model that learns to generate the pixel-level appearance of future frames based on predicted flows.
– We demonstrate the effectiveness of our method for predicting sequences that contain both articulated (e.g., humans) objects and dynamic textures (e.g., clouds).

2   Related Work

Action Prediction. The macroscopic analysis of prediction based on the given frame(s) can be predicting what event is going to happen [9–11], trajectory paths [12], or recognizing the type of human activities [13,14]. Some of early methods are supervised, requiring labels (e.g., bounding boxes) of the moving object. Later approaches [14] realize the unsupervised way of prediction by relying on the context of scenes. However, these approaches usually only provide coarse predictions of how the future will evolve and are unable to tell richer information except for a action (or event) label.

Pixel-Level Frame Prediction. Recent prediction methods move to the microcosmic analysis of more detailed information in the future. This is directly reflected by requiring the pixel-level generation of future frames in multiple time steps. With the development of deep neural networks, especially when recursive modules are extensively used, predicting realistic future frames has being dominated. Much progress has been made in the generated quality of future outputs by designing different network structures [2,15–18] or using different learning techniques, including adversarial loss [19,20], motion/content separation [4,5,21], and transformation parameters [22,23].

Our work also aims at accurate frame predictions but the specific setting is to model the uncertainties of multi-frame prediction given a single still image as input. In terms of multi-frame predictions conditioning on still images, closest work to ours are [24,25]. However, [24] only predicts the pose information and the proposed model is deterministic. The work in [25] also estimates poses first and then use an image-analogy strategy to generate frames. But their pose generation step relies on observing multiple frames. Moreover, both approaches employ the recursive module (e.g., recurrent neural networks) for consecutive predictions which may overemphasize on learning the temporal information only. Instead, we use the 3D convolutional layer [6] which takes a volume as input. Since both spatial and temporal information are encoded together, the 3D convolution can generally capture correlations between the spatial and temporal dimension of signals, thereby rendering distinctive spatial-temporal features [6]. In addition,

both [24,25] focus on human dynamics while our work targets on both articulated objects and dynamic textures.

In terms of modeling future uncertainties, two methods [7,8] are closely related. However, Xue et al. [8] only model the uncertainty in the next one-step prediction. If we iteratively run the one-step prediction model for multi-step predictions, the frame quality will degrade fast through error accumulations, due to the lack of temporal relationships modeling between frames. Though Walker et al. [7] could keep forecasting over the course of one second, instead of predicting real future frames, it only predicts the dense trajectory of pixels. Also such a trajectory-supervised modeling requires laborious human labeling. Different from these methods, our approach integrates the multi-frame prediction and uncertainty modeling in one model.

Dynamic Textures. The above-mentioned methods mainly focus on the movement of articulated objects (e.g., human). In contrast, dynamic textures often exhibit more randomness in the movement of texture elements. Both traditional methods based on linear dynamical systems [26,27] and neural network based methods [28] require learning a model for each sequence example. Different from those methods, we collect a large number of dynamic texture video data and aims at modeling the general distribution of their motions. Such a model can immediately serve as an editing tool when animating static texture examples.

3   Proposed Algorithm

We formulate the video prediction as two phases: flow prediction and flow-to-frame generation. The flow prediction phase, triggered by a noise, directly predicts a set of consecutive flow maps conditioned on the observed first frame. Then the flow-to-frame phase iteratively synthesizes future frames with the previous frame and the corresponding predicted flow map, starting from the first given frame and first predicted flow map.
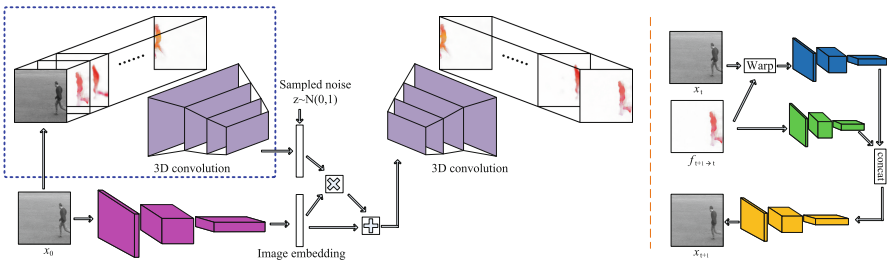

Fig. 2. Architecture of the proposed multi-step prediction network. It consists of a 3D-cVAE (left) for predicting consecutive flows and a *Flow2rgb* model to generate future frame pixels (right). During the testing, the encoder (blue rectangle) of 3D-cVAE is no longer used and we directly sample points from the distribution for predictions. (Color figure online)

