Static2Dynamic: Video Inference from a Deep Glimpse

Yu-Ying Yeh, Yen-Cheng Liu, Wei-Chen Chiu, and Yu-Chiang Frank Wang, *Member, IEEE*

***Abstract*—In this paper, we address a novel and challenging task of video inference, which aims to infer video sequences from given non-consecutive video frames. Taking such frames as the anchor inputs, our focus is to recover possible video sequence outputs based on the observed anchor frames at the associated time. With the proposed Stochastic and Recurrent Conditional GAN (SR-cGAN), we are able to preserve visual content across video frames with additional ability to handle possible temporal ambiguity. In the experiments, we show that our SR-cGAN not only produces preferable video inference results, it can also be applied to relevant tasks of video generation, video interpolation, video inpainting, and video prediction.**

***Index Terms*—video synthesis, video inference, generative model, adversarial learning.**

I. Introduction

In this paper, we tackle a unique video synthesis problem of *video inference*, which requires one to generate possible video sequences based on few observed (and non-consecutive) anchor frames, as shown in Fig. 1(a). Different from prior video synthesis tasks like *video generation from prior distribution* (see Sec. II-C) or *video prediction* (see Sec. II-D) from one or few consecutive frames, video inference needs to output more than one possible video while matching the input anchor frames at the associated time.

Take Fig 1(a) for example, given two face images (one with mouth closing and the other with mouth opening), it is possible to recover more than one complete video sequences (disgusting or smiling). We regard that a robust video inference model should be able to capture the intrinsic uncertainty of the observed input video frames in addition to sufficient temporal smoothness while preserving context information.

To address this unique and challenging task, we propose a novel recurrent network architecture of Stochastic and Recurrent Conditional GAN (SR-cGAN). To enforce the consistency of the output content across frames, we advance image-based conditional Generative Adversarial Network (cGAN) [1] to a video-based conditional GAN. To handle the ambiguity during video inference, stochasticity is introduced into our model to preserve intra-video temporal transition; this would allow randomness during the prediction of video frames. As detailed later, this is achieved by learning a smooth trajectory connecting the resulting visual representations of the observed frames in the resulting latent space, as shown in Fig. 1(b).

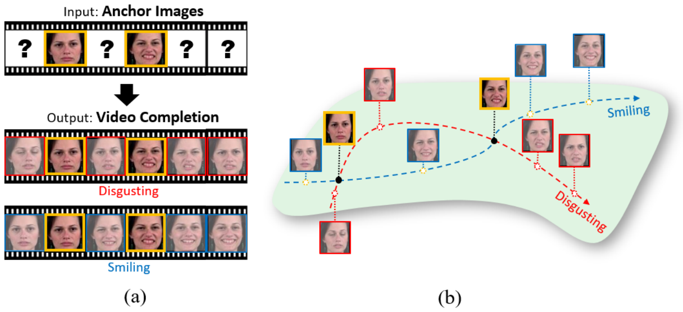


Fig. 1. Illustration of video inference via deep latent representation. Given an arbitrary number of non-consecutive video frames as anchors (in yellow bounding boxes), we observe a deep latent space for recovering more than one possible video sequence (e.g., curves in red and blue).

We highlight the contributions of this paper as follows:

- We address a more general type of task of video synthesis – *video inference*, which recovers more than one possible video sequence conditioned on non-consecutive input anchor frames.
- Our proposed Stochastic and Recurrent Conditional GAN (SR-cGAN) learns video representation not only preserving visual content but also modeling the ambiguity of the temporal structure in the recovered video sequences.
- We show that our proposed model can be generalized to the tasks of video generation from prior distribution, video interpolation, video inpainting, and video prediction.

We note that applications for video inference include discovering unseen past activities or events, synthesizing slow-motion videos (i.e., video super-resolution), and predicting future events due to the capability of recovering the past, intermediate and future visual dynamics from static input images. Therefore, we regard video inference is an important problem and worth exploring.

II. Related Works

We regard *video inference* is one of the tasks in video synthesis. Table I highlights and compares a number of works in video synthesis, including *video interpolation*, *video inpainting*, *video generation from prior information*, and *video prediction*, which are detailed in the following subsections.

Y.-Y. Yeh was with the Department of Computer Science and Engineering, University of California San Diego, La Jolla, CA, 92093 USA e-mail: yuyeh@eng.ucsd.edu

Y.-C. Liu was with the Department of Machine Learning, Georgia Institute of Technology, Atlanta, CA, 30332 USA e-mail: ycliu@gatech.edu

W.-C. Chiu was with the Department of Computer Science, National Chiao Tung University, Hsinchu, 30010 Taiwan e-mail: walon@cs.nctu.edu.tw

Y.-C. F. Wang was with the Department of Electrical Engineering, National Taiwan University, Taipei, 10617 Taiwan e-mail: ycwang@ntu.edu.tw

Manuscript received Oct 31, 2019; revised Oct 31, 2019.

TABLE I
COMPARISONS OF RECENT WORKS ON VIDEO SYNTHESIS.

	Synthesize Sequential Frames	Synthesize Intermediate Frames	Synthesize Succeeding Frames	Synthesize Preceding Frames
Video Interpolation	- / ✓	✓	-	-
Video Inpainting	✓	✓	-	-
Video Generation from Prior Dist.	✓	-	-	-
Video Prediction	✓	-	✓	-
Video Inference (Ours)	✓	✓	✓	✓

A. Video Interpolation

Video interpolation focus on recovering intermediate frames between any two consecutive original video frames. Recently, several neural network based models [2]–[7] are proposed to handle the challenging scenes including motion blur, light changing, or occlusion reasoning. Particularly, the model proposed by [8] is able to perform both interpolation and extrapolation. Nevertheless, while these models can synthesize high-quality intermediate frames, they require the input frames to be consecutive and cannot handle the case when the interval between frames is large (e.g., no more than 0.05 seconds apart) [9]). Video inference consider much larger temporal gaps between input frames and more general input condition than video interpolation.

B. Video Inpainting

While some video inpainting methods attempt to reconstruct missing image patches within one/few frames in a video, here we focus our discussions on those aiming at reconstructing a missing sequence of frames given preceding frames and succeeding frames. In other words, compared to video interpolation, the approaches for the above video inpainting tasks consider a larger temporal gap between input frames. Previously, several models [10], [11] have been evaluated in this challenging setting, while a recent deep learning-based model of TAI network [9] was proposed to tackle this task. Although video inpainting is close to our task of video inference, video inpainting relies on both preceding and succeeding frames to synthesize frames, not non-consecutive input frames (as the requirement of video inference).

C. Video Generation from Prior Distribution

With the observation of abundant video clips, several works [12]–[16] generate sequential video data from observed prior distributions by modeling the temporal and spatial structures from training videos. Thus, given a random sample from some prior (e.g., Gaussian prior), one can produce a sequence of frames which is similar to real ones (but cannot be conditioned on particular input frames).

D. Video Prediction

Another line of research related to video synthesis is video prediction, which generates the succeeding video sequences based on few consecutive video frames. As a growing popularity of interests, a large number of methods [15], [17]–[31] have been proposed to tackle this task. In addition, some models [16], [32]–[36] further synthesize video frames based on a single input frame.

