This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2018.2882061, IEEE Transactions on Circuits and Systems for Video Technology

Predicting Diverse Future Frames with Local Transformation-Guided Masking

Jinzhuo Wang, Wenmin Wang*, *Member, IEEE*, Wen Gao, *Fellow, IEEE*

***Abstract*—Video prediction is the challenging task of generating the future frames of a video given a sequence of previously observed frames. This task involves constructing of an internal representation that accurately models the frame evolutions, including contents and dynamics. Video prediction is considered difficult due to the inherent compounding of errors in recursive pixel level prediction. In this work, we present a novel video prediction system that focuses on regions of interest (ROIs) rather than the entire frames and learns frame evolutions at the transformation level rather than at the pixel level. We provide two strategies to generate high-quality ROIs that contains potential moving visual cues. The frame evolutions are modeled with a transformation generator that produces transformers and masks simultaneously, which are then combined to generate the future frame in a transformation-guided masking procedure. Compared with recent approaches, our system is able to generate more accurate predictions by modeling the visual evolutions at the transformation level rather than at the pixel level. Focusing on ROIs avoids heavy computational burden and enables our system to generate high-quality long-term future frames without severely amplified signal loss. Moreover, our system is able to generate diverse plausible future frames, which is important in many real-world scenarios. Furthermore, we enable our system to perform video prediction conditioned on a single frame by revising the transformation generator to produce motion-centric transformers. We test our system on four datasets with different experimental settings and demonstrate its advantages over recent methods both quantitatively and qualitatively.**

***Keywords*—Video prediction, diverse future frames, local transformation level, transformation-guided masking, region of interest, video prediction on single frame**

I. Introduction

GIVEN the considerable progress in video recognition [1] [2] [3] [4] [5] [6] [7] [8], prediction has become an essential module for intelligent agents to plan actions or to make decisions in real-world videos [9] [10] [11] [12]. This paper considers the task of video prediction, where the goal is to generate future frames of a video based on the sequential frames that have already been observed. Video prediction has broad prospects in real-world scenarios, such as robots planning, autonomous driving, and anomaly detection in surveillance videos. Learning to predict future frames from a video sequence involves constructing an internal representation that models frame evolutions accurately, including contents and dynamics. However, predicting realistic and sharp future frames is a challenging task given the high dimensionality of the data, the complexity and ambiguity inherent in videos, and the complex dynamics of the environment [13].

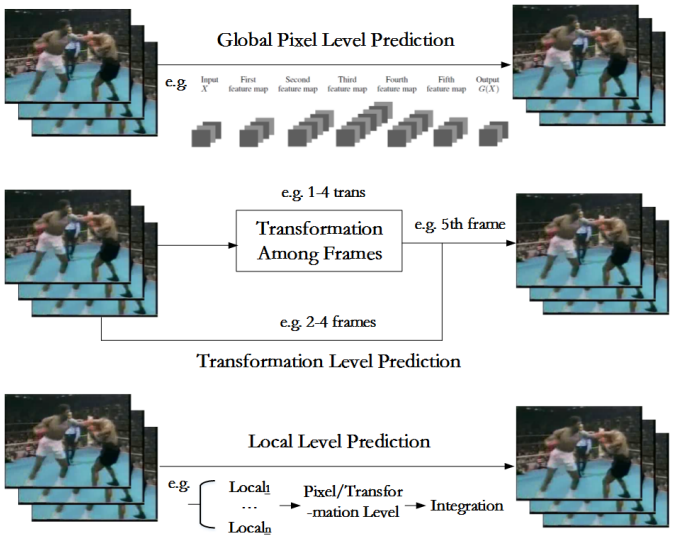

Fig. 1. Video prediction at different visual levels, including global and local levels, raw pixel level and our proposed transformation level. Our proposed video prediction system is conducted at the local transformation level, which is able to accurately model frame evolutions and perform long-term prediction.

Traditional approaches for video prediction have used independent component analysis, slow feature analysis, Boltzmann machines, and Lie group theory. However, these techniques are typically not suitable for scaling to high-resolution videos, and they are not flexible enough for capturing the complexity of real-world data. Another line of methods utilize optical flow to model frame evolutions, which requires accurately capturing the optical field [14] [15]. These methods are computationally expensive, and more importantly, optical flow estimation has inherent difficulties in accurately generating an entire frame. This situation is severe when multiple future frames are needed due to the amplified loss in recursive prediction.

Recent competitive approaches are often performed at the pixel level and treat video prediction as a regression problem with machine learning techniques, particularly techniques equipped with successful deep neural networks [16] [17] [13] [18] [9] [10] [11] [12]. However, in practice, these methods often generate blurry predictions, especially in long-term future frames. We attribute this result to two main reasons. First, in most cases, making reasonable long-term frame predictions in natural videos highly depends on observing the generated frames in the past to make predictions further into the future.

Jinzhuo Wang, Wenmin Wang and Wen Gao are with the Department of Electrical and Computer Engineering, Peking University, China. Corresponding author is Wenmin Wang. E-mail: wangwm@ece.pku.edu.cn

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2018.2882061, IEEE Transactions on Circuits and Systems for Video Technology

Pixel-level prediction makes it difficult for the models to learn powerful internal representations because these approaches need to be highly robust to pixel-level noise. This situation becomes severe when the noise amplifies quickly through time until it overwhelms the signal with which we are concerned. Commonly, the first few prediction steps are of decent quality, but then the prediction dramatically degrades until all the video context is lost [13] [19]. Second, previous works often consider frame evolutions at the global level. This setting brings computational complexity especially in real high-resolution videos. Besides, in most natural videos, motion signals are dense in certain regions rather than being distributed over the entire frame. Focusing on the entire frame brings computation burden; moreover, it is not able to explicitly distinguish multiple objects in a particular scene and extract an internal object-centric representation. This may result in distinct objects in the same scene being subject to the same motion [20].

In this paper, we propose a novel video prediction system that achieves predicting future frames at the local transformation level. A schematic comparison of video prediction works performed at different levels is presented in Fig. 1. This setting helps us prevent the inherent compounding of errors in recursive global pixel level prediction, especially when we need to conduct iterative prediction for long-term future frames based on previously generated frames. We show how to focus on high-quality regions of interest (ROIs) with two approaches (pyramid sampling and spatial-transformed learning). The transformations for ROIs are learned with an auto-encoder structure that is responsible for modeling their visual evolutions by producing the transformers and masks, which are then prepared for transformation-guided masking procedure to generate future frames. Our transformation generator is governed by a recurrent neural network (RNN), which is used to integrate the transformation patterns of each ROI and determine the final choices of future frames.

To generate long-term future frames, we propose two strategies: the first is to learn multiple stacked transformers and masks to directly generate long-term future frames, and the second is to conduct recursive prediction for each future frame. Experiments suggest that both strategies can yield promising results, thus demonstrating the advantages of our method because most solutions performed at the global pixel level can hardly generate high-quality long-term future frames in a recursive manner. We impart our system with the ability to generate diverse future frames with different transformation generators through sampling latent variables combined with input visual cues. Moreover, our system is able to generate future frames from a single frame, which is very difficult. We achieve this function by revising the transformation generator architecture with the ability to produce motion-centric transformations, which can be used to form diverse multiple frames with a single frame. Our system is an integrated and flexible video prediction solution. We examine it with different experimental settings on four datasets and demonstrate the advantages over recent competitive methods both quantitatively and qualitatively.

The remaining content is organized as follows. Section II reviews related works and discuss the relations to our system. Section III describes our video prediction system including its key modules, future frame generation method, training strategy and how to perform video prediction on single frame. The experimental results and analysis are given in Section IV. Finally, Section V concludes this paper.

II. Related Works

Video prediction is a subtask of video generation since it can be viewed as video generation conditioned on previously observed frames. However, video generation is more general and can be conditioned on other types of data, even noise. Meanwhile, video prediction has a true label that the predicted frames should be close to in certain evaluation measurements, whereas video generation often requires only the plausibility of the generated video sequence. In this section, we mainly review recent progress on video prediction. We also cover some works of "learning diversity under uncertainty", which is the situation in our system that predicts diverse future frames of multiple possibilities. There are other types of video prediction works that focus on certain contents in videos (e.g. optical flow, motion, human pose, and semantic activity), which are beyond the scope of our paper and thus not covered.