3.1 Flow Prediction

Figure 2(left) illustrates the architecture of our proposed model for predicting consecutive optical flows. Formally, our model is a conditional variational autoencoder [29,30] with a spatial-temporal convolutional architecture (3D-cVAE). Given a sequence $X = \{x_i\}_0^M$ with x_0 as the starting frame, we denote the set of consecutive optical flows between adjacent frames in X as $F = \{f_i\}_0^{M-1}$. The network is trained to map the observation F (conditioned on x_0) to the latent variable z which are likely to reproduce the F. In order to avoid training a deterministic model, we produces a distribution over z values, which we sample from before the decoding. Such a variational distribution $q_\phi(z|x_0, F)$, known as the recognition model in [30], is assumed to be trained to follow a Gaussian distribution $p_z(z)$. Given a sampled z, the decoder decodes the flow F from the conditional distribution $p_\theta(F|x_0, z)$. Therefore the whole objective of network training is to maximize the variational lower-bound [29] of the following negative log-likelihood function:

$$\mathcal{L}(x_0, F; \theta, \phi) \approx -\mathcal{D}_{KL}(q_\phi(z|x_0, F)||p_z(z)) + \frac{1}{L}\sum_1^L \log p_\theta(F|x_0, z), \qquad (1)$$

where $\mathcal{D}_{KL}$ is the Kullback-Leibler (K-L) divergence and L is the number of samples. Maximizing the term at rightmost in (1) is equivalent to minimizing the L1 distance between the predicted flow and the observed flow. Hence the loss $\mathcal{L}$ consists of a flow reconstruction loss and a K-L divergence loss.

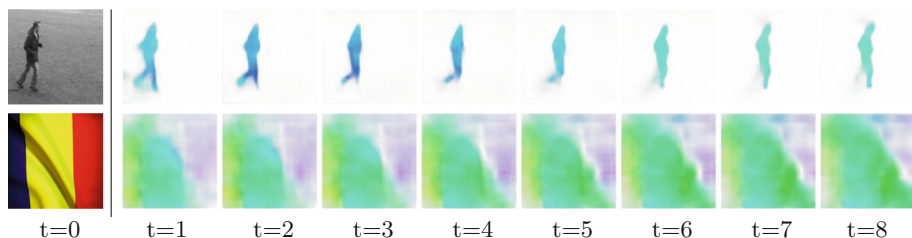| t=0 | t=1 | t=2 | t=3 | t=4 | t=5 | t=6 | t=7 | t=8 |

Fig. 3. Examples of our multi-step flow prediction. During the testing, by simply sampling a noise from $N \sim (0,1)$, we obtain a set of consecutive flows that describe the future motion field in multiple time steps. Note that since we have a warp operation in the later flow-to-frame step (Sect. 3.2) and the backward warping will not result in *holes* in results, we predict the backward flow in this step, i.e., the motion from x_{t+1} to x_t. This is just for convenience and we empirically do not find obvious difference between predicting forward and backward flows.

Different from traditional cVAE models [7,8,30], our 3D-cVAE model employs the 3D convolution (purple blocks in Fig. 2) which is demonstrated to be well-suited for spatial-temporal feature learning [6,19]. In terms of network architecture, the 3D convolutional network outputs multiple (a volume of)

flow maps instead of one, which can be used to predict multiple future frames. More importantly, the spatial-temporal relationship between adjacent flows are implicitly modeled during the training due to the 3D convolution operations, ensuring that the predicted motions are continuous and reasonable over time. In order to let the variational distribution $q_\phi(z|x_0, F)$ conditioned on the starting frame, we stack x_0 with each flow map f_i in F as the encoder input. Meanwhile, learning the conditional distribution $p_\theta(F|x_0, z)$ for flow reconstruction also needs to be conditioned on x_0 in the latent space. Therefore, we propose an image encoder (pink blocks in Fig. 2) to first map x_0 to a latent vector that has the same dimension as z. Inspired by the image analogy work [31], we use a conditioning strategy of combining the multiplication and addition operation, as shown in Fig. 2(left). After we obtain the flow sequence for the future, we proceed to generate the pixel-level full frames.

3.2 Frame Generation

Given the flow information, a common way to obtain the next frame is warping or pixel copying [32]. However, due to the existence of occlusions, the result is often left with unnecessary pixels inherited from the previous frame. The frame interpolation work [33] predicts a mask indicating where to copy pixels from previous and next frame. But they require at least two frames to infer the occluded parts. Since we only observe one image, it is straightforward to formulate this step as a generation process, meaning that this model can "imagine" the appearance of next frame according to the flow and starting frame. The similar idea is also applied in the task of novel view synthesis [34].
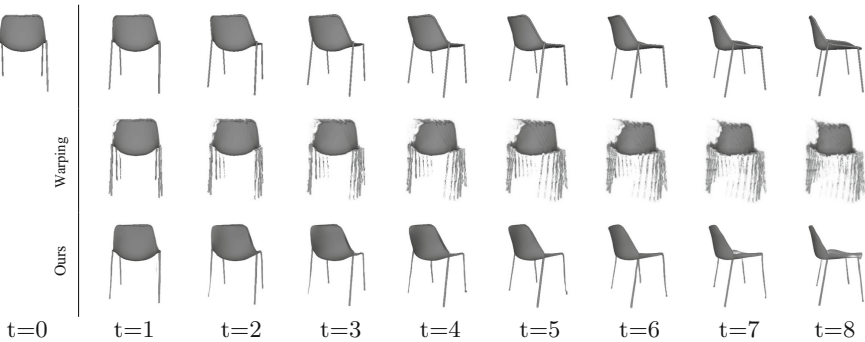

Fig. 4. Comparisons between our *Flow2rgb* model and warping operation, given the first frame and all precomputed flows (between adjacent ground truth frames). Starting from the first frame and first flow, we iteratively run warping or the proposed *Flow2rgb* model based on the previous result and next flow to obtain the sequence. Top: ground truth, Middle: warping results, Bottom: our results.