E. Video Inference

Video inference, a unique task in video synthesis, aims at recovering/synthesizing preceding, intermediate, and succeeding frames based on few non-consecutive input frames. The most related and recent work proposed by [15] can complete a video with a constraint on the first and the last frames. Although video inference can be handled if we set the constrained frames other than the first and the last frames, the optimization process is required. Moreover, this method can be only applied to synthesis of human action videos since the model leverages human pose information as prior information, and decomposes the task of video inference into separated stages to complete. Our proposed SR-cGAN does not need to leverage explicit information (e.g., human pose or facial landmark), and can directly generate output videos as verified later in the experiments.

III. Proposed Method

A. Problem Definition and Notations

We define the observed M input frames as *anchor frames* $\mathcal{A} = \{x_1^a, ..., x_M^a\}$, which are imposed to appear in the synthesized video sequence of N-frames $\tilde{v} = \{\tilde{x}_1, ..., \tilde{x}_N\}$ at the associated time steps $\mathcal{T} = \{t_1, ..., t_M\}$. Note that t_i indicates that x_i^a appears at the t_i-th frame in $\tilde{v}$, while we have $M < N$ without loss of generality. As shown in Fig. 2 as an example, one needs to infer $N = 6$ frames video from $M = 2$ input anchor frames at time $t_1 = 2$ and $t_2 = 5$, respectively.

Due to the high dimensionality and diversity of real-world videos, we approach the task of video inference problem by decomposing it into the modeling of spatial and temporal structures. The former learns the frame-based representation from the input video to describe their content and spatial information, while the latter is to capture temporal variation within that video sequence, resulting in video-based representation. Such representations will be utilized in a recurrent network for video inference, which needs to observe visual content consistency while allowing temporal ambiguity in recovered outputs.

B. Base Model: Video Synthesis from Deep Video Representation

1) Learning Frame-Based Video Representation: Since a video is composed of consecutive frames, we first utilize

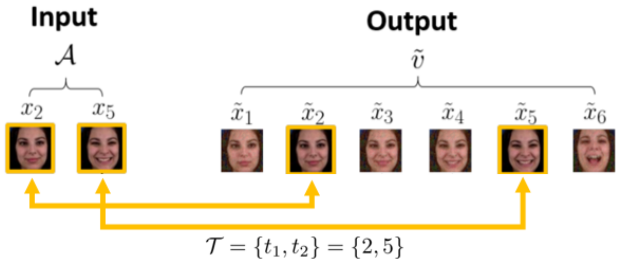


Fig. 2. Example of video inference. We take input anchor frames $\mathcal{A} = \{x_1^a, x_2^a\} = \{x_2, x_5\}$ at time $\mathcal{T} = \{t_1, t_2\} = \{2, 5\}$ for generating a video $\tilde{v} = \{\tilde{x}_1, ..., \tilde{x}_6\}$, while $\{\tilde{x}_2, \tilde{x}_5\}$ would match $\{x_2, x_5\}$.

the architecture of VAE-GAN [37] for learning frame-based representation, which consists of components of Variational Autoencoder (VAE) [38] and GAN [39]. As shown in Fig. 3(a), we have E_I, G_I, and D_I be the *image encoder*, *image generator*, and *image discriminator*, respectively. For the VAE part of VAE-GAN, E_I encodes an image (video frame) x to an *image latent representation* z_I and G_I decodes z_I back to image space:

$$z_I \sim E_I(x) = q_I(z_I|x),\ \tilde{x} \sim G_I(z_I) = p_I(x|z_I). \quad (1)$$

We thus define the objective function of this image-based VAE $\mathcal{L}_{I_{VAE}}$ as:

$$\begin{aligned} &\mathcal{L}_{I_{VAE}}(E_I, G_I) = \\ &\quad - \mathbb{E}_{q(z_I|x)}[\log p_I(x|z_I)] + KL(q(z_I|x)||p_I(z_I)), \end{aligned} \quad (2)$$

where the first term denotes the image reconstruction error, and the second term is the Kullback-Leibler divergence over the auxiliary distribution $q(z_I|x)$ and the prior distribution $p_I(z_I)$. Following [38], we have $z_I \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$.

For the component of GAN in VAE-GAN, G_I is to recover realistic images from the latent space to fool D_I, which aims to distinguish between the real images x and synthesized ones $G_I(z_I)$, so that $G_I(z_I)$ would fit the real data distribution. Thus, the objective function for this frame-based GAN $\mathcal{L}_{I_{GAN}}$ is defined as:

$$\begin{aligned} \mathcal{L}_{I_{GAN}}(G_I, D_I) = \log(D_I(x)) + \log(1 - D_I(G_I(z_I))) \\ + \log(1 - D_I(G_I(E_I(x)))). \end{aligned} \quad (3)$$

To sum up, the objective function of learning frame-based visual representation is defined as:

$$\underset{E_I, G_I}{min}\ \underset{D_I}{max}\ \mathcal{L}_{I_{VAE}}(E_I, G_I) + \mathcal{L}_{I_{GAN}}(G_I, D_I). \quad (4)$$

2) Learning Video Representation: Since a video consists of consecutive frames exhibiting temporal and content consistency, it is desirable to learn a more robust visual representation, which is beyond representation simply at the frame level. In our proposed network architecture in Fig. 3(b), we specifically have the latent space of video data exhibit the ability in modeling the distribution of reasonable representation trajectories. If such spaces are derived, video can be generated by randomly drawing a sample of video latent representation z_T from it.

As depicted in Fig. 3(b), the above process is achieved by utilizing a Recurrent Neural Network (RNN) as the *temporal generator* G_T, which models the distribution of latent representation trajectories. Thus, given a sample z_T, a sequence of frame-based representations $\{\tilde{z}_I^{(1)}, ..., \tilde{z}_I^{(N)}\}$ can be generated by G_T in each time step:

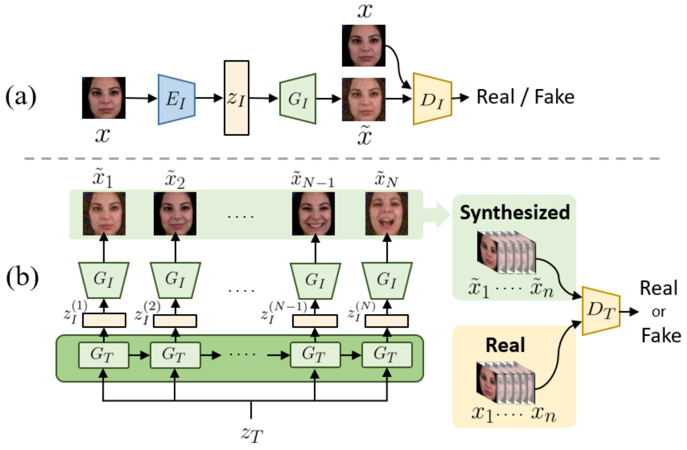


Fig. 3. Learning of (a) frame-based representation z_I via VAE-GAN and (b) video representation z_T. Note that RNN is utilized in (b) for learning video representation which preserves the associated temporal structure. Note that adversarial training is introduced in both (a) and (b) for improved video output quality.