Early work on video prediction focused on small patches containing simple predictable motions [21] [22] [23] and motions in real videos [16] [18]. High-resolution videos contain substantially more complicated motion that cannot be modeled in a patch-wise manner due to the well-known aperture problem, which causes blockiness in predictions as we move forward in time. [16] attempted to overcome blockiness by averaging over spatial displacements after predicting patches. However, this approach does not work for long-term predictions. Recent approaches in video prediction have moved from predicting patches to full frame prediction. Recently, a line of studies [11] [13] [24] [25] [26] focused on developing advanced networks to directly generate pixel values. In particular, [17] proposed a network architecture for action conditioned video prediction in Atari games. [13] proposed an adversarial loss for video prediction and a multi-scale network architecture that results in high-quality prediction for a few time steps in natural video. [10] proposed a network architecture to directly transform pixels from the current frame into the next frame by predicting a distribution over pixel motion from previous frames. [25] proposed a probabilistic model for predicting the possible motions of a single input frame by training a motion encoder in a variational auto-encoder approach. [24] constructed a model that generates realistic-looking video by separating background and foreground motion. [12] improved the convolutional auto-encoder architecture by separating motion and content features. [11] built an architecture inspired by the predictive coding concept in the neuroscience literature that predicts realistic looking frames. However, these methods often produce blurry predictions since modeling the complex pixel-level distributions of natural images is difficult. Several approaches [27] [28] [29] alleviated this blurring problem by resorting to motion field prediction for copying pixels from previous frames. [30] learned to explicitly enforce future frame predictions to be consistent with the pixel-wise flows in the video through a dual-learning mechanism.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2018.2882061, IEEE Transactions on Circuits and Systems for Video Technology

Most of the previously mentioned approaches attempt to perform video generation in a global pixel-to-pixel process. In contrast, we seek to learn frame evolutions at the local transformation level from the past to the future. Prior works have explored learning transformations in restricted domains [10] [25], such as for robotic arms or clip art. We consider different settings and perform long-term diverse predictions. This paper is also related to learning to understand transformations in images and videos [31] [32] [33] [34]. We also study transformations, but we focus on learning the transformations only for ROIs and design a transformation-guided masking procedure to generate future frames. Learning only local transformations helps us produce accurate long-term predictions, even in a recursive manner.

Learning Diversity Under Uncertainty. Deep latent variable models such as generative neural nets (GANs) and variational auto-encoders (VAEs) can be used to handle the inherent uncertainty in future prediction tasks [35]. An interesting approach that uses GANs for unsupervised image representation learning was simultaneously proposed in [36] and [37], where the generative model is trained along with an inference model that maps images to their latent representations. [24] used a two-stream generative model: one stream generates a static background, while the other stream generates a dynamic foreground sequence that is pasted on the background. [38] used similar ideas to develop an iterative image generation model where objects are sequentially pasted on the image canvas using a recurrent GAN. [25] predicted future frames from a single frame based on VAEs. Similarly, [39] performed video forecasting with VAEs, predicting feature point trajectories from still images. Concurrent works from [25] and [10] applied the learned kernels on input images to produce diverse futures. In contrast, [14] predicted where pixels will move using direct optical flow supervision from a single image. Our system is also able to predict future video frames from a single frame by revising the transformation generator architecture with the ability to produce motion-centric transformations. We achieve diversity generation by diversifying the transformation generator through sampling latent variables, which receives contributions from the specific distribution and input frame sequence.

III. The Proposed Video Prediction System

In this section, we introduce our video prediction system. We first present a system overview, and then we detail the key modules in our system and describe their relations and interactions. We show how to generate transformations and perform transformation-guided masking to generate future frames. Finally, we present the training methods of our system and demonstrate its ability to predict future frames conditioned on a single frame.

A. Overview

The proposed system is shown in Fig. 2. Given an input frame sequence (single frame situation is also allowed in our system, as addressed in Section III-F), the goal of our system is to predict diverse plausible future frames. We first perform a process of generating ROIs generation that produces several spatiotemporal regions containing potential motions. These ROIs are sequentially sent to the transformation generator, which is govern by an RNN. It is responsible for generating transformers and masks. In the next stage, the learned transformers and the corresponding ROIs are sent to a merge network to form transformed ROIs, which are prepared to be combined with the learned masks to generate a candidate of the next frame.

To produce multiple long-term future frames, the newly generated frame is re-sent backwards in a recurrent manner to apply the system recursively to synthesize further frames. Because we model the pattern of input frame sequence at the transformation level, the reconstruction loss will not amplify in the procedure of long-term prediction. Another strategy is to directly generate multiple sets of transformations to produce long-term predictions at once.

Meanwhile, generating diverse future frames is achieved with designed conditional codes applied to generate diverse transformations. The conditional codes are learned from a specific distribution (e.g., Gaussian distribution) along with the input visual information of the original input frame sequences.

Finally, our system has the ability to predict diverse future frames using one single frame, which is very difficult because motion information that is easily mined in sequential frames is not provided in a single frame. The key idea behind our solution is to adapt the merge network to perform volumetric convolutions with a single frame to form diverse frame sequences. In the following, we detail the key modules of our system, describe how to perform video prediction, and show the training methods.

B. ROI Generation

Due to the inherent compounding of errors in recursive pixel-level prediction, we choose to perform video prediction at the transformation level. However, learning the transformation of the entire frame sequence is difficult. We prefer to concentrate our transformation learning in ROIs where motions have a high probability of occurrence. To generate high-quality ROIs, we introduce two methods: a pyramid sampling method and spatial-transform learning method.

Pyramid Sampling. Inspired by spatial pyramid pooling [40] and its recent variants [41] for visual recognition, we introduce a pyramid sampling method to detect the ROIs that may contain key objects with crucial motion information. As shown in Fig 3, at the first level of the pyramid, each input frame in the sequence is split into $S \times S$ bins (bins can overlap with adjacent bins by specifying a stride). All 2×2 bins from the first level are then combined as a larger bin at the second level. In this way, the top pyramid level has a single bin that covers the entire input frames. The magnitude of the motion information for the ith bin at level l can then be defined as f_i^l. The T bins with top scores (using f_i^l as score) will be selected as the ROIs. Here, we use the l_2 differences between the voxels of all frame patches in the same bin to evaluate f_i^l (take the average for grids in the bin), which outperforms conventional descriptors such as histogram of gradients (HOG) according to our experiments.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2018.2882061, IEEE Transactions on Circuits and Systems for Video Technology

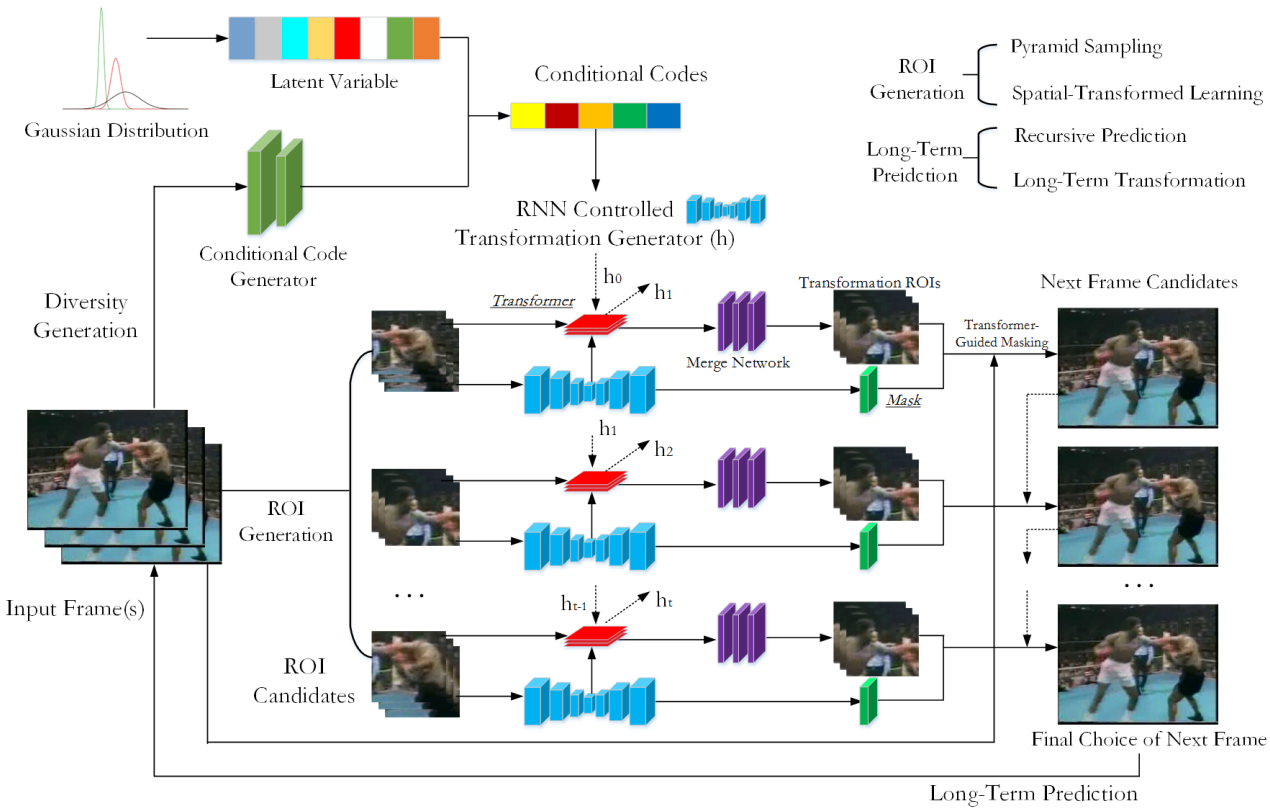

Fig. 2. Illustration of the proposed video prediction system. The input frame sequence is first sent to the ROI generator to obtain several ROIs, which sequentially act as input to the generator that is responsible for generating transformers and masks. The transformers model the visual evolution of each ROI while the masks preserve the generation rules. These two parts are combined with the original ROI to generate a candidate of the next frame. The final choice is determined when the transformation generator processes all ROIs and stops, governed by an RNN controller. To produce multiple long-term future frames, the first solution is to directly generate multiple frames with multiple sets of transformations, and the second is recursive implementation performed on each future frame. Both are tested and shown to yield promising results. Diverse generation is achieved with designed conditional codes applied to generate diverse transformations and masks. The conditional codes are learned from specific distribution along with visual cues of the input frame sequence. Our system is also able to predict future frames using a single frame, where the key idea is to revise the transformation generator to produce motion-centric transformers, equipped with the first long-term prediction method as discussed in Section III-F.