The architecture of the proposed frame generation model *Flow2rgb* is shown in Fig. 2(right). Given the input x_t and its optical flow f_t that represents the

motion of next time step, the network is trained to generate the next frame x_{t+1}. Since two adjacent frames often share similar information (especially in the static background regions), in order to let the network focus on learning the difference of two frames, we first warp the x_t based on the flow to get a coarse estimation $\tilde{x}_{t+1}$. Then we design a Siamese-like [35] network with the warped frame and the flow as two streams of input. The frame and flow encoders (blue and green blocks) borrow the same architecture of the VGG-19 up to the Relu_4_1 layer, and the decoder (yellow blocks) is designed as being symmetrical to the encoder with the nearest neighbor upsampling layer used for enlarging feature maps. We train the model using a pixel reconstruction loss and a feature loss [36,37] as shown below:

$$\mathcal{L} = \|\hat{x}_{t+1} - x_{t+1}\|_2 + \sum_{K=1}^{5} \lambda \|\Phi_K(\hat{x}_{t+1}) - \Phi_K(x_{t+1})\|_2 \ , \tag{2}$$

where $\hat{x}_{t+1}$, x_{t+1} are the network output and ground truth (GT), and Φ_K is the VGG-19 [38] encoder that extracts the Relu_K_1 features. λ is the weight to balance the two losses. This model is learned in an unsupervised manner without human labels. Note that this is a one-step flow-to-frame model. Since we predict multi-step flows in the flow prediction stage, starting with the first given frame, we iteratively run this model to generate the following frame based on the next flow and previous generated frame.
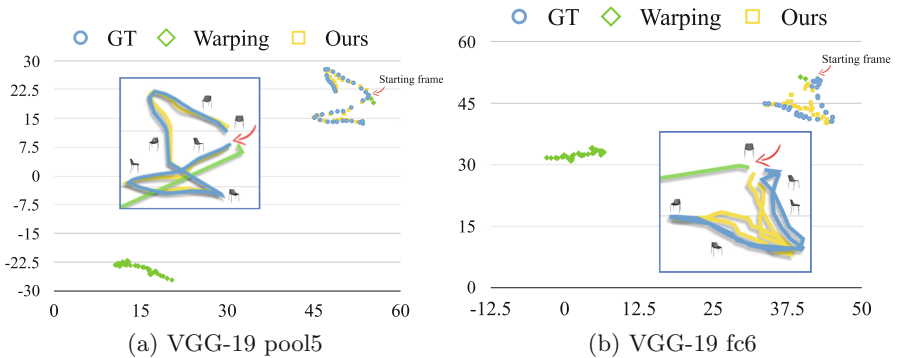

(a) VGG-19 pool5

(b) VGG-19 fc6

Fig. 5. Visualization of sequence (a chair turning around) manifold in deep feature space. Staring from the same frame, each predicted frame of three sequences is visualized as a 2-D point by applying t-SNE [39] on its deep features. The moving average is shown as lines to imply the shape (or trending) of the manifold. For example in (a), the GT rotating chair (blue) follows a "8" like manifold in pool5 feature space, which our predicted sequence (yellow) follows closely but the warping sequence (green) deviates much further. (Color figure online)

We show the effectiveness of our *Flow2rgb* model in Fig. 4 with an example of chair rotating sequence [40]. To verify the frame generation phase alone, we

assume that the flows are already available (computed by [41]). Then given the first frame and future flows, the second row of Fig. 4 shows the iterative warping results where the chair legs are repeatedly copied in future frames as the warping is unable to depict the right appearance of chair in difference views. In contrast, our model iteratively generates the occluded parts and removed unnecessary parts in the previous frame according to the flow at each time step. As claimed in [40], the deep embeddings of objects under consecutively changing views often follow certain manifold in feature space. If we interpret this changing view as a type of rotating motion, our predicted results for different views also needs to stay close to the manifold shape of the GT sequence. We demonstrate this by extracting the VGG-19 [38] features of each predicted frame, mapping it to a 2-D point through t-SNE [39], and visualizing it in Fig. 5. It clearly shows that our predictions follows closely with the manifold of the GT sequence, while the warping drives the predictions to deviate from the GT further and further.

4   Experimental Results

In this section, we first discuss the experimental settings and implementation details. We then present qualitative and quantitative comparisons between the proposed algorithm and several competing algorithms. Finally, we analyze the diversity issue in uncertainty modeling.

Datasets. We mainly evaluate our algorithm on three datasets. The first one is the KTH dataset [42] which is a human action video dataset that consists of six types of action and totally 600 videos. It represents the movement of articulated objects. Same as in [4,5], we use person 1–16 for training and 17–25 for testing. We also collect another two datasets from online websites, i.e., the *WavingFlag* and *FloatingCloud*. These two datasets represents dynamic texture videos where motions may bring the shape changes on dynamic patterns. The *WavingFlag* dataset contains 341 videos of 80K+ frames and the *FloatingCloud* dataset has 415 videos of 150K+ frames in total. In each dataset, we randomly split all videos into the training (4/5) and testing (1/5) set.