$$\{\tilde{z}_I^{(1)}, ..., \tilde{z}_I^{(N)}\} \sim G_T(z_T), \quad (5)$$

and thus the corresponding video $\tilde{v} = \{\tilde{x}_1, ..., \tilde{x}_N\}$ can be produced accordingly:

$$\tilde{x}_t \sim G_I(\tilde{z}_I^{(t)}),\ \forall t \in \{1, ..., N\}. \quad (6)$$

As a result, one can obtain the synthesized video by:

$$\tilde{v} = \{\tilde{x}_1, ..., \tilde{x}_N\} \sim G_I(G_T(z_T)). \quad (7)$$

The above video synthesis model is trained by adversarial learning strategies. That is, the video generator $G_I(G_T(z_T))$ aims to generate realistic videos to make the *temporal discriminator* D_T hard to determine whether a video is synthesized from $G_I(G_T(z_T))$ or is a real video v. Thus, the objective function for this video-based GAN is defined as:

$$\begin{aligned} &\underset{G_T, G_I}{min}\ \underset{D_T}{max} \mathcal{L}_{T_{GAN}}(G_T, G_I, D_T) = \\ &\qquad \log(D_T(v)) + \log(1 - D_T(G_I(G_T(z_T)))). \end{aligned} \quad (8)$$

C. Stochastic & Recurrent Conditional-GAN (SR-cGAN) for Video Inference

The architecture of our proposed network for video inference, *Stochastic & Recurrent Conditional-GAN (SR-cGAN)*, is illustrated in Fig. 4. The full version of SR-cGAN is composed of *image encoder* E_I, *image generator/decoder* G_I, RNN-based *temporal encoder* E_T, *generator/decoder* G_T, *and discriminator* D_T, respectively.

We now explain why the use of our SR-cGAN is able to synthesize video frames based on non-consecutive inputs. Moreover, how we preserve the recovered content and temporal consistency while exhibiting stochasticity in producing more than one possible video output will be also discussed.

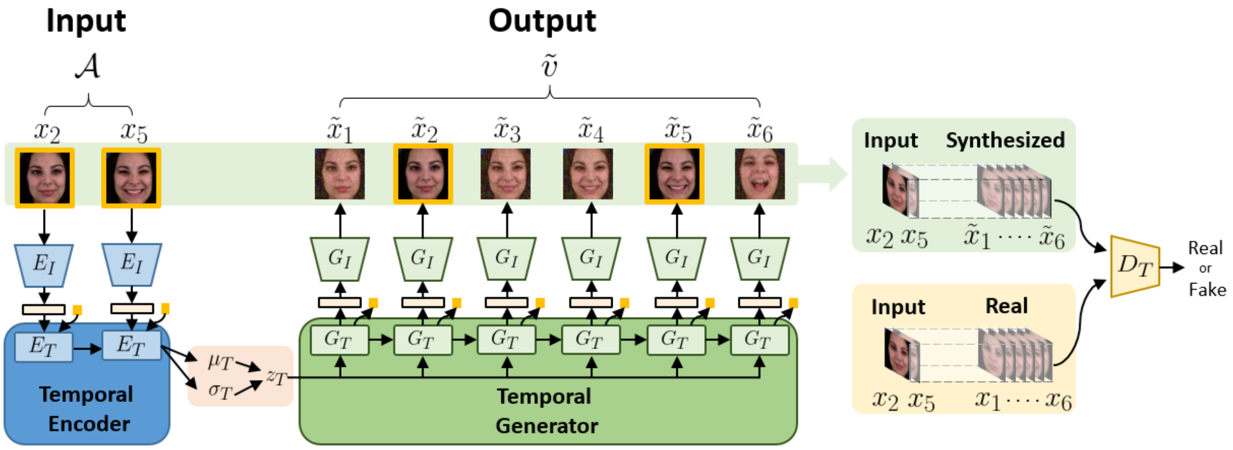


Fig. 4. Illustration of our proposed SR-cGAN for video inference. The architecture consists of RNN-based temporal encoder/generator and a discriminator, which is conditioned on the input anchor frames.

1) Stochastic Video Inference Module: Given few input anchor frames, it is desirable to observe more than one possible video sequence output. In order to model such temporal ambiguity across frames, we introduce a VAE-like stochastic video inference module into our SR-cGAN, which consists of *image encoder* E_I, *image generator/decoder* G_I, RNN-based *temporal encoder* and *generator/decoder* E_T and G_T.

The image encoder E_I first converts the input sequence of anchor frames $\mathcal{A}$ to frame-based representations $\{z_I^{(t_1)}, ..., z_I^{(t_M)}\}$, followed by the encoding of E_T which further maps the features into a video representation z_T by encoding a sequence of image representations $\{z_I^{(t_1)}, ..., z_I^{(t_M)}\}$ with time $\mathcal{T} = \{t_1, ..., t_M\}$. To be more precise, our temporal encoder E_T outputs the mean μ_T and standard deviation σ_T via observing the anchor frames, where the distribution of video representation z_T is given by $q(z_T|\mathcal{A}, \mathcal{T}) = \mathcal{N}(\mu_T(\mathcal{A}, \mathcal{T}), \sigma_T(\mathcal{A}, \mathcal{T}))$.

On the other hand, the generator G_T in our SR-cGAN decodes z_T to generate a sequence of image representations $\{\tilde{z}_I^{(1)}, ..., \tilde{z}_I^{(N)}\}$ at each time step $\{\tilde{t}_1, ..., \tilde{t}_N\}$. Such a sequence of image representations $\{\tilde{z}_I^{(1)}, ..., \tilde{z}_I^{(N)}\}$ can be regarded as a plausible trajectory in the latent space of video data. With the above definitions, the objective function $\mathcal{L}_{T_{VAE}}$ can be written as:

$$\begin{aligned}\mathcal{L}_{T_{VAE}}(E_I, E_T, G_T, G_I) = & -\mathbb{E}_{q(z_T|\mathcal{A},\mathcal{T})}[\log p_T(\tilde{v}|z_T)] \\ & + KL(q(z_T|\mathcal{A}, \mathcal{T})||p_T(z_T)),\end{aligned} \tag{9}$$

where the first term denotes reconstruction loss of all synthesized frames within the generated video $\tilde{v}$, and the prior distribution $p_T(z_T)$ in the second term follows $\mathcal{N}(\mathbf{0}, \mathbf{I})$ in our model.

To enforce the output video matching input anchor frames $\mathcal{A}$ at each corresponding time $\mathcal{T}$, the RNN cells in G_T needs to predict the correct time step at the associated time. In other words, the predicted time step $\tilde{t}_i$ should match the correct time step i for anchor frame x_i^a. Let y_i, $\hat{y}_i$ be the one-hot encodings of i and $\tilde{t}_i$, respectively. We have the *anchor loss* $\mathcal{L}_{anchor}$ with cross-entropy as follows:

$$\mathcal{L}_{anchor}(E_T, G_T) = -\sum_{i=1}^{N} y_i \log(\hat{y}_i) \tag{10}$$

which matches each anchor frame x_i^a and its corresponding timing t_i. Note that we consider N (the number of output frames) instead of M (the number of anchor frames) in (10), since the resulting anchor loss would not only enforce the anchor frames to match the corresponding frames in the output video, the overall temporal consistency of the output video can be preserved as well.