The most prominent advantage of the pyramid attention method is that the key object with dense motion information around it may be selected in multiple bins of different levels. This allows the attention controller to focus on the same object at multiple scales, which can help to learn a more representative transformation. Moreover, pyramid attention provides a more flexible approach to most existing attention methods that choose fixed scales manually [42].

Spatial-Transform Learning. The other attention mechanism that we use is inspired by recently proposed spatial transformer [32] [43], which is a powerful and general method that can provide invariance to the shapes and sizes of objects in the image. The spatial transformer (ST) operates on an arbitrary input image or feature map f using parameter θ to generate an output

$$\mathrm{ST}(f,\theta) = [\kappa_h(\theta) \otimes \kappa_w(\theta)] * f, \tag{1}$$

where κ_h and κ_w are 1-dimensional kernels, and $\otimes$ and $*$ are outer-product and convolution respectively.

A spatial transformer can be flexibly inserted into an existing network when it requires an attention mechanism. Meanwhile, it could be directly trained with back-propagation, without turning to reinforcement learning for help as performed in most attention-based approaches [42] [44]. To automatically learn to focus on multiple discriminative object patches of the input video frames, spatial-transform attention applies a spatial transformer at each time step. Following the original ST work [32], all these transformers at different time steps share a single localization network to predict multiple transformation parameters θ as shown in Fig. 3.

C. Transformation Generator

Once the ROIs are obtained, we use them to generate future frames. However, simply modeling visual evolutions at the pixel level suffers from inherent compounding errors and is difficult to be applied for long-term frame generation. In practice, we choose to model the visual evolution trajectory of ROIs at the transformation level. We expect to learn a

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2018.2882061, IEEE Transactions on Circuits and Systems for Video Technology

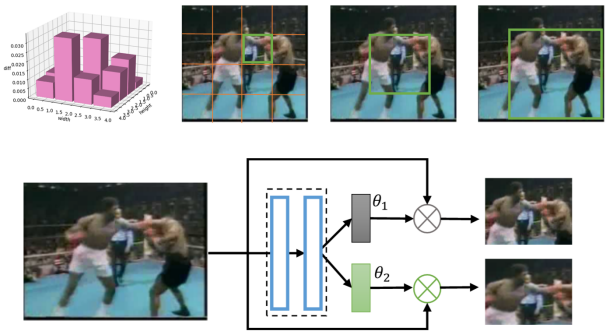

Fig. 3. Two methods used to generate ROIs: (a) Pyramid sampling. The first image shows the l_2 difference for each bin at level 1. The next three images illustrate three levels of the pyramid. (b) Spatial-transform learning. The multi-stream architecture is able to focus on multiple key object patches. For simplicity, we only show the top two most possible regions where motion occurs.

high-level pattern in the frame sequence to perform video prediction. Specifically, we represent such a pattern with two parts, which we show are suitable to generate future frames. The first one is called the transformer, which is used to synthesize transformed ROIs that model the changed contents while preserving unchanged background. The other is called the mask, which stores the generation rule of how the next frame is formed with the original frames and the transformed ROIs.

To achieve this, we design the transformation generator with an encoder-decoder network. We expect that the ideal structure can simultaneously produce transformer and mask. In particular, the encoder-decoder structure enjoys the advantage that the output can be of the same size as input, acting as the mask. Meanwhile, the learned codes in the middle preserve higher level abstraction of the input data, which in our situation can be used as the transformer.

Concretely, for the first ROI, the transformation network first generates N transformers from the middle of the encoder-decoder network. These transformers can be viewed as convolutional kernels and are then applied to each ROI to produce N transformed ROIs for the next frame. Note that these transformed ROIs are of the same size as the input ROI. Then, all these transformed ROIs are combined with the original input frame sequence to generated the next frame candidate. This procedure is called transformation-guided masking and is presented in Section III-C. To leverage the sequential information of different ROIs, we establish an RNN to govern the transformation generator. At each time step t, the generation procedure for the next frame candidate is the same as the first ROI, except that the next frame candidate generated in the last time step is also used in the current transformation-guided masking procedure, which acts as additional information to generate the next candidate of the next frame. Finally, the next frame candidate of the last ROI in the last time step is the final choice of the next frame.

D. Predicting Future Frames

In this subsection, we detail how to form the next frame candidate given the previously learned transformer and mask and determine the final choice of the next frame. Then, we show how our system can be extended to generate long-term future frames with diversity.

Transformation-Guided Masking. The next frame generation is implemented with the proposed transformation-guided masking procedure. At each RNN time step, the multiple transformers are combined with input ROIs with a merge network using convolution computations to generate a stack of transformed ROIs, which records different patterns of frame evolutions. The mask stores the generation rule to form the next frame with the transformed ROIs and the next frame candidate generated in the last time step. In more detail, at time step t, the current predicted candidate of next frame p_t is formulated as

$$p_t = m_t^0 \odot p_{t-1} + \sum_{i=1}^{N} m_t^i \odot (k_t^i, r_t^n), \tag{2}$$

where m_t^0 is the mask for the next frame candidate generated in the last time step, m_t^i is the mask for each transformed ROI, $\odot$ is element-wise multiplication, (k_t^i, r_t^n) indicates the transformed ROIs obtained by applying convolutional kernel k_t^i at every position of the ROI r_t^n, and p_0 is the input frame sequence. In particular, we perform channel-wise softmax to ensure that the masks sum to 1 at each pixel.

The first term uses the last generated next frame candidate, which contains the transformation information of all previous ROIs. This part preserves global contents, including moving objects and background. The second term addresses the current ROI with each learned transformer, which records how each ROI evolves from the past to the future in previously observed frames. Masks are generation rules used to integrate transformed ROIs with original frames to generate future frames. We use the frame generated at the last time step as the final prediction of the next frame.

Long-term Prediction. We design two methods to achieve long-term prediction. The first is to learn multiple stacked transformers and masks in the transformation generator, and directly generate multiple future frames with them. This approach can use the full potential of our structure since local transformation is easy to learn, even in a long trajectory. The second choice is the recursive implementation, which applies the procedures of ROI generation and transformation generation for each generated future frame recursively. This type of implementation is widely used in recent video prediction approaches. However, the difference between our situation with others is that the recursive prediction of local regions suffers less from intermediate blurry results and loss amplification and can thus model long-term transformation patterns. Experiments suggest both can yield promising results, which demonstrate the advantages of our method because most global pixel-level solutions can hardly generate high-quality long-term future frames in a recursive manner.

Predicting Diverse Future Frames. To generate diverse future frames, the key idea behind our system is to sample dif-

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2018.2882061, IEEE Transactions on Circuits and Systems for Video Technology

ferent transformation generators to produce multiple plausible transformations and masks. To achieve this, we use a popular strategy in the GAN community that introduces a latent variable that follows a specific distribution (e.g. Gaussian Distribution). This latent variable is then sent to conditional code generator with input frame sequence to obtain conditional codes. These codes preserve the visual information of the original inputs and can thus be used to revise the transformation generator through sampling different latent variables.

By sampling different condition codes, our transformation generator is able to generate diverse patterns of how frame evolutions can be modeled. In practice, we choose to place the conditional codes with the controller RNN that governs the transformation generator. In this way, the conditional code generator can be trained with the transformation generator in an end-to-end manner as follows.

E. Training Method

Since our system is end-to-end, we use three types of loss to train our system: l_2 loss between predicted frames and ground truth, gradient difference loss (GDL) and adversarial loss. The first two losses are easy to implement. We mainly address how to design adversarial loss in our training method.

Standard GAN trains two adversarial models simultaneously: a generator G that captures the data distribution and a discriminator D that distinguishes between fake samples drawn from G and real samples coming from the training data. As suggested in [13], an adversarial loss can address blurry predictions caused by l_2 loss. For a specific input sequence, if our model can produces the future frames p and p' with equal probability, then the value $p_{avg} = (p+p')/2$ will minimize the l_2 loss over the data, even if p_{avg} is not the expected future frame. Thus, we decide to add adversarial loss in our training method.