Implementation Details. Given the starting frame x_0, our algorithm predicts the future in next $M = 16$ time steps. Each frame is resized to 128×128 in experiments. Similar to [14,43], we employ an existing optical flow estimator SPyNet [41] to obtain flows between GT frames for training the 3D-cVAE. As described in Sect. 3.1, we stack x_0 with each flow map f_i in F. Thus during the training, the input cube to the 3D-cVAE is of size $16 \times 5 \times 128 \times 128$ where $5 = 2 + 3$ (2-channel flow and 3-channel RGB). The dimension of the latent variable z in the bottle neck is set as 2000. Another important factor for a successful network training is to normalize the flow roughly to $(0, 1)$ before feeding it into the network, ensuring pixel values of both flows and RGB frames are within the similar range. Since the *Flow2rgb* model can be an independent module for motion transfer with known flows, we train the 3D-cVAE and *Flow2rgb* model separately in experiments.
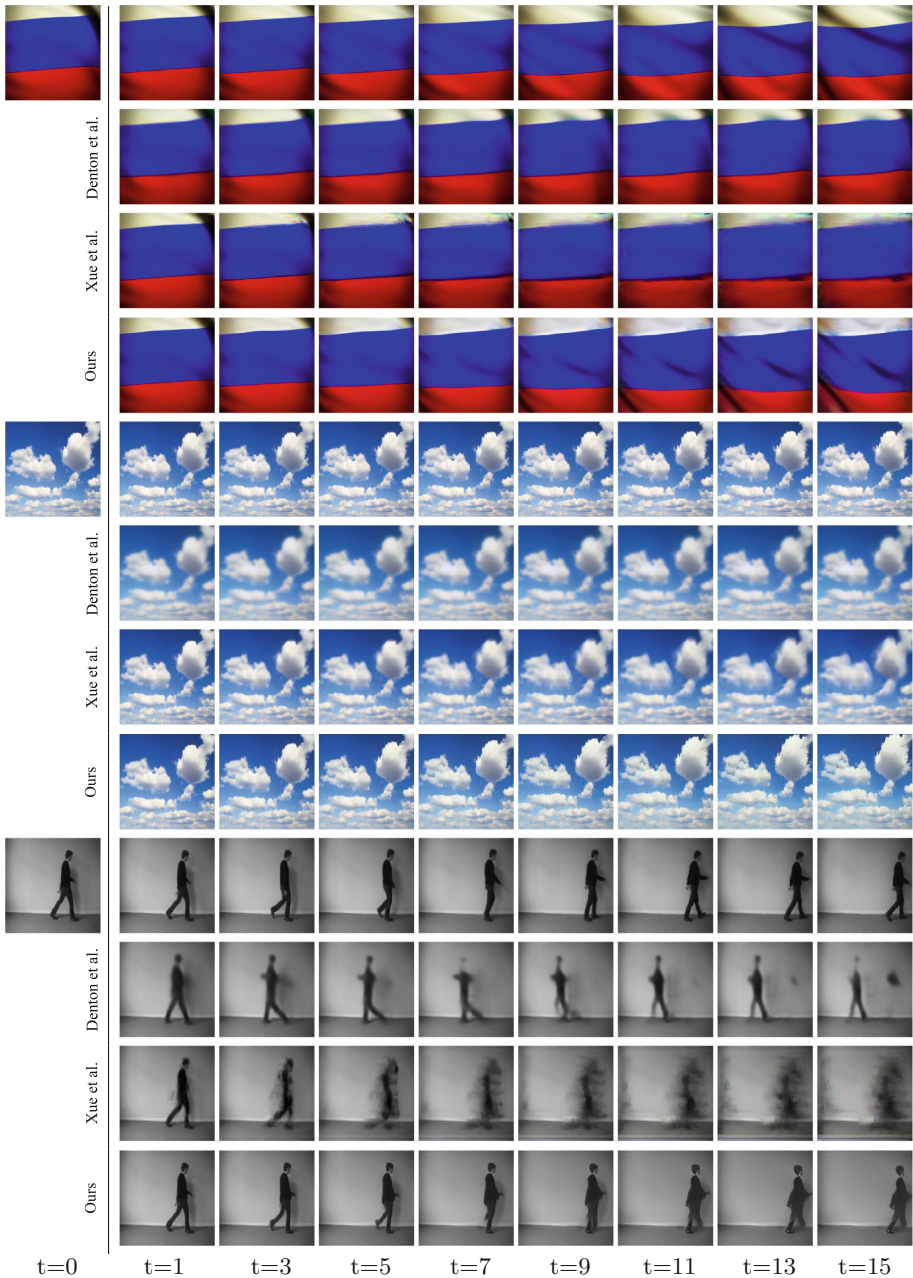
Fig. 6. Visual comparisons of different prediction algorithms. Top left: the starting frame. From top to bottom in example: GT, Denton et al. [5], Xue et al. [8], Ours. The GT sequence provides a sense of motion rightness, while the predicted sequence is unnecessary to be exactly the same with GT.

Evaluations. Different prediction algorithms have their unique settings and assumptions. For example, Mathieu et al. [2] requires four frames stacked together as the input. Villegas et al. [4] ask for feeding the image difference (at least two frames). Their following work [25], though based on one frame, additionally needs multiple historical human pose maps to start the prediction. For fair comparisons, we mainly select prediction methods [5,8] that accept one single image as the only input to compare. The work of [5] represents the typical recursive prediction pipeline, which builds upon a fully-connected long short-term memory (FC-LSTM) layer for predictions. Their model is originally trained and tested by observing multiple frames. Here we change their setting to one-frame observance in order to be consistent with our setting. The work of [8] is the typical one-step prediction method based on one given frame. To get multi-frame predictions, we train their model and iteratively test it to get the next prediction based on the previous prediction.