2) Observing Content Consistency in SR-cGAN: A successful video inference output should preserve the content information across video frames while matching the input anchor frames at the associated time. To achieve this goal, the component of conditional GAN in our SR-cGAN shows that we particularly have the discriminator D_T take the concatenation of anchor frames $\mathcal{A}$ and either the generated video $\tilde{v}$ or the real video v as the input. Thus, the resulting loss function $\mathcal{L}_{content}$ is defined as:

$$\begin{aligned}\mathcal{L}_{content}(G_I, G_T, D_T) = & \ \mathbb{E}_{v\sim p_{data}(v)}[\log D_T(v|\mathcal{A})] \\ & + \mathbb{E}_{z_T\sim q(z_T|\mathcal{A})}[\log(1 - D_T(G_I(G_T(z_T|\mathcal{A}))))].\end{aligned} \tag{11}$$

D. Learning of SR-cGAN

To sum up, the overall objective function of our proposed SR-cGAN for video inference can be written as below:

$$\begin{aligned}\underset{E_I,E_T,G_I,G_T}{min}\ \underset{D_I,D_T}{max}\ & \mathcal{L}(E_I, E_T, G_I, G_T, D_I, D_T) = \\ & \mathcal{L}_{I_{VAE}}(E_I, G_I) + \mathcal{L}_{I_{GAN}}(G_I, D_I) \\ & + \mathcal{L}_{T_{VAE}}(E_I, E_T, G_T, G_I) \\ & + \mathcal{L}_{anchor}(E_T, G_T) + \mathcal{L}_{content}(G_T, G_I, D_T).\end{aligned} \tag{12}$$

While our model can be trained in an end-to-end manner ideally, we observe that it would be desirable to first initialize/update frame-based network parameters of $\{E_I, G_I, D_I\}$, followed by their fine-tune and the update of the video-based model parameters $\{E_T, G_T, D_T\}$. Therefore, for the former initialization phase, the learning process aims to derive a proper image latent space, which treats each frame in the videos as separate images and update the parameters of $\{E_I, G_I, D_I\}$ with respect to the objective terms $\mathcal{L}_{I_{VAE}} + \mathcal{L}_{I_{GAN}}$, as noted in (4).

As for the latter training phase, we then focus on capturing the temporal structure of videos. This is achieved by *randomly* sampling M frames from each N-frame video as the input anchor frames $\mathcal{A}$ with its corresponding timing $\mathcal{T}$ to learn our video inference model. In other words, we update the parameters of $\{E_T, G_T, D_T\}$ and fine-tune $\{E_I, G_I\}$ with respect to $\mathcal{L}_{T_{VAE}} + \mathcal{L}_{anchor} + \mathcal{L}_{content}$.

IV. Experiments

In this section, we first demonstrate that the base model of our SR-cGAN is able to synthesize video frames from observed prior distributions. Then, we show the results of video inference generated by the full version of our SR-cGAN. We also evaluate the stochasticity of the results and the special cases of video inference can also be applied to solving the tasks of video interpolation/inpainting and video prediction.

A. Datasets

In our experiments, we consider a variety of video datasets: **Shape Motion**, **KTH**, and **MUG Facial Expression**, as their properties and settings discussed below.

1) Shape Motion – Synthesized Object Motion.: As proposed in the work of [14], this dataset contains videos of synthesized moving shapes, and is first used to verify the effectiveness of our model. In each video, the shapes (i.e., circle or square) with various colors and sizes move along different trajectories in front of a black background. The dataset consists of 8000 videos with the length of 32 frames and resolution of 64×64 pixels. We use 80% of sequences for training and the rest for testing.

2) KTH – Human Actions.: We further evaluate on KTH dataset [40] which contains 25 identities performing six different actions in four scenarios. We select *"hand-waving"* action as our target, and each video in the dataset is divided into clips of 16-frame length as a sequence in our experiments. In total, we extract 1855 sequences of hand-waving action with each frame resized to 64×64 pixels. The common rule of thumb 80/20 is used for the training/testing dataset split.

3) MUG – Facial Expressions.: MUG Facial Expression Database [41] covers various facial expressions of several identities, including: *anger, disgust, happiness, fear, surprise, and sadness*. We partition each video into sequences with the length of 16 frames and finally obtain 8,147 sequences in total for experiments. We use the same training/testing dataset split (80/20 respectively) as the other two datasets.

B. Video Synthesis from Prior Distribution

1) Qualitative experiments.: We first conduct video synthesis to verify the effectiveness of learned video representation z_T. As described in Sec. III-B, given a random sample z_T, our model is able to synthesize a corresponding video. From the results shown in Fig. 5, 6, and 7, our model obtains satisfactory results for all three datasets with temporal consistency. For instance, in Fig. 6, we observe the smooth and correct sequence of hand movements for the action of hand-waving, and in the first row of Fig. 7 the changes of facial expression from non-smiling to smiling are clearly shown.

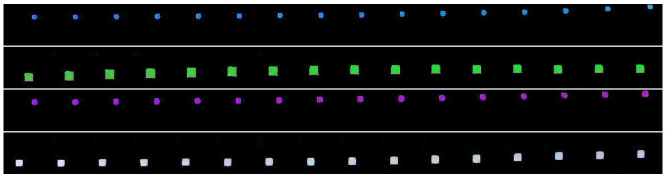


Fig. 5. Example video synthesis results of SR-cGAN trained by the **Shape Motion** dataset. Each row shows a synthesized video from a random sample z_T. Note that shape/color consistency and motion continuity can be observed within a video sequence.

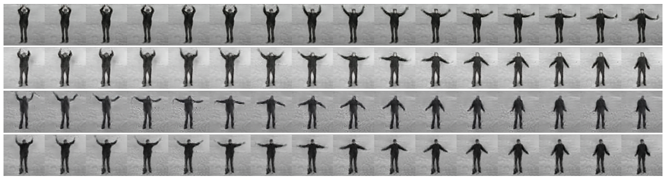


Fig. 6. Example video synthesis results of SR-cGAN trained by the **KTH** dataset. Each row shows a synthesized video sequence from a random sample z_T. Note that identity/content information is preserved within a video, while motion continuity can be observed.

2) Quantitative Comparison.: In order to evaluate the quality of synthesized videos, we adopt the Averaged Content Distance (ACD) metric used in MoCoGAN [14] to measure the content consistency among frames of a video sequence. We note that the reasonable video outputs need to exhibit identity/content consistency. For example, a generated video sequence for Shape Motion should contain the same object with the same color while moving. For Shape Motion dataset, with representing each frame by its mean color vector, ACD is calculated as the average of pairwise L2 distance across frames. While for the MUG dataset, the ACD is defined in the same way but with per-frame feature vectors produced by OpenFace [42].