Specifically, we denote y as the predicted future frame, and $R = \{R_1, R_2, ..., R_T\}$ as the ROIs. Training D requires keeping the weights of G fixed and classifying $D(y)$ into label 1 while classifying $D(G(R_t))$ into label 0. Therefore, the loss function of D can be defined as

$$\mathcal{L}^D_{adv}(R, y) = \sum_{t=1}^{T} \mathcal{L}_{bce}(D(y), 1) + \mathcal{L}_{bce}(D(G(R_t)), 0), \quad (3)$$

where L_{bce} is the binary cross-entropy loss defined as

$$\mathcal{L}_{bce}(Y, Y') = -\sum_i (Y'_i \log(Y_i) + (1 - Y'_i) \log(1 - Y_i)) \quad (4)$$

where Y_i takes its values in $\{0, 1\}$ and Y'_i in $[0, 1]$.

Training G requires keeping the weights of D fixed, and confusing D to classify the fake samples $G(R_t)$ into label 1. We formulate the loss function of G as follows

$$\mathcal{L}^G_{adv}(R, y) = \sum_{t=1}^{T} \mathcal{L}_{bce}(D(G(R_t)), 1). \quad (5)$$

Minimizing this loss means that the generative model G makes the discriminative model D as confused as possible in the sense that D will not discriminate the prediction correctly.

Finally, we combine the three losses with different weights as

$$\mathcal{L}(R, y) = \lambda_{adv}\mathcal{L}^G_{adv}(R, y) + \lambda_{l_2}\mathcal{L}_{l_2}(R, y) + \lambda_{gdl}\mathcal{L}_{gdl}(R, y) \quad (6)$$

where there is a tradeoff to adjust, by means of the three λ parameters, among sharp predictions due to the adversarial principle, similarity with the ground truth, and the image gradient predictions. In practice, we determine them through a grid search experiment.

F. Standing on Single Frame

Finally, we show that our system is able to generate future frames conditioned on a single frame. Compared with a consecutive frame sequence, a single frame encodes less motion information and has a higher possibility of evolving into diverse future frames. In our proposed two strategies for long-term generation, the first one (generating multiple stacked transformations) is suitable for this situation because the iterative prediction is hard to capture meaningful motion pattern due to the limited input signal. However, multiple sets of transformations can occasionally produce plausible future frame sequences because they are trained with real videos with adversarial loss.

In practice, we choose to generate multiple transformers and masks at once and directly produce multiple future frames. In our experiments, we find that simply using the same networks as when the frame sequence is input can result in blurry predictions, which we attribute to a single frame containing less motion information and consequently affecting the quality of transformers and masks. To solve this problem, we use deeper networks for the condition code generator. We expect to mine more information from the input frame and consequently learn a more plausible and potential transformation that exists in the original input single frame. The specific configurations are given in the experimental part regarding the setting of video prediction on a single frame.

IV. Experiments

A. Datasets

The 2D Shape dataset [25] consists of synthetic 2D shape RGB videos. There are only three types of objects in this dataset moving horizontally, vertically or diagonally with random velocity. All three objects are simple 2D shapes: circles, squares, and triangles. The original dataset only contains image pairs that have 2 consecutive frames. We extrapolate it to convert image pairs into video clips that have 5 frames. We use $20,000$ clips for training and 500 for testing as in the original work [25].

The Moving MNIST dataset [18] consists of videos where two MNIST digits move in random directions with constant speed inside a 64×64 frame. The $64,000$ training video clips and 320 testing clips are generated on-the-fly. We use the code provided by [18] to generate $100,000$ training sequences to train our model, and we test it on the test set, which contains $10,000$ sequences. Each video clip consists of 5 frames.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2018.2882061, IEEE Transactions on Circuits and Systems for Video Technology

The UCF101 dataset [45] contains $13,320$ videos belonging to 101 action categories. The videos have a resolution of 240×320 and were sampled at 30 frames per second. Most frames in this dataset only have a very small portion of the image actually moving. In our experiments, we train our model directly on this difficult dataset and report our results on a subset of 378 test videos from UCF101.

The THUMOS15 test set [46] consists of over $5,600$ temporally untrimmed videos. The UCF101 dataset is the training dataset for the THUMOS challenge, and thus THUMOS is a relevant choice for the testing set.

The CalTech Pedestrian dataset [47] consists of videos from a dashboard-mounted camera on a vehicle driving around Los Angeles. Testing sequences were made to match the frame rate of the THUMOS15 and cropped to 128×160 pixels. Quantitative evaluation was performed on the entire CalTech test partition, split into sequences of 10 frames.

B. Experimental Setup and Evaluation Metrics

Our video prediction system is flexible, as discussed previously; thus we consider three experimental settings to evaluate its performance on different datasets, as follows.

- Standard video prediction: Given an observed sequence of frames as input, the system is asked to predict the next frame and future frames of several steps. In practice, we condition generation on 4 observed ground truth frames and we predict the following 1, 4 and 8 frames. Following [13] and [27], we train models on the generic consumer videos from UCF101, and we evaluate on the UCF101 and THUMOS15 test sets for comparison with recent approaches. This part is examined on the Moving MNIST for ablation experiments.
- Diverse video prediction: The input is the same as above, but the output is required to be diverse and plausible, compared to the ground truth and plausibility. In detail, we condition generation on 4 observed frames and we predict 5 types of the following 8 frames. We experiment with this setting on the Moving MNIST dataset.
- Video prediction on single frame: Based on the second setting, the input is a single frame rather than 4 frames. This setting is examined on the 2D Shape dataset.

For the evaluation metrics, for the first setting, we employ the widely used Peak Signal to Noise Ratio (PSNR) and Structural Similarity Index Measure (SSIM) [48]. PSNR is commonly used to measure the quality of reconstruction, and SSIM is a popular method for predicting the perceived quality of images. A larger score is better for both PSNR and SSIM.

For the other two settings, since there are no general evaluation metrics for this diverse prediction due to the only one ground truth (existing evaluation metrics are designed in method oriented such as [39] [25]), we use three evaluation metrics. The first two metrics are PSNR and SSIM. The third is a non-reference measurement, i.e., a recently proposed relative image quality assessment (RIQA) [35], which is based on the popular blind image quality assessment (BIQA) method BRISQUE [49]. RIQA calculates the decreasing proportion of quality score between inputs and outputs, defined as

$$\text{RIQA} = \frac{\text{BRISQUE(Input) - BRISQUE(Output)}}{\text{BRISQUE(Input)}} \tag{7}$$

where BRISQUE(Output) is the average BRISQUE of all the predicted future frames. We believe that RIQA is fair and reasonable because it reflects the plausibility of predicted frames in terms of both continuity and consistency.

C. Implementation Details

The major implementations of our system, including ROI generation (spatial-transform learning), transformation generation that is governed by an RNN architecture, discriminator and the adversarial training, are all derived from the TensorFlow platform [50]. We use one Tesla K80 GPU to accelerate the procedure of training and testing. For 2D Shape and Moving MNIST, all input images are resized to 32×32, and the size of each ROI is 16×16. For UCF101, the input images are resized to 64×64 with an ROI size of 32×32. When the pyramid sampling strategy is applied for ROI generation, each input image is split into 3×3 grids for 2D Shape and Moving MNIST, and 4×4 grids for UCF101. The core transformation generator is constructed in an auto-encoder that includes 3 convolutional layers and 3 deconvolutional layers. For spatial-transform learning, we constrain the size of the attention window to be 16×16 for 2D Shape and Moving MNIST and 32×32 for UCF101. Rather than using more advanced LSTMs, we employ the standard RNN structure to govern the transformation generator. Specifically, the current hidden state h_t is computed as the sum of the encoder output and the output of convolving the previous state h_{t-1}. All encoder-decoders share parameters at different RNN steps. Throughout our experiments, we fix the RNN to observe 5 ROIs ($T = 5$) for each sample. The number of transformers N is set to 6. The discriminator D is constructed by 4 spatiotemporal convolutional layers followed by 2 fully connected layers. Our system takes approximately 100ms to predict one future frame and flow given a sequence of 4 previous frames on the UCF101 dataset. From diverse future frame prediction, we use standard Gaussian distribution to generate the latent variables. The conditional codes are obtained from the input frames using a two fully connected layers.

D. Ablation Experiments

We first conduct several experiments to examine some typical architectural variants of our system on the Moving MNIST dataset. Then, we study the impacts of different designs to verify our concept and investigate the optimal architecture. Keeping the main framework fixed, we experiment on 6 system structures as follows

- GT: Learning transformations at the global level without ROI generation, which observes the entire frames 5 times and merges the generated frames into an integrated one.
- LT-FixedROI-Parallel: Learning local transformation network that observes 5 fixed locations in parallel, which

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2018.2882061, IEEE Transactions on Circuits and Systems for Video Technology

TABLE I. COMPARISON OF PSNR AND SSIM RESULTS FOR 5 VARIANTS OF OUR ARCHITECTURE FOR NEXT FRAME PREDICTION ON THE MOVING MNIST TEST DATASET.