In Fig. 6, we provide a visual comparison between the proposed algorithm and [5,8]. In [5], a pre-trained and disentangled *pose* embedding is employed to keep predicting the pose of the next frame through a FC-LSTM module. For articulated objects, the pose is often compact and in low dimensions, which is relatively easier to handle with a single LSTM module. However, for dynamic textures (e.g., flag, cloud) where all pixels are likely to move, the global pose becomes complex and is no longer a low-dimensional structure representation. Therefore the capacity of recursive models is not enough to capture the spatial and temporal variation trend at the same time. The first two examples in Fig. 6 show that the flag and cloud in predicted frames are nearly static. Meanwhile, the pose only describes the static structure of the object in the current frame and cannot tell as much information as the flow about the next-step motion. In the third example of Fig. 6, it is obvious that the human is walking to the right. But the results of [5] show that the human is going in a reverse direction. Moreover, since they directly predict frame pixels and use the reconstruction loss only, their results are relatively blurry. In [8], as they only predict the next one frame, the motion is often clear in the second frame. But after we keep predicting the following frame using the previous predicted frame, the motion gradually disappears and the quality of results degrades fast during a few steps. Moreover, they choose to predict the image difference which only shows global image changes but does not capture how each pixel will move to its corresponding one in the next frame. In contrast, our results show more continuous and reasonable motion, reflected by better generated full frames. For example, in the first flag example, the starting frame indicates that the fold on top right will disappear and the fold at bottom left will bring bigger folds. Our predicted sequence presents the similar dynamics as what happens in the GT sequence, which makes it look more realistic.

We also quantitatively evaluate these prediction methods using three different metrics, i.e., the root-mean-square error (RMSE), perceptual similarity [44], and user preference. The RMSE is the classic per-pixel metric which measures the spatial correspondence without considering any high-level semantics and is often
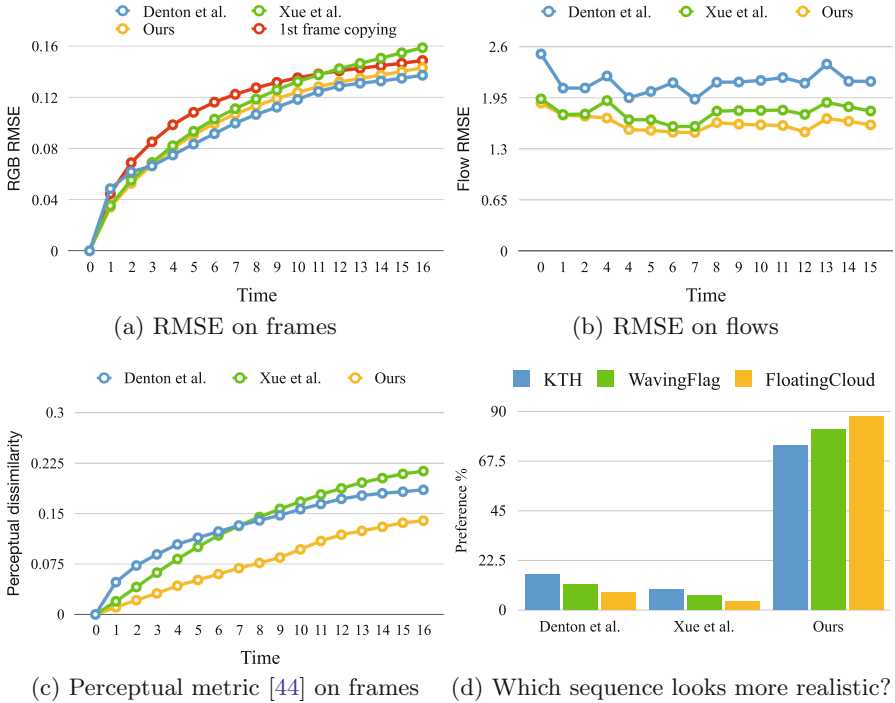
(a) RMSE on frames

(b) RMSE on flows

(c) Perceptual metric [44] on frames

(d) Which sequence looks more realistic?

Fig. 7. Quantitative evaluations of different prediction algorithms. We start from the per-pixel metrics (e.g., RMSE) and gradually take human perception into consideration. Our method achieves the best performance under metrics (b)–(d).

easily favored by smooth results. Based on this observation, the recent work of [44] proposes a perceptual similarity metric by using deep network embeddings. It is demonstrated to agree with human perceptions better. Lastly, we directly ask the feedback from users by conducting user studies to understand their preference towards the predicted results by different algorithms.