We compare our SR-cGAN against several state-of-the-arts of video synthesis, including VGAN [12], TGAN [13], and MoCoGAN [14]. Following the same setting in [14], for each dataset we sample 256 different videos and compute the mean value of ACD. As listed in Table II, our SR-cGAN outperforms others in a clear margin, in which it shows that our model is able to better preserve the content consistency between frames of a synthesized video.

C. Video Frame Inference

1) Video Inference with Random Anchor Frames: Our proposed SR-cGAN aims to generate a complete video based on the condition provided by anchor frames and their specific time of occurrence. As shown in Fig. 8 and 9, we complete 16-frame videos based on 6 anchor frames (annotated by the yellow border) and 8-frame videos based on 3 anchor frames respectively, with specifying different settings of anchor timing for each row. The visualizations demonstrate the capability of

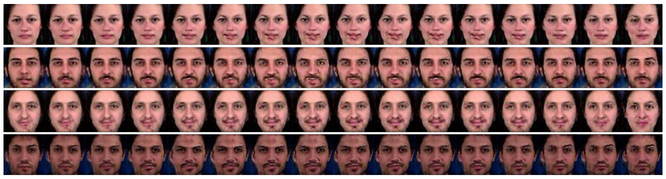


Fig. 7. Example video synthesis results of SR-cGAN trained by the **MUG** dataset. Each row shows a synthesized video from a random sample z_T. Note that identity/content information is preserved within a video, while motion continuity can be observed.

TABLE II
COMPARISONS OF CONTENT CONSISTENCY IN TERMS OF AVERAGE CONTENT DISTANCE (ACD) [14]. NOTE THAT REFERENCE CALCULATES ACD FROM THE TRAINING VIDEOS (NOT THE SYNTHESIZED ONES), AND THUS THE RESULTING SCORES CAN BE VIEWED AS LOWER BOUNDS.

ACD	Shape Motion	Facial Expressions
Reference	0	0.116
VGAN	5.02	0.322
TGAN	2.08	0.305
MoCoGAN	1.79	0.201
Ours	**1.05**	**0.137**

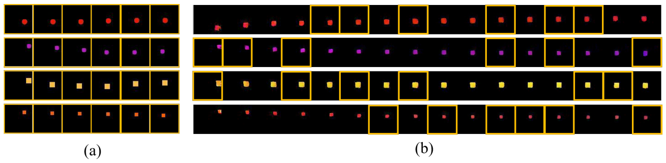


Fig. 8. Example video inference results for **Shape Motion**. For each row, six input anchor frames (in yellow bounding boxes) with the associated time are provided in (a), while the recovered video sequence is shown in (b).

our model for reasonably completing the temporal structure with respect to anchor frames.

2) Stochasticity: As described in Sec. III-C, one of the contributions for our SR-cGAN is to handle the stochastic dynamics within real-world videos. In order to verify this, we refer to the similar procedure used in [24] for evaluation. The basic idea behind this evaluation is to understand whether the stochasticity modeled by our framework covers the real dynamics happened in ground truth videos. To be more detailed, even our model generates videos based on a video representation z_T with noise perturbation, as the number of z_T increases, there should be higher chances that the ground truth video or the ones with a similar appearance are included within the group of generated videos. Therefore, the similarity between the ground truth video frames and their best matches from generated videos should be increasing with respect to the associated number of z_T.

In our experiment, we utilize PSNR (Peak Signal to Noise Ratio, [43]) as the metric to measure the similarity between video frames. We first define 100 distinct settings of assigning anchor frames. For each setting, we sample 200 z_T with noise perturbation, i.e., $z_T \sim \mathcal{N}(\mu_T(\mathcal{A}, \mathcal{T}), \sigma_T(\mathcal{A}, \mathcal{T}))$. By computing the PSNR numbers of the best matches found in different amounts of generated videos, we plot the changes of average PSNR values across all anchor settings in correspondence to the number of z_T samples. As observable in Fig. 10, in comparison to the deterministic model which uses z_T without noise perturbation, i.e., $z_T = \mu_T(\mathcal{A}, \mathcal{T})$, our stochastic model is verified accordingly to handle ambiguity in video inference. We further provide in supplementary materials the qualitative visualization of experimenting this stochasticity.

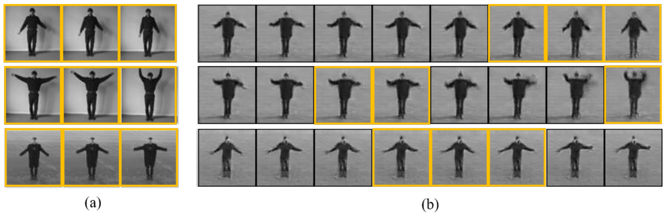


Fig. 9. Example video inference results for **KTH**. For each row, three input anchor frames (in yellow bounding boxes) with the associated time are provided in (a), while the recovered video sequence is shown in (b).

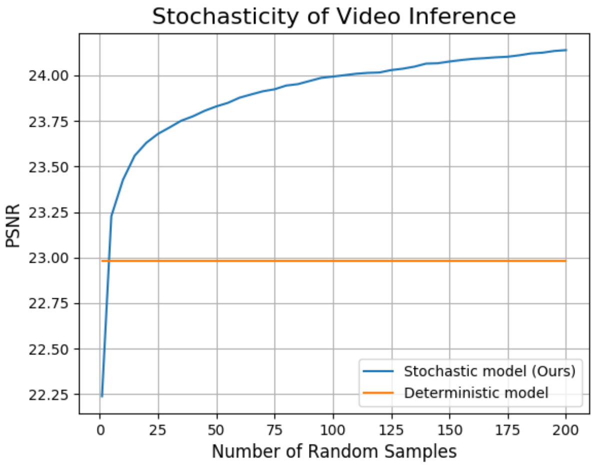


Fig. 10. Evaluation of stochasticity of video inference. Note that the orange curve shows the highest PSNR (compared to the ground truth video) reported by z_T with random noise perturbation. The blue line indicates the results of the deterministic baseline model using z_T without noise perturbation. With the introduced stochasticity, our model achieves improved video inference performances.

Note that it is crucial to evaluate the stochasticity of our model. If the stochasticity of a model cannot properly observe underlying data distribution, it would fail to generate plausible outcomes or results similar to ground truth. Not all generative models would exhibit satisfactory stochasticity performances. Moreover, a good video inference model is expected to synthesize various results given the same set of anchor frames. This is the reason why we adopt stochasticity in our model.