Model	PSNR	SSIM
GT	21.0	0.88
LT-FixedROI-Parallel	22.4	0.89
LT-FixedROI-RNNSerial	25.6	0.90
LT-MaskCNNROI-RNNSerial	22.7	0.89
LT-PyramidROI-RNNSerial	28.4	0.91
LT-STLROI-RNNSerial	27.7	0.93

TABLE II. COMPARISON OF AVERAGE PSNR AND SSIM RESULTS FOR LONG-TERM PREDICTION ON THE MOVING MNIST TEST DATASET USING THE LT-PYRMAIDROI-RNNSERIAL CONFIGURATION.

Model	PSNR	SSIM
Recursive Prediction 4	25.6	0.90
Long-Term Transformation 4	25.9	0.91
Recursive Prediction 8	23.1	0.89
Long-Term Transformation 8	22.4	0.88

means directly learning transformation for each ROI and merging them into a future frame.

- LT-FixedROI-RNNSerial: Learning local transformation network that observes 5 fixed locations in serial governed by an RNN.
- LT-MaskCNNROI-RNNSerial: Learning local transformation network that observes 5 locations in serial with the boxes that contains the masks from Mask RCNN [51].
- LT-PyramidROI-RNNSerial: Learning local transformation network that observes 5 locations in serial with pyramid sampling.
- LT-STLROI-RNNSerial: Learning local transformation network that observes 5 locations in serial with spatial transformer learning.

Table I reports the results of the above variants on the Moving MNIST test set. We find that methods using ROI generation can outperform the global transformation model, which illustrates that learning local transformations can describe object movement more accurately. Additionally, the LT-FixedROI-RNNSerial model formulated as a recurrent network achieves better performance than its parallel counterpart LT-FixedROI-Parallel, since internal representations encoded from previous patches can provide auxiliary information for current step. Compared with learning transformations at fixed locations, models with our proposed ROI generation strategies (LT-FixedROI-RNNSerial and LT-STLROI-RNNSerial) can achieve better PSNR and SSIM results. This result suggests that attending to patches that contain dense motion information and converting objects step by step is a practical way to improve performance for video prediction. We also examine a method using the ROIs generated from Mask RCNN [51], where the masks of arbitrary-size are resized as ROIs to match Eq. 2. The result in Table I suggests that this method produces is worse than other choices in terms of video prediction quality.

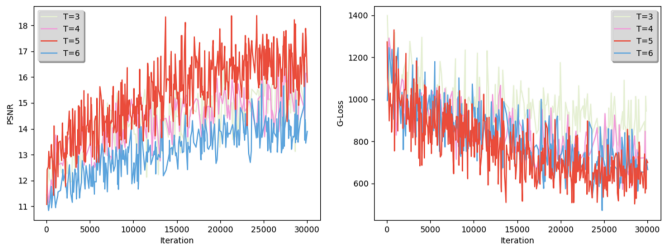

Fig. 4. PSNR and generator losses using different numbers (T) of RNN steps. With increasing training iterations, the PSNR increases and the G-loss gradually decreases. When $T = 5$, the model achieves the best performance.

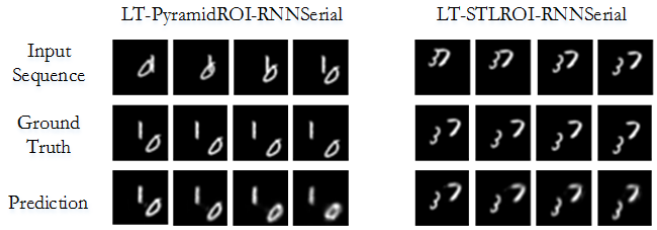

Fig. 5. Examples on the Moving MNIST dataset using LT-PyramidROI-RNNSerial (left) and LT-STLROI-RNNSerial (right).

In addition, we explore the impact of the number T of RNN steps in serial implementations. We conduct experiments for $T = 3, 4, 5, 6$ based on the LT-STLROI-RNNSerial model, with 100k training iterations. As shown in Fig. 4, during the training procedure, the model achieves the best performance in terms of PSNR and generator loss when $T = 5$.

For long-term prediction, we examine two types of strategies as addressed in Section III-C, where Recursive Prediction indicates recursive implementation using previously generated frame as the current input, while Long-term Transformation denotes directly learning multiple transformers and masks at one time and using them to form long-term future frames. The results are listed in Table II under two settings (predicting 4 and 8 future frames), where the results show that both strategies can yield similar performance. Fig. 5 demonstrates some generated frames (output 4 frames given 4 input frames) of LT-FixedROI-RNNSerial and LT-STLROI-RNNSerial on the Moving MNIST test set. Referring to the ground truth, the figure clearly indicates that our model can well predict the movement for different digits in the future frames.

E. Standard Video Prediction

Given the results from the previous ablation experiments, we use LT-FixedROI-RNNSerial and LT-STLROI-RNNSerial as our competitive models to compare with other approaches for standard video prediction including next frame prediction and long-term prediction. We mainly evaluate and compare with several models: a baseline that merely copies the last frame used for conditioning; a baseline method that estimates optical flow [52] from two consecutive frames and extrapolates flow in subsequent frames under the assumption of constant flow speed; an adversarially trained multi-scale CNN [13],

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2018.2882061, IEEE Transactions on Circuits and Systems for Video Technology

TABLE III. PERFORMANCE COMPARISON OF NEXT FRAME PREDICTION ON UCF101 AND THUMOS15 DATASETS.

Method	UCF101		THUMOS15	
	PSNR	SSIM	PSNR	SSIM
Last Frame	28.2	0.89	27.8	0.87
Optical Flow [52]	28.2	0.89	27.8	0.87
BeyondMSE [13]	28.2	0.89	27.8	0.87
EpicFlow [53]	29.1	0.91	28.6	0.89
DVF [27]	29.6	0.92	29.3	0.91
Nextflow [28]	29.9	-	-	-
Dual Motion GAN [30]	30.5	**0.94**	30.1	0.92
LT-FixedROI-RNNSerial (Ours)	32.9	0.92	33.2	**0.93**
LT-STLROI-RNNSerial (Ours)	**33.1**	0.93	**32.4**	0.92

TABLE IV. PERFORMANCE (MSE AND SSIM) OF VIDEO FRAME PREDICTION ON CALTECH AND YOUTUBE CLIPS AFTER TRAINING ON KITTI DATASET.

Method	UCF101		YouTube Clip	
	MSE	SSIM	MSE	SSIM
Last Frame	0.007985	0.762	0.01521	0.785
Optical Flow [52]	0.00628	0.789	0.01019	0.807
BeyondMSE [13]	0.00326	0.881	0.00853	0.820
PredNet [11]	0.00313	0.884	0.00679	0.858
Dual Motion GAN [30]	0.00241	0.899	0.00558	0.870
LT-FixedROI-RNNSerial (Ours)	0.00319	0.902	0.00551	0.889
LT-STLROI-RNNSerial (Ours)	**0.00235**	0.918	**0.00506**	**0.893**

several optical-flow-based methods extrapolate future frames by predicting intermediate flows including EpicFlow [53], deep voxel flow (DVF) [27], and Nextflow [28], a pixel-motion combined method Dual Motion GAN [30], and a transformation-based model [34].

UCF101 and THUMOS15 Table III shows the performance comparison on UCF101 and THUMOS15 in terms of PSNR and SSIM. As shown, our two methods achieve better results than other approaches in most cases, except on THUMOS15 regarding the SSIM measurement, where Dual Motion GAN is approximately 2% better than ours. Compared with the popular multi-scale CNN model, which needs to design generators and discriminators for each scale, our model is more flexible and lighter to deploy since all encoders and decoders in different RNN steps for each ROI share the same structures and parameters in our transformation generator. Note that the transformation-based model [34] did not provide all their generated images or the PSNR and SSIM results, we use their published examples to perform comparisons.

To provide more analysis with the transformation-based model [34] which also performs prediction at the transformation level, we present the qualitative comparison in Fig. 6. As shown, their model can produce sharper future frames, but it occasionally predicts multiple motions for the same object or fails to characterize motions for different objects. We draw the difference frames in Fig. 7 where we can see that the body of one basketball player from their predictions moves toward different directions, but this is not the case in reality. In contrast, our system can well capture motion information from different objects and predict how they move

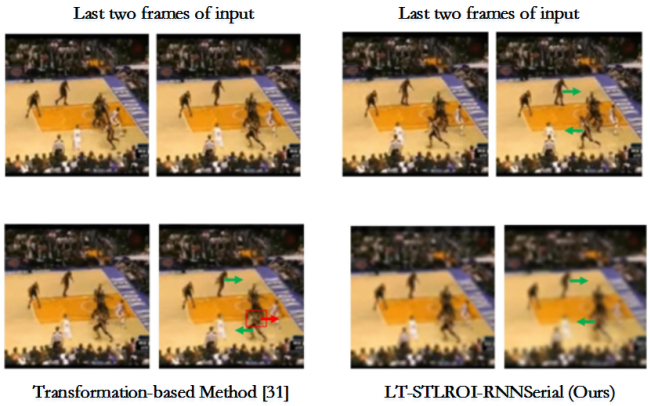

Fig. 6. Comparison of the predictions between transformation-based model [34] and our LT-STLROI-RNNSerial. The green arrows in the images indicate the realistic movement direction while red arrows indicate wrong prediction of the movement.