We start with the traditional RMSE to compute the difference between predicted sequence and GT sequence frame-by-frame and show the result in Fig. 7(a). To understand how effective these prediction methods are, we design a simple baseline by copying the given frame as multi-step predictions. However, we do not observe obvious difference among all these methods. While the prediction from one single image is originally ambiguous, the GT sequence can be regarded as just one possibility of the future. The trending of motion may be similar but the resulted images can be significantly different in pixel-level. But the RMSE metric is actually very sensitive to the pixel spatial mismatch. Similar observations are also found in [5,44]. That is why all these methods, when comparing with the GT sequence, shows the similar RMSE results. Therefore, instead of measuring the RMSE on frames, we turn to measure the RMSE on optical flows because the optical flow represents whether the motion field is pre-

dicted similarly or not. We compute the flow maps between adjacent frames of the GT sequence and other predicted sequences using the SPyNet [41] and show the RMSE results in Fig. 7(b). Now the difference becomes more clear and our method achieves the lowest RMSE results, meaning that our prediction is the closest to the GT in terms of the predicted motions.

However, the evaluation of prediction results still need to take human perception into consideration in order to determine whether sequences look as realistic as the GT sequence. Therefore we turn to the perceptual similarity metric [44]. We use the Alex-Net [45] for feature extraction and measure the similarity between predicted sequence and GT sequence frame-by-frame. Since this metric is obtained by computing feature distances, we denote it as perceptual dissimilarity so that small values means being more similar. The results in Fig. 7(c) show that the proposed method outperforms other algorithms with an even larger margin than that in Fig. 7(b), which means that the predicted sequence of our method is perceptually more similar to the GT sequence.

Finally, we conduct the user study to get the feedback from human subjects on judging different predicted results. We prepare 30 starting frames (10 from each dataset) and generated 30 sequences (16-frame) for each method. For each subject, we randomly select 15 sets of sequences predicted by three methods. For each starting frame, the three predicted sequences are displayed side-by-side in random order. Each subject is asked to vote one sequence that looks most realistic for each starting frame. We finally collect 900 votes from 60 users and report the results (in percentage) in Fig. 7(d). The study results clearly show that the proposed method receives the most votes for more realistic predictions among all three categories. Both Fig. 7(c) and (d) indicate that the proposed method performs favorably against [5,8] in terms of perceptual quality.
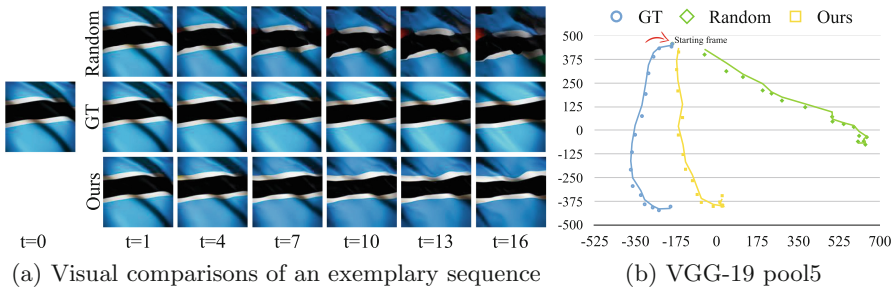

(a) Visual comparisons of an exemplary sequence   (b) VGG-19 pool5

Fig. 8. Comparison with a naive baseline which transfers a random motion field. (b) The GT sequence follows a "C" like manifold in pool5 feature space, which our prediction follows closely but the random prediction deviates much further.

Random Motion. We also compare with a naive approach which uses random flow maps (e.g., sampling from the Gaussian distribution $N(0, 2)$ for each pixel). We apply the proposed *flow2rgb* model to both random and the learned motions

by our method to generate frames. Figure 8(a) shows one example. In Fig. 8(b), we visualize the manifold of predicted sequences in the deep feature space using the t-SNE scheme (as did in Fig. 5). Both demonstrate that the learned motion generates much better results than those by the random motion, as the naive approach neither models the motion distribution nor considers the temporal relationship between frames.
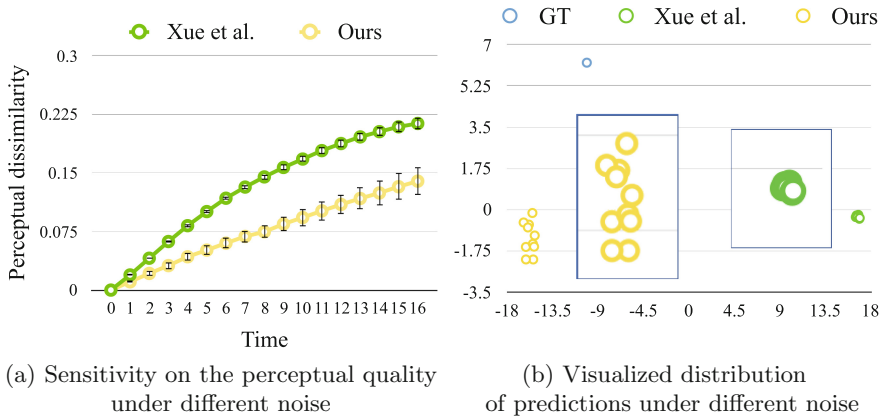

(a) Sensitivity on the perceptual quality under different noise

(b) Visualized distribution of predictions under different noise

Fig. 9. Comparisons between [8] and the proposed algorithm on uncertainty modeling given the same starting frame. By drawing different samples, the generated predictions by our method exhibits more diversities while still being more similar to GT.