3) Comparison with Naive Linear Interpolation: In order to prove that the proposed SR-cGAN does capture the smooth and reasonable trajectories among latent image representations of video frames, we experiment to observe the temporal transition between two static frames. To be precise, we aim to generate the intermediate frames between two anchor frames which are positioned on the first and the last moments of a synthesized video. This can be regarded as a more general setting of video interpolation (synthesizing intermediate frames between two consecutive video frames with the number of intermediate frames more than one) or a more specific setting of inpainting (synthesizing intermediate frames between preceding and succeeding frames when the

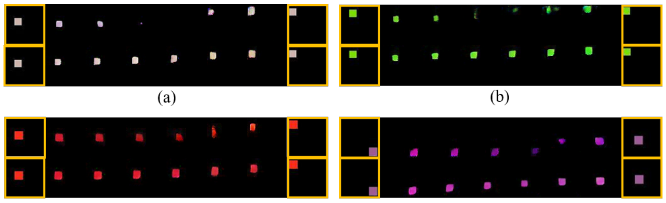


Fig. 11. Comparisons with naive video interpolation. Four interpolated videos are shown in (a) to (d), with starting and ending frames as anchors. In each sub-figure, the upper sequence is recovered by latent representation interpolation (via VAE-GAN), while the lower one is our output.

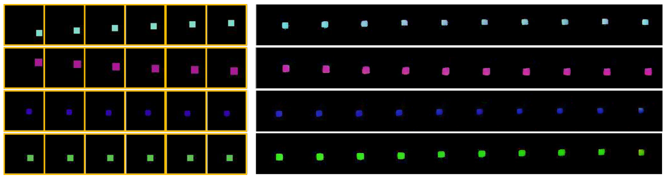


Fig. 12. Example video prediction results using the **Shape Motion** dataset. Note that each row shows six consecutive anchor frames (in yellow bounding boxes), followed by ten predicted future frames.

number of preceding and succeeding frame are only one, respectively). Moreover, one might consider that a reasonable trajectory can be easily approximated by piecewise linear paths connecting representations of conditioned inputs. As mentioned in Sec. II-E, the model proposed by [15] has tackled this setting. However, their model can only be applied to human motion videos.

We design a naive baseline which generates video frames decoded from linearly-interpolated samples between image representations of two anchor frames. As shown in Fig. 11, the comparison between the results produced by our model and the ones from the naive baseline clearly shows that our network is able to synthesize the frames with smooth motion and simultaneously preserve identity/content (i.e., color and shape) at each time step, while the naive baseline fails to achieve so. In other words, the video latent space learned in our approach successfully captures the distribution of reasonable trajectories over image representations.

4) Prediction of Future Frames: In this part of the experiments, we treat the anchor frames to be the first few frames observed, and thus video inference can be viewed as the task of video prediction for producing future consecutive frames. As shown in Fig. 12, given six anchor frames at the beginning of a target video, our SR-cGAN is able to predict the next ten frames. It can be seen that our model not only estimates the future motion trajectories but also preserves the consistency of content across output frames. It is worth noting that while recent video prediction works attempt to tackle more difficult scenarios (e.g. natural traffic scenes in [22]) than the ones used in this paper, those works are not able to solve other tasks (e.g. video synthesis and inference) under the same architecture.

V. Discussions

A. Importance of Content Consistency Loss

We note that enforcing the content consistency loss $\mathcal{L}_{content}$ in our proposed network architecture is necessary. From our ablation study, we observe that removing this term would result in mismatch between the content of input anchor frames and output frames.

B. Video Length and Computational Restrictions

While we do not limit the length of our synthesized video, the computation cost does practically restrict such implementation. In the case of video inference model with 6 anchor frames and 16 output frames, we need 11 GB GPU memory and takes about 72 hours to obtain satisfactory results on a GTX 1080Ti.

C. Sensitivity to the Speed/Amount of Motion

It is expected that videos with more dramatic motion changes would be more difficult to handle/synthesize. Thus, we do our best to include the video datasets for experiments exhibiting different speeds of motion. For instance, the KTH dataset has videos in which people waving hands at different speeds, while the moving speeds of the objects in the Shape Motion dataset also vary. While our model is able to infer the motion across the input anchor frames, we did observe that anchor frames with larger motion differences are more difficult to handle. Nevertheless, it can still be observed that larger amounts of motion in our inference results were presented, with the anchor frames were positioned on extreme time instants (e.g., the first and the last moments, as shown in Fig. 11).

VI. Conclusion

We proposed Stochastic and Recurrent Conditional GAN (SR-cGAN) for solving the task of video inference, which learns video representation in an RNN-based framework. Since more than one possible video sequence is expected given a fixed number of input anchor frames, we not only have our model preserve visual content within a recovered video, we also introduce the ability to handle temporal ambiguity during the inference process. In our experiments, in addition to satisfactory video inference results, we also applied our SR-cGAN to video interpolation/inpainting and prediction with promising qualitative and quantitative performances. Therefore, the use of our proposed model for the above tasks can be successfully verified.

Appendix A
Video Example Results

In the supplementary video[1], we provide several animations to demonstrate our results.

A. Video Synthesis

In the first part of the video, we train our model with Shape Motion, KTH, and MUG datasets and synthesize videos by sampling from different z_t.

[1] https://drive.google.com/file/d/1zFP-P7ktqIVSgcyjc8TQylYtyUHDjrPN/view?usp=sharing

B. Video Inference

The second part of the video aims to demonstrate the results of video inference trained on Shape Motion and KTH datasets. Moreover, we show the stochasticity of our model for video inference task. Given the same anchor frames, the model is able to generate different videos.

Appendix B
Implementation Details

A. Frame-Based VAE-GAN

In practice, we update each component with respect to its related loss terms alternately. For loss terms in $\mathcal{L}_{I_{VAE}}$ (i.e., Equation (2) in the main paper), we let $\mathcal{L}_{I_{prior}} = KL(q(z_I|x)||p_I(z_I))$ and $\mathcal{L}_{I_{recon}} = -\mathbb{E}_{q(z_I|x)}[\log p_I(x|z_I)]$. Let θ_{E_I}, θ_{G_I}, and θ_{D_I} be the parameters of image encoder, image generator, and image discriminator respectively. For each iteration, we found that satisfactory results were obtained with each component being updated once. We update θ_{E_I}, θ_{G_I}, and θ_{D_I} with following gradients:

$$\begin{aligned}
\theta_{E_I} &\xleftarrow{+} -\Delta_{\theta_{E_I}}(\mathcal{L}_{I_{prior}} + \lambda_1 \mathcal{L}_{I_{recon}}) \\
\theta_{G_I} &\xleftarrow{+} -\Delta_{\theta_{G_I}}(\mathcal{L}_{I_{GAN}} + \lambda_2 \mathcal{L}_{I_{recon}}) \\
\theta_{D_I} &\xleftarrow{+} +\Delta_{\theta_{D_I}}(\mathcal{L}_{I_{GAN}}),
\end{aligned} \tag{13}$$

where the coefficients λ_1 and λ_2 are set as 50 and 0.5, respectively. We use ADAM [44] as the optimizer with $\beta_1 = 0.5$ and $\beta_2 = 0.999$. The learning rate is set as 10^{-4} for updating θ_{E_I} and θ_{G_I} and 2×10^{-7} for updating θ_{D_I}.