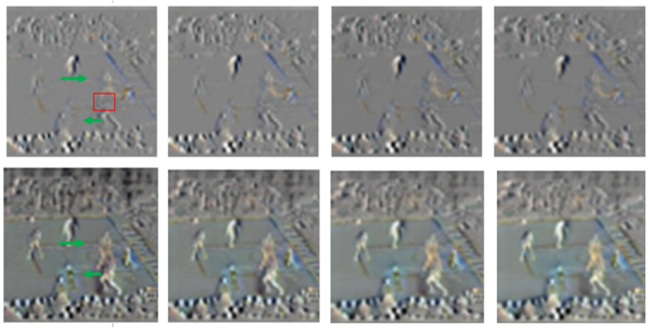

Fig. 7. Comparison of the difference images in Fig. 6 (bottom is ours while upper is the transformation-based model [34]). We can observe the movement trend for each basketball player more clearly in this way. The red box indicates the region that does not move correctly.

next. This is attributed to the iterative procedure of attending and transferring each ROI.

KITTI and Caltech Pedestrian. We also conduct video prediction experiments on a more challenging dataset, i.e. Caltech Pedestrian dataset. We evaluate the video prediction capabilities of our model on complex real-world sequences. Following the state-of-the-art PredNet [11] and MotionGAN [30], the models are trained using raw videos from the KITTI dataset [54] and evaluated on the test partition of the Caltech Pedestrian dataset [47]. We follow PredNets [11] procedure for training and validation, sampling 10 sequential frames from each video in the City, Residential, and Road categories, resulting in roughly 41k frames for the training set. In order to further validate our models generalization capability, we evaluate the trained model on 500 raw 1-minute clips from YouTube, collected using the keywords "dashboard videos", following the setting of Dual Motion GAN [30]. Fig. 9 demonstrates the training procedure of our method, where we can see a clear convergence at around 15000 training iterations.

Table IV reports the quantitative comparison with the state-of-the-art models BeyondMSE [13], PredNet [11] and Mo-

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2018.2882061, IEEE Transactions on Circuits and Systems for Video Technology

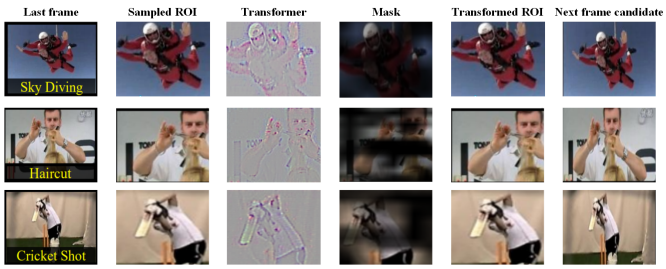

Fig. 8. Visualization of the intermediate outputs on UCF101 (test set).

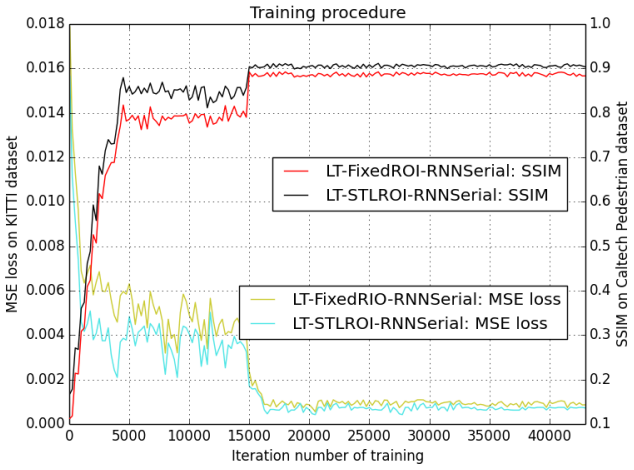

Fig. 9. Training procedure of our method on KITTI (MSE loss) and Caltech Pedestrian (SSIM) dataset.

tionGAN [30] on the video next-frame prediction task. We obtain the results of BeyondMSE [13] by training a model that minimizes the loss functions including adversarial loss and gradient descent loss, and replaces the backbone network with our frame generator, except for the motion autoencoder. Our model significantly outperforms two baselines, achieving MSE of 2.35×10^{-3} and SSIM of 0.899, compared to 2.41×10^{-3} and 0.899 of Dual Motion GAN [30], 3.13×10^{-3} and 0.884 of Prednet [11], and 3.26×10^{-3} and 0.881 of BeyondMSE. We show qualitative comparisons on the Caltech Pedestrian dataset in Figure 5. In Figure 10, the model is able to predict the motions of two vehicles and their shadows as they approach from different directions, as well as handle the stationary vehicle.

Computational time. Since Moreover, another key insight about the usage of transformation level prediction is to reduce the computation cost of calculating entire frame prediction as baseline. Could authors add some computation comparison with baselines or the method variant? My concern is that the framework tries to transform multiple ROIs and thus needs to run generators and RNN controller multiple times. It is not sure whether this local procedure is faster than entire-frame ways. Table V shows the computational cost to predict the next frame in the test stage. We compare four optical-based methods

Fig. 10. Qualitative comparisons with PredNet [11] and Dual Motion GAN [30] for next-frame prediction on car-cam videos from the Caltech dataset.

including Optical Flow [52] Nextflow [28] EpicFlow [53] and DVF [27], and three global-level prediction methods using deep neural networks, i.e., Beyond MSE [13], PredNet [11] and Dual Motion GAN [30]. The results show that our method can produce competitive next frame with faster execution time than most of the optical-flow-based methods and global-level methods.

Predicting multiple future frames. To predict multiple frames, we presented two type of implementations as addressed in Section III-D denoted as Recursive Prediction and Long-Term Transformation, respectively. [11] and [13] stated that recursive video prediction methods tend to perform poorly when predicting long-term frames, as deviations of the predictions unavoidably accumulate over time. However, as shown in Fig. 11, our system can surprisingly conduct good recursive prediction. Additionally, our Long-Term Transformation implementation that directly learns diverse transformations to generate multiple possibilities also provides promising results. In detail, both implementations of our system can achieve at least a 3% improvement over other methods, including multi-CNN [13], convolutional LSTM [55], and MCNet [12]. Fig. 10 shows a typical example of predicting multiple frames using PredNet, Dual Motion GAN and our method (LT-STLROI-RNNSerial). We can observe that our method can accurately predict the motions of two vehicles.

F. Predicting Diverse Future Frames

For diverse generation of future frames, we condition generation on 4 observed frames and we predict 5 types of the following 8 frames. We experiment with this setting on the Moving MNIST dataset. PSNR, SSIM and RIQA are used to evaluate the generated frames. PSNR and SSIM measure the distance between the generated frames and the ground truth, while RIQA judges the plausibility of the generated frames in terms of the continuity and consistency, compared with the input frames. We expect that the diverse generations resemble the ground truth or evolve smoothly.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2018.2882061, IEEE Transactions on Circuits and Systems for Video Technology

TABLE V. COMPUTATIONAL TIME (MS) TO PREDICT THE NEXT FRAME IN THE TEST STAGE ON THE UCF101 DATASET.

Methods	Time (ms)	PSNR	SSIM
Optical Flow [52]	245.2	28.2	0.89
Nextflow [28]	179.1	29.9	0.89
EpicFlow [53]	162.6	29.1	0.91
DVF [27]	194.5	29.6	0.92
Beyond MSE [13]	185.1	28.2	0.89
PredNet [11]	93.1	30.1	0.92
Dual Motion GAN [30]	**80.5**	30.5	**0.94**
LT-STLROI-RNNSerial (Ours)	98.3	**33.1**	0.93

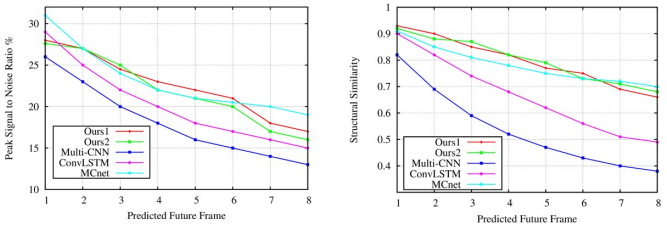

Fig. 11. Quantitative comparison for long-term video prediction. Given 4 input frames, these models predict 8 frames recursively except our Long-Term Transformation which directly generates 8 frames. Ours1 and Ours2 indicate Recursive Prediction and Long-Term Transformation, respectively.