Diversity. Both [8] and the proposed method model the uncertainty in predictions, but are different in one-step [8] or multi-step uncertainties. By drawing different samples, we evaluate how the quality of predictions is affected by the noise input and how diverse the predicted sequences are. While [8] uses a noise vector of 3200 dimensions and we use that of 2000 dimensions, the noise inputs of two models are not exactly the same but they are all sampled from $N(0, 1)$. We sample 10 noise inputs for each method, while ensuring that the two sets of noise inputs have the similar mean and standard deviation. Then we obtain 10 sequences for each method, and compare them with the GT sequence. Figure 9(a) shows the mean and standard deviation of the perceptual metric over each method's 10 predictions when compared with the GT frame-by-frame. Under different noise inputs, our method keeps generating better sequences that are more similar to the GT. Meanwhile, the results of our algorithm show larger deviation, which implies that there are more diversities in our predictions. To further verify this, we show the embeddings of generated sequences in Fig. 9(b). For each sequence, we extract the VGG-19 [38] features (e.g., fc6 layer) of each frame, stack them as one vector, and map it to a 2-D point through t-SNE [39]. Figure 9(b) shows that our 10 predictions are much closer to the GT sequence while being scattered to be different from each other. In contrast, the 10 predictions of [8] huddle together and are far from the GT. Those comparisons
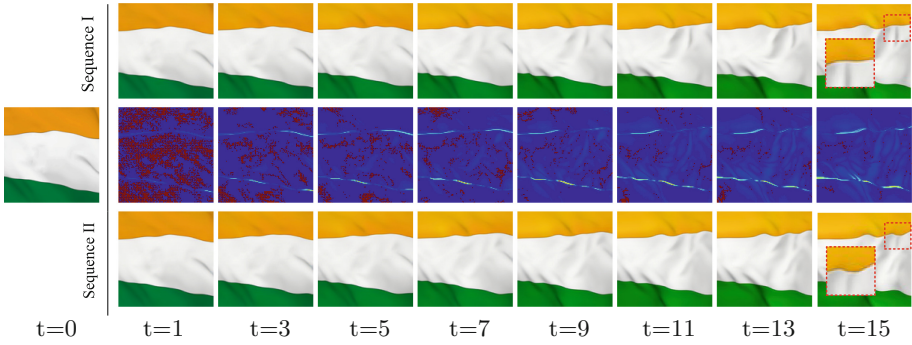
Fig. 10. Given a still image, by sampling different noise in the latent space, our algorithm synthesizes different future outcomes to account for the intrinsic uncertainties. In the middle row, we show the difference of two generated sequences frame-by-frame.

demonstrate that the proposed algorithm generates more realistic and diverse future predictions. Figure 10 shows an example of two predicted sequences.

Fig. 11. Potential application of our algorithm in video editing.

Bringing Still Images to Life. Unlike previous video prediction methods [4,7,25] that mainly focus on humans for action recognition, our algorithm is more general towards bringing elements in the still image to life, i.e., turning a still image into a vivid GIF for aesthetic effects. It can be an effective tool for video editing.

In Fig. 11, we show a example of turning a photo into a vivid sequence. We mask out the sky region, apply our model trained on the *FloatingCloud* dataset and generate the effect of clouds floating in the sky. This could further benefit existing sky editing methods [46]. Moreover, if we replace our flow prediction with known flows from a reference sequence, our flow-to-frame model *Flow2rgb* becomes a global motion style transfer model. As the current random sampling strategy for flow predictions is uncontrollable, future work may include introducing more interactions from users to control detailed motions.

5   Conclusions

In this work, we propose a video prediction algorithm that synthesizes a set of likely future frames in multiple time steps from one single still image.

Instead of directly estimating the high-dimensional future frame space, we choose to decompose this task into a flow prediction phase and a flow-grounded frame generation phase. The flow prediction models the future uncertainty and spatial-temporal relationship in a 3D-cVAE model. The frame generation step helps prevent the manifold shape of predicted sequences from straying off the manifold of real sequences. We demonstrate the effectiveness of the proposed algorithm on both human action videos and dynamic texture videos.

References

1. Ekman, M., Kok, P., de Lange, F.P.: Time-compressed preplay of anticipated events in human primary visual cortex. Nat. Commun. **8**, 15276 (2017)
2. Mathieu, M., Couprie, C., LeCun, Y.: Deep multi-scale video prediction beyond mean square error. In: ICLR (2016)
3. 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: NIPS (2015)
4. Villegas, R., Yang, J., Hong, S., Lin, X., Lee, H.: Decomposing motion and content for natural video sequence prediction. In: ICLR (2017)
5. Denton, E., Birodkar, V.: Unsupervised learning of disentangled representations from video. In: NIPS (2017)
6. Tran, D., Bourdev, L., Fergus, R., Torresani, L., Paluri, M.: Learning spatiotemporal features with 3D convolutional networks. In: ICCV (2015)
7. Walker, J., Doersch, C., Gupta, A., Hebert, M.: An uncertain future: forecasting from static images using variational autoencoders. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9911, pp. 835–851. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46478-7_51
8. Xue, T., Wu, J., Bouman, K., Freeman, B.: Visual dynamics: probabilistic future frame synthesis via cross convolutional networks. In: NIPS (2016)
9. Yuen, J., Torralba, A.: A data-driven approach for event prediction. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010. LNCS, vol. 6312, pp. 707–720. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-15552-9_51
10. Lan, T., Chen, T.-C., Savarese, S.: A hierarchical representation for future action prediction. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8691, pp. 689–704. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10578-9_45
11. Hoai, M., De la Torre, F.: Max-margin early event detectors. IJCV **107**(2), 191–202 (2014)
12. Kitani, K.M., Ziebart, B.D., Bagnell, J.A., Hebert, M.: Activity forecasting. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012. LNCS, vol. 7575, pp. 201–214. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-33765-9_15
13. Vondrick, C., Pirsiavash, H., Torralba, A.: Anticipating visual representations from unlabeled video. In: CVPR (2016)