B. SR-cGAN

For loss terms in $\mathcal{L}_{T_{VAE}}$ (i.e., Equation (9) in the main paper), we let $\mathcal{L}_{T_{prior}} = KL(q(z_T|\mathcal{A},\mathcal{T})||p_T(z_T))$ and $\mathcal{L}_{T_{recon}} = -\mathbb{E}_{q(z_T|\mathcal{A},\mathcal{T})}[\log p_T(\tilde{v}|z_T)]$. Let θ_{E_T}, θ_{G_T}, and θ_{D_T} be the parameters of temporal encoder, temporal generator, and temporal discriminator respectively. We also update each component once for each iteration. We update θ_{E_T}, θ_{G_T}, and θ_{D_T} with following gradients:

$$\begin{aligned}
\theta_{E_T} &\xleftarrow{+} -\Delta_{\theta_{E_T}}(\lambda_3 \mathcal{L}_{T_{prior}} + \lambda_4 \mathcal{L}_{T_{recon}} + \mathcal{L}_{anchor} + \mathcal{L}_{content}) \\
\theta_{G_T} &\xleftarrow{+} -\Delta_{\theta_{G_T}}(\mathcal{L}_{T_{recon}} + \mathcal{L}_{anchor} + \mathcal{L}_{content}) \\
\theta_{D_T} &\xleftarrow{+} +\Delta_{\theta_{D_T}}(\mathcal{L}_{content}),
\end{aligned} \tag{14}$$

where the coefficients λ_3 and λ_4 are set as 0.1 and 10, respectively. We also use ADAM [44] as the optimizer with $\beta_1 = 0.5$ and $\beta_2 = 0.999$. The learning rate is set as 2×10^{-4} for updating θ_{E_T} and θ_{G_T} and 2×10^{-5} for updating θ_{D_T}. We also fine-tune θ_{E_I} and θ_{G_I} with following gradients:

$$\begin{aligned}
\theta_{E_I} &\xleftarrow{+} -\Delta_{\theta_{E_T}}(\lambda_3 \mathcal{L}_{T_{prior}} + \lambda_4 \mathcal{L}_{T_{recon}} + \mathcal{L}_{anchor} + \mathcal{L}_{content}) \\
\theta_{G_I} &\xleftarrow{+} -\Delta_{\theta_{G_T}}(\mathcal{L}_{T_{recon}} + \mathcal{L}_{anchor} + \mathcal{L}_{content}),
\end{aligned} \tag{15}$$

C. Network Architecture

The network architecture for our frame-based VAE-GAN is listed in Table III. For the SR-cGAN, we implement E_T and G_T using two-layer GRU [45] network. After E_T we attach a Fully-Connected layer with activation size $256 \cdot 2$ and Leaky ReLU to predict the mean μ_T and the standard deviation σ_T. The architecture of D_T for our SR-cGAN is listed in Table IV. The slope of Leaky ReLU in our model is set as 0.2.

References

[1] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *arXiv:1411.1784*, 2014.

[2] G. Long, L. Kneip, J. M. Alvarez, H. Li, X. Zhang, and Q. Yu, "Learning image matching by simply watching video," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016.

[3] S. Niklaus, L. Mai, and F. Liu, "Video frame interpolation via adaptive convolution," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[4] ——, "Video frame interpolation via adaptive separable convolution," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017.

[5] S. Niklaus and F. Liu, "Context-aware synthesis for video frame interpolation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[6] S. Meyer, A. Djelouah, B. McWilliams, A. Sorkine-Hornung, M. Gross, and C. Schroers, "Phasenet for video frame interpolation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[7] H. Jiang, D. Sun, V. Jampani, M.-H. Yang, E. Learned-Miller, and J. Kautz, "Super slomo: High quality estimation of multiple intermediate frames for video interpolation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[8] Z. Liu, R. A. Yeh, X. Tang, Y. Liu, and A. Agarwala, "Video frame synthesis using deep voxel flow," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017.

[9] X. Sun, R. Szeto, and J. J. Corso, "A temporally-aware interpolation network for video frame inpainting," *arXiv preprint arXiv:1803.07218*, 2018.

[10] Y. Wexler, E. Shechtman, and M. Irani, "Space-time completion of video," in *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2007.

[11] V. Cheung, B. J. Frey, and N. Jojic, "Video epitomes," in *International Journal of Computer Vision (IJCV)*, 2008.

[12] C. Vondrick, H. Pirsiavash, and A. Torralba, "Generating videos with scene dynamics," in *Advances in Neural Information Processing Systems (NIPS)*, 2016.

[13] M. Saito, E. Matsumoto, and S. Saito, "Temporal generative adversarial nets with singular value clipping," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017.

[14] S. Tulyakov, M.-Y. Liu, X. Yang, and J. Kautz, "Mocogan: Decomposing motion and content for video generation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[15] H. Cai, C. Bai, Y.-W. Tai, and C.-K. Tang, "Deep video generation, prediction and completion of human action sequences," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

[16] J. He, A. Lehrmann, J. Marino, G. Mori, and L. Sigal, "Probabilistic video generation using holistic attribute control," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

[17] M. Mathieu, C. Couprie, and Y. LeCun, "Deep multi-scale video prediction beyond mean square error," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2016.

[18] C. Finn, I. Goodfellow, and S. Levine, "Unsupervised learning for physical interaction through video prediction," in *Advances in Neural Information Processing Systems (NIPS)*, 2016.

[19] R. Villegas, J. Yang, S. Hong, X. Lin, and H. Lee, "Decomposing motion and content for natural video sequence prediction," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2017.

[20] W. Lotter, G. Kreiman, and D. Cox, "Deep predictive coding networks for video prediction and unsupervised learning," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2017.

[21] R. Villegas, J. Yang, Y. Zou, S. Sohn, X. Lin, and H. Lee, "Learning to generate long-term future via hierarchical prediction," in *Proceedings of the International Conference on Machine Learning (ICML)*, 2017.

TABLE III
THE NETWORK ARCHITECTURE OF OUR FRAME-BASED VAE-GAN.

Encoder			
Component	Layer	Activation Size	Activ. Fun.
Input	-	$64 \times 64 \times 3$	-
4*E_I	5×5 Conv.	$32 \times 32 \times 64$	BN, ReLU
	5×5 Conv.	$16 \times 16 \times 128$	BN, ReLU
	5×5 Conv.	$8 \times 8 \times 256$	BN, ReLU
	FC	$1024 \cdot 2$	ReLU
Generator			
Input	-	1024	
5*G_I	FC	$8 \cdot 8 \cdot 256$	ReLU
	5×5 Conv.	$16 \times 16 \times 256$	BN, ReLU
	5×5 Conv.	$32 \times 32 \times 128$	BN, ReLU
	5×5 Conv.	$64 \times 64 \times 32$	BN, ReLU
	3×3 Conv.	$64 \times 64 \times 3$	Tanh
Discriminator			
Input	-	$64 \times 64 \times 3$	
6*D_I	5×5 Conv.	$64 \times 64 \times 32$	ReLU
	5×5 Conv.	$32 \times 32 \times 128$	BN, ReLU
	5×5 Conv.	$16 \times 16 \times 256$	BN, ReLU
	5×5 Conv.	$8 \times 8 \times 256$	BN, ReLU
	FC	512	BN, ReLU
	FC	1	Sigmoid

TABLE IV
THE NETWORK ARCHITECTURE OF D_T IN SR-CGAN.