We compare our model against popular multi-CNN [13], and two probabilistic models, i.e., visual dynamics [25] and conditional VAEs [39]. Since they did not provide experimental results on Moving MNIST, we reimplement their methods. In particular, for multi-CNN, we obtain their predictions at 5 levels to represent 5 diverse future frames. As shown in Table VI, our model again shows better performance in terms of all three metrics compared to the other methods. From the experiments, we find that their methods can produce better single instance than ours, but their average scores of all 5 diverse future frames are lower than ours. This phenomenon indicates that our system can learn multiple plausible transformation patterns in the consecutive frames. We also provide some qualitative results on the UCF101 dataset in Fig. 12, which shows that our method can generate diverse plausible future frames for real-world videos.

G. Video Prediction on Single Frame

Finally, we test a more difficult setting: taking only a single frame as input. We experiment with our framework using the synthetic 2D Shapes dataset [25]. We compare with existing works that can generate future frames on single frame: Visual Dynamics [25], Dense Optical Flow [14], and a transformation-based method [34]. We also set a baseline where the transformation generator in our system is replaced by a standard generator that directly outputs flattened pixels.

Table VII presents the detailed performance comparisons. As shown, our framework outperforms the baseline, suggesting that the architecture of our framework is reasonable. In addition, we observe that our system and [34] produce frames with

TABLE VI. AVERAGE SCORES OF PSNR, SSIM AND RIQA FOR DIVERSE GENERATION (5 TYPES OF FUTURE FRAMES).

Methods	PSNR	SSIM	RIQA
Beyond MSE [13]	17.1	87.3	12.4
Visual Dynamics [25]	16.6	86.9	9.6
Conditional VAEs [39]	19.5	86.2	10.8
Ours	**21.3**	**88.3**	**6.9**

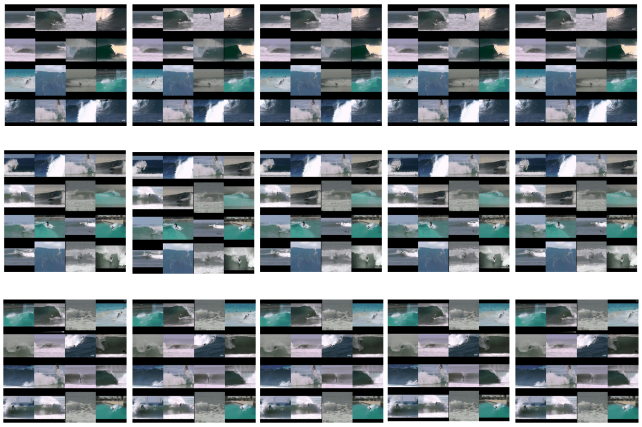

Fig. 12. Sixteen Examples of 3 diverse (one type for one row) video predictions (5 frames from left to right) on the UCF101 dataset .

better qualities than others, which indicates the advantages of learning transformations between frames. However, our system is better than [34] in terms of both PSNR and SSIM, which is perhaps due to our condition codes preserving visual information of the input frame. Note that [34] yields an RIQA that is approximately 1.5% better than ours.

Multiple results are shown in Fig. 13. It is clear that different condition codes lead to different imaginary videos with the same input image. The motions in those videos are notably dissimilar. Figs. 13 (a) and (b) present a perception comparison among our framework and the baseline where both are trained in the same iteration. As show, generation from the baseline leads to blur because of the intrinsic ambiguity of the image. Fig. 13 (c) shows two sampled difference frames with different condition codes, which indicates that our system can learn multiple transformations with different latent variables.

V. CONCLUSION

In this work, we have presented a flexible and powerful video prediction system. Unlike popular video prediction methods that are performed at the global pixel level, we focus on ROIs and learn patterns of frame evolutions at the transformation level. Given a sequence of frames or even a single frame, our system is able to accurately predict the next frame and long-term future frames. Moreover, it can produce diverse plausible future frames that preserve continuity and consistency with the input.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2018.2882061, IEEE Transactions on Circuits and Systems for Video Technology

TABLE VII. QUANTITATIVE COMPARISON WITH RELATED VISUAL PREDICTION WORKS FROM A SINGLE FRAME ON THE 2D SHAPE DATASET.

Methods	PSNR	SSIM	RIQA
Baseline	12.2	85.6	23.3
Optical Flow [14]	15.7	87.8	15.7
Beyond MSE [13]	17.1	88.3	12.1
Transformation-based [34]	16.6	86.9	**7.8**
Ours	**21.3**	**88.3**	9.3

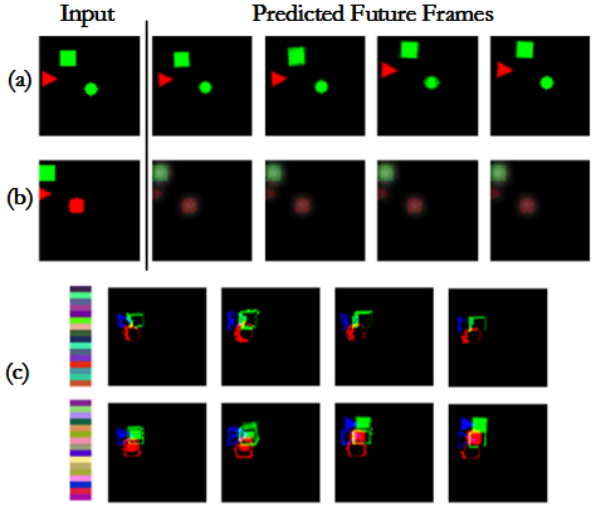

Fig. 13. Video prediction from a single frame. (a)(b) denote the input image and one generated video by our method and the baseline. The first column of (c) indicates different condition codes, the remaining columns show the difference frames of the generated difference frames compared with the input.

ACKNOWLEDGMENT

This work was supported in part by the Shenzhen Peacock Plan (20130408-183003656), in part by the Shenzhen Key Laboratory for Intelligent Multimedia and Virtual Reality (ZDSYS-201703031405467), and in part by the National Natural Science Foundation of China (U-1613209).

REFERENCES

[1] P. V. K. Borges, N. Conci, and A. Cavallaro, "Video-based human behavior understanding: A survey," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 23, no. 11, pp. 1993–2008, 2013.

[2] Z. Li, W. Wang, N. Li, and J. Wang, "Tube convnets: Better exploiting motion for action recognition," in *ICIP*, 2016, pp. 3056–3060.

[3] J. Wang, W. Wang, R. Wang, W. Gao *et al.*, "Deep alternative neural network: Exploring contexts as early as possible for action recognition," in *NIPS*, 2016, pp. 811–819.

[4] K. Kang, H. Li, J. Yan, X. Zeng, B. Yang, T. Xiao, C. Zhang, Z. Wang, R. Wang, X. Wang *et al.*, "T-cnn: Tubelets with convolutional neural networks for object detection from videos," *IEEE Transactions on Circuits and Systems for Video Technology*, 2017.

[5] X. Chen, W. Wang, W. Li, and J. Wang, "Attention-based two-phase model for video action detection," in *CAIP*, 2017, pp. 81–93.

[6] W. Li, W. Wang, X. Chen, J. Wang, and G. Li, "A joint model for action localization and classification in untrimmed video with visual attention," in *ICME*, 2017, pp. 619–624.

[7] G. Wang, W. Wang, J. Wang, and Y. Bu, "Better deep visual attention with reinforcement learning in action recognition," in *ISCAS*, 2017, pp. 1–4.

[8] J. Wang, W. Wang, and W. Gao, "Multiscale deep alternative neural network for large-scale video classification," *IEEE Transactions on Multimedia*, vol. 20, no. 10, pp. 2578–2592, 2018.

[9] R. Goroshin, M. F. Mathieu, and Y. LeCun, "Learning to linearize under uncertainty," in *NIPS*, 2015, pp. 1234–1242.

[10] C. Finn, I. Goodfellow, and S. Levine, "Unsupervised learning for physical interaction through video prediction," in *NIPS*, 2016, pp. 64–72.

[11] W. Lotter, G. Kreiman, and D. Cox, "Deep predictive coding networks for video prediction and unsupervised learning," *arXiv preprint arXiv:1605.08104*, 2016.

[12] R. Villegas, J. Yang, S. Hong, X. Lin, and H. Lee, "Decomposing motion and content for natural video sequence prediction," *arXiv preprint arXiv:1706.08033*, 2017.

[13] M. Mathieu, C. Couprie, and Y. LeCun, "Deep multi-scale video prediction beyond mean square error," *arXiv preprint arXiv:1511.05440*, 2015.

[14] J. Walker, A. Gupta, and M. Hebert, "Dense optical flow prediction from a static image," in *ICCV*, 2015, pp. 2443–2451.

[15] A. Dosovitskiy, P. Fischer, E. Ilg, P. Hausser, C. Hazirbas, V. Golkov, P. van der Smagt, D. Cremers, and T. Brox, "Flownet: Learning optical flow with convolutional networks," in *ICCV*, 2015, pp. 2758–2766.