14. Walker, J., Gupta, A., Hebert, M.: Dense optical flow prediction from a static image. In: ICCV (2015)
15. Srivastava, N., Mansimov, E., Salakhudinov, R.: Unsupervised learning of video representations using LSTMS. In: ICML (2015)
16. Oh, J., Guo, X., Lee, H., Lewis, R.L., Singh, S.: Action-conditional video prediction using deep networks in atari games. In: NIPS (2015)
17. Babaeizadeh, M., Finn, C., Erhan, D., Campbell, R.H., Levine, S.: Stochastic variational video prediction. In: ICLR (2018)
18. Finn, C., Levine, S.: Deep visual foresight for planning robot motion. In: ICRA (2017)
19. Vondrick, C., Pirsiavash, H., Torralba, A.: Generating videos with scene dynamics. In: NIPS (2016)
20. Liang, X., Lee, L., Dai, W., Xing, E.P.: Dual motion GAN for future-flow embedded video prediction. In: ICCV (2017)
21. Tulyakov, S., Liu, M.Y., Yang, X., Kautz, J.: Mocogan: decomposing motion and content for video generation. arXiv preprint arXiv:1707.04993 (2017)
22. Finn, C., Goodfellow, I., Levine, S.: Unsupervised learning for physical interaction through video prediction. In: NIPS (2016)
23. Vondrick, C., Torralba, A.: Generating the future with adversarial transformers. In: CVPR (2017)
24. Chao, Y.W., Yang, J., Price, B., Cohen, S., Deng, J.: Forecasting human dynamics from static images. In: CVPR (2017)
25. Villegas, R., Yang, J., Zou, Y., Sohn, S., Lin, X., Lee, H.: Learning to generate long-term future via hierarchical prediction. In: ICML (2017)
26. Doretto, G., Chiuso, A., Wu, Y.N., Soatto, S.: Dynamic textures. IJCV **51**(2), 91–109 (2003)
27. Yuan, L., Wen, F., Liu, C., Shum, H.-Y.: Synthesizing dynamic texture with closed-loop linear dynamic system. In: Pajdla, T., Matas, J. (eds.) ECCV 2004. LNCS, vol. 3022, pp. 603–616. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-24671-8_48
28. Xie, J., Zhu, S.C., Wu, Y.N.: Synthesizing dynamic patterns by spatial-temporal generative convnet. In: CVPR (2017)
29. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. In: ICLR (2014)
30. Sohn, K., Lee, H., Yan, X.: Learning structured output representation using deep conditional generative models. In: NIPS (2015)
31. Reed, S.E., Zhang, Y., Zhang, Y., Lee, H.: Deep visual analogy-making. In: NIPS (2015)
32. Zhou, T., Tulsiani, S., Sun, W., Malik, J., Efros, A.A.: View synthesis by appearance flow. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9908, pp. 286–301. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46493-0_18
33. Liu, Z., Yeh, R., Tang, X., Liu, Y., Agarwala, A.: Video frame synthesis using deep voxel flow. In: ICCV (2017)
34. Park, E., Yang, J., Yumer, E., Ceylan, D., Berg, A.C.: Transformation-grounded image generation network for novel 3D view synthesis. In: CVPR (2017)
35. Chopra, S., Hadsell, R., LeCun, Y.: Learning a similarity metric discriminatively, with application to face verification. In: CVPR (2005)
36. Johnson, J., Alahi, A., Fei-Fei, L.: Perceptual losses for real-time style transfer and super-resolution. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9906, pp. 694–711. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46475-6_43

37. Dosovitskiy, A., Brox, T.: Generating images with perceptual similarity metrics based on deep networks. In: NIPS (2016)
38. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: ICLR (2015)
39. van der Maaten, L., Hinton, G.: Visualizing data using t-SNE. JMLR **9**(Nov), 2579–2605 (2008)
40. Wu, Z., et al.: 3D shapenets: a deep representation for volumetric shapes. In: CVPR (2015)
41. Ranjan, A., Black, M.J.: Optical flow estimation using a spatial pyramid network. In: CVPR (2017)
42. Schuldt, C., Laptev, I., Caputo, B.: Recognizing human actions: a local SVM approach. In: ICPR (2004)
43. Gao, R., Xiong, B., Grauman, K.: Im2Flow: motion hallucination from static images for action recognition. arXiv preprint arXiv:1712.04109 (2017)
44. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep networks as a perceptual metric. In: CVPR (2018)
45. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: NIPS (2012)
46. Tsai, Y.H., Shen, X., Lin, Z., Sunkavalli, K., Yang, M.H.: Sky is not the limit: semantic-aware sky replacement. ACM Trans. Graph. **35**(4), 149–159 (2016)