Discriminator			
Input	-	$64 \times 64 \times 3$	
4*D_T	$4 \times 4 \times 4$ Conv.	$64 \times 64 \times 64 \times 64$	BN, Leaky ReLU
	$4 \times 4 \times 4$ Conv.	$32 \times 32 \times 32 \times 128$	BN, Leaky ReLU
	$4 \times 4 \times 4$ Conv.	$16 \times 16 \times 16 \times 256$	BN, Leaky ReLU
	FC	1	Sigmoid

[22] X. Liang, L. Lee, W. Dai, and E. P. Xing, "Dual motion gan for future-flow embedded video prediction," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017.

[23] E. Denton and V. Birodkar, "Unsupervised learning of disentangled representations from video," in *Advances in Neural Information Processing Systems (NIPS)*, 2017.

[24] M. Babaeizadeh, C. Finn, D. Erhan, R. H. Campbell, and S. Levine, "Stochastic variational video prediction," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2018.

[25] W. Liu, W. Luo, D. Lian, and S. Gao, "Future frame prediction for anomaly detection–a new baseline," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[26] J. X. B. Ni and Z. L. S. C. X. Yang, "Structure preserving video prediction," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[27] E. Denton and R. Fergus, "Stochastic video generation with a learned prior," in *Proceedings of the International Conference on Machine Learning (ICML)*, 2018.

[28] N. Wichers, R. Villegas, D. Erhan, and H. Lee, "Hierarchical long-term video prediction without supervision," in *Proceedings of the International Conference on Machine Learning (ICML)*, 2018.

[29] W. Liu, A. Sharma, O. Camps, and M. Sznaier, "Dyan: A dynamical atoms network for video prediction," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

[30] M. Oliu, J. Selva, and S. Escalera, "Folded recurrent neural networks for future video prediction," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

[31] W. Byeon, Q. Wang, R. K. Srivastava, and P. Koumoutsakos, "Contextvp: Fully context-aware video prediction," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

[32] T. Xue, J. Wu, K. L. Bouman, and W. T. Freeman, "Visual dynamics: Probabilistic future frame synthesis via cross convolutional networks," in *Advances in Neural Information Processing Systems (NIPS)*, 2016.

[33] Y. Jang, G. Kim, and Y. Song, "Video prediction with appearance and motion conditions," in *Proceedings of the International Conference on Machine Learning (ICML)*, 2018.

[34] Y. Li, C. Fang, J. Yang, Z. Wang, X. Lu, and M.-H. Yang, "Flow-grounded spatial-temporal video prediction from still images," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

[35] C. Yang, Z. Wang, X. Zhu, C. Huang, J. Shi, and D. Lin, "Pose guided human video generation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

[36] L. Zhao, X. Peng, Y. Tian, M. Kapadia, and D. Metaxas, "Learning to forecast and refine residual motion for image-to-video generation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

[37] A. B. L. Larsen, S. K. Sønderby, H. Larochelle, and O. Winther, "Autoencoding beyond pixels using a learned similarity metric," in *Proceedings of the International Conference on Machine Learning (ICML)*, 2016.

[38] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2014.

[39] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems (NIPS)*, 2014.

[40] C. Schuldt, I. Laptev, and B. Caputo, "Recognizing human actions: a local svm approach," in *Proceedings of the International Conference on Pattern Recognition (ICPR)*, 2004.

[41] N. Aifanti, C. Papachristou, and A. Delopoulos, "The mug facial expression database," in *International Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS)*, 2010.

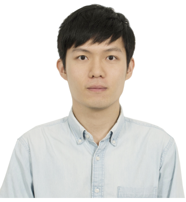
[42] B. Amos, B. Ludwiczuk, and M. Satyanarayanan, "Openface: A general-purpose face recognition library with mobile applications," 2016.

[43] Q. Huynh-Thu and M. Ghanbari, "Scope of validity of psnr in image/video quality assessment," *Electronics letters*, vol. 44, no. 13, pp. 800–801, 2008.

[44] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[45] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," in *Advances in Neural Information Processing Systems (NIPS)*, 2014.

Yu-Ying Yeh received the B.S. degree in Physics and B.A. degree in Economics from National Taiwan University in 2015. She is currently a Ph.D. student in Computer Science and Enginneering at University of California, San Diego. She was a research assistant in Department of Electrical Engineering, National Taiwan University from Aug. 2017 to Aug. 2018 and in Research Center for Information Technology Innovation, Academia Sinica, from Nov. 2016 to July 2017. Her current research interests mainly focus on machine learning and computer vision. She is also interested in representation learning, feature disentanglement, video prediction and 3d reconstruction.



Yen-Cheng Liu received the B.S. degree in Electrical and Computer Engineering from National Chiao Tung University in 2015. He received the M.S. degree in Electrical Engineering from National Taiwan University in 2017. He is currently a Ph.D. student in Machine Learning at Georgia Institute of Technology. He was a research assistant in Department of Electrical Engineering, National Taiwan University from Jul. 2017 to July 2018. His research interests are in the areas of computer vision and machine learning, particularly representation learning, transfer learning, and robotics perception.

Wei-Chen Chiu received the B.S. degree in Electrical Engineering and Computer Science and the M.S. degree in Computer Science from National Chiao Tung University (Hsinchu, Taiwan) in 2008 and 2009 respectively. He further received Doctor of Engineering Science (Dr.-Ing.) from Max Planck Institute for Informatics (Saarbrucken, Germany) in 2016. He joints Department of Computer Science, National Chiao Tung University as an Assistant Professor from August 2017 and leads the Enriched Vision Applications Laboratory. He was a postdoctoral researcher in Research Center for Information Technology Innovation, Academia Sinica, from Feb. to July. 2017, and a research scientist in a Taiwanese startup, Viscovery, from Aug. 2016 to Jan. 2017. His current research interests generally include computer vision, machine learning, and deep learning, with special focus on generative models.

Yu-Chiang Frank Wang received the B.S. degree in Electrical Engineering from the National Taiwan University, Taipei, Taiwan in 2001. He obtained the M.S. and Ph.D. degrees in electrical and computer engineering from Carnegie Mellon University, Pittsburgh, USA, in 2004 and 2009, respectively. Dr. Wang joined the Research Center for Information Technology Innovation (CITI), Academia Sinica, Taiwan, in 2009 as an assistant research fellow, and was later promoted as an associate research fellow in 2013. From 2015 to 2017, Dr. Wang also served as a Deputy Director of CITI at Academia Sinica.

In 2017, Dr. Wang joined the Graduate Institute of Communication Engineering and Department of Electrical Engineering at National Taiwan University, Taipei, Taiwan, as an associate professor, and is promoted to professor in 2019. He leads the Vision and Learning Lab at NTU, and focuses on research topics of computer vision and machine learning. He serves as Program Committee Members and Area Chairs at multiple international conferences or activities, and several of his papers were nominated for the Best Paper Awards at related international conferences such as IEEE ICIP, IEEE ICME and IAPR MVA. In 2013 and 2015, he was twice selected among the Outstanding Young Researchers by the Ministry of Science and Technology of Taiwan.