[16] M. Ranzato, A. Szlam, J. Bruna, M. Mathieu, R. Collobert, and S. Chopra, "Video (language) modeling: a baseline for generative models of natural videos," *arXiv preprint arXiv:1412.6604*, 2014.

[17] J. Oh, X. Guo, H. Lee, R. L. Lewis, and S. Singh, "Action-conditional video prediction using deep networks in atari games," in *NIPS*, 2015, pp. 2863–2871.

[18] N. Srivastava, E. Mansimov, and R. Salakhutdinov, "Unsupervised learning of video representations using lstms." in *ICML*, 2015, pp. 843–852.

[19] R. Villegas, J. Yang, Y. Zou, S. Sohn, X. Lin, and H. Lee, "Learning to generate long-term future via hierarchical prediction," *arXiv preprint arXiv:1704.05831*, 2017.

[20] X. Chen, W. Wang, J. Wang, and W. Li, "Learning object-centric transformation for video prediction," in *ACM Multimedia*, 2017, pp. 1503–1512.

[21] I. Sutskever, G. E. Hinton, and G. W. Taylor, "The recurrent temporal restricted boltzmann machine," in *NIPS*, 2009, pp. 1601–1608.

[22] V. Michalski, R. Memisevic, and K. Konda, "Modeling deep temporal dependencies with recurrent grammar cells"","," in *NIPS*, 2014, pp. 1925–1933.

[23] R. Mittelman, B. Kuipers, S. Savarese, and H. Lee, "Structured recurrent temporal restricted boltzmann machines," in *ICML*, 2014, pp. 1647–1655.

[24] C. Vondrick, H. Pirsiavash, and A. Torralba, "Generating videos with scene dynamics," in *NIPS*, 2016, pp. 613–621.

[25] T. Xue, J. Wu, K. Bouman, and B. Freeman, "Visual dynamics: Probabilistic future frame synthesis via cross convolutional networks," in *NIPS*, 2016, pp. 91–99.

[26] X. Chen, W. Wang, and J. Wang, "Long-term video interpolation with bidirectional predictive network," in *VCIP*, 2017, pp. 1–4.

[27] Z. Liu, R. Yeh, X. Tang, Y. Liu, and A. Agarwala, "Video frame synthesis using deep voxel flow," *arXiv preprint arXiv:1702.02463*, 2017.

[28] N. Sedaghat, "Next-flow: Hybrid multi-tasking with next-frame prediction to boost optical-flow estimation in the wild," *arXiv preprint arXiv:1612.03777*, 2016.

[29] V. Patraucean, A. Handa, and R. Cipolla, "Spatio-temporal video autoencoder with differentiable memory," *arXiv preprint arXiv:1511.06309*, 2015.

[30] X. Liang, L. Lee, W. Dai, and E. P. Xing, "Dual motion gan for future-flow embedded video prediction," in *CVPR*, 2017, pp. 1744–1752.

[31] P. Isola, J. J. Lim, and E. H. Adelson, "Discovering states and transformations in image collections," in *CVPR*, 2015, pp. 1383–1391.

[32] M. Jaderberg, K. Simonyan, A. Zisserman *et al.*, "Spatial transformer networks," in *NIPS*, 2015, pp. 2017–2025.

[33] X. Wang, A. Farhadi, and A. Gupta, "Actions˜ transformations," in *CVPR*, 2016, pp. 2658–2667.

[34] J. van Amersfoort, A. Kannan, M. Ranzato, A. Szlam, D. Tran, and S. Chintala, "Transformation-based models of video sequences," *arXiv preprint arXiv:1701.08435*, 2017.

[35] B. Chen, W. Wang, and J. Wang, "Video imagination from a single image with transformation generation," in *ACM Multimedia Thematic Workshops*, 2017, pp. 358–366.

[36] J. Donahue, P. Krähenbühl, and T. Darrell, "Adversarial feature learning," *arXiv preprint arXiv:1605.09782*, 2016.

[37] V. Dumoulin, I. Belghazi, B. Poole, A. Lamb, M. Arjovsky, O. Mastropietro, and A. Courville, "Adversarially learned inference," *arXiv preprint arXiv:1606.00704*, 2016.

[38] J. Yang, A. Kannan, D. Batra, and D. Parikh, "Lr-gan: Layered recursive generative adversarial networks for image generation," *arXiv preprint arXiv:1703.01560*, 2017.

[39] J. Walker, C. Doersch, A. Gupta, and M. Hebert, "An uncertain future: Forecasting from static images using variational autoencoders," in *ECCV*, 2016, pp. 835–851.

[40] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," in *ECCV*, 2014, pp. 346–361.

[41] L. Wu, Y. Wang, X. Li, and J. Gao, "Deep attention-based spatially recursive networks for fine-grained visual recognition," *IEEE Transactions on Cybernetics*, 2018.

[42] V. Mnih, N. Heess, A. Graves *et al.*, "Recurrent models of visual attention," in *NIPS*, 2014, pp. 2204–2212.

[43] D. J. Rezende, S. Mohamed, I. Danihelka, K. Gregor, and D. Wierstra, "One-shot generalization in deep generative models," *arXiv preprint arXiv:1603.05106*, 2016.

[44] S. Yeung, O. Russakovsky, G. Mori, and L. Fei-Fei, "End-to-end learning of action detection from frame glimpses in videos," in *CVPR*, 2016, pp. 2678–2687.

[45] K. Soomro, A. R. Zamir, and M. Shah, "Ucf101: A dataset of 101 human actions classes from videos in the wild," *arXiv preprint arXiv:1212.0402*, 2012.

[46] A. Gorban, H. Idrees, Y. Jiang, A. R. Zamir, I. Laptev, M. Shah, and R. Sukthankar, "Thumos challenge: Action recognition with a large number of classes," in *CVPR workshop*, 2015.

[47] P. Dollár, C. Wojek, B. Schiele, and P. Perona, "Pedestrian detection: A benchmark," in *CVPR*, 2009, pp. 304–311.

[48] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.

[49] A. Mittal, A. K. Moorthy, and A. C. Bovik, "No-reference image quality assessment in the spatial domain," *IEEE Transactions on Image Processing*, vol. 21, no. 12, pp. 4695–4708, 2012.

[50] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin *et al.*, "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," *arXiv preprint arXiv:1603.04467*, 2016.

[51] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *ICCV*, 2017, pp. 2980–2988.

[52] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert, "High accuracy optical flow estimation based on a theory for warping," *ECCV*, pp. 25–36, 2004.

[53] J. Revaud, P. Weinzaepfel, Z. Harchaoui, and C. Schmid, "Epicflow: Edge-preserving interpolation of correspondences for optical flow," in *CVPR*, 2015, pp. 1164–1172.

[54] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, 2013.

[55] S. Xingjian, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-c. Woo, "Convolutional lstm network: A machine learning approach for precipitation nowcasting," in *NIPS*, 2015, pp. 802–810.

Jinzhuo Wang received the B.S. degree from the School of Electronic Engineering and Computer Science, Peking University, in 2013. He is currently a Ph.D. candidate at the school of Electronic Engineering and Computer Science, Peking University. His research interests include computer vision and deep learning. He has published several papers on relevant conferences and journals, such as NIPS, ACM Multimedia, AAAI, ICME, ICIP, and IEEE Transactions on multimedia.

Wenmin Wang (M'16) received the Ph.D. degrees in computer architecture from Harbin Institute of Technology, China, in 1989. After then, he worked as an assistant professor and associate professor, at Harbin University of Science and Technology as well as Harbin Institute of Technology. Since 1992, he gained about 18 years of oversea industrial experiences in Japan and America, in where served as staff engineer, chief engineer, general manager of software division, and etc. He came back the academia of China by the end of 2009, as a professor works at the School of Electronic and Computer Engineering, Peking University, China. His current research interests include computer vision, multimedia retrieval, artificial intelligence and machine learning.

Wen Gao (M'92-SM'05-F'09) received the Ph.D. degree in electronics engineering from the University of Tokyo, Japan, in 1991.

He is a Professor of computer science with Peking University, China. Before joining Peking University, he was a Professor of computer science with the Harbin Institute of Technology from 1991 to 1995, and a Professor with the Institute of Computing Technology of Chinese Academy of Sciences. He has published extensively including five books and over 600 technical articles in refereed journals and conference proceedings in the areas of image processing, video coding and communication, pattern recognition, multimedia information retrieval, multimodal interface, and bioinformatics. He served on the editorial board for several journals, such as the IEEE TRANSACTIONS ON CIRUCITS AND SYSTEMS FOR VIDEO TECHNOLOGY, the IEEE TRANSACTIONS ON MULTIMEDIA, the IEEE TRANSACTIONS ON AUTONOMOUS MENTAL DEVELOPMENT, the *Eurasip Journal of Image Communications*, and the *Journal of Visual Communication and Image Representation*. He chaired a number of prestigious international conferences on multimedia and video signal processing, such as IEEE ICME and ACM Multimedia, and also served on the advisory and technical committees of numerous professional organizations.