

# Video Imagination from a Single Image with Transformation Generation

Baoyang Chen, Wenmin Wang, Jinzhuo Wang, Xiongtao Chen  
 School of Electronics and Computer Engineering, Peking University

## ABSTRACT

In this work, we focus on a challenging task: synthesizing multiple imaginary videos given a single image. Major problems come from high dimensionality of pixel space and the ambiguity of potential motions. To overcome those problems, we propose a new framework that produce imaginary videos by transformation generation. The generated transformations are applied to the original image in a novel volumetric merge network to reconstruct frames in imaginary video. Through sampling different latent variables, our method can output different imaginary video samples. The framework is trained in an adversarial way with unsupervised learning. For evaluation, we propose a new assessment metric *RQA*. In experiments, we test on 3 datasets varying from synthetic data to natural scene. Our framework achieves promising performance in image quality assessment. The visual inspection indicates that it can successfully generate diverse five-frame videos in acceptable perceptual quality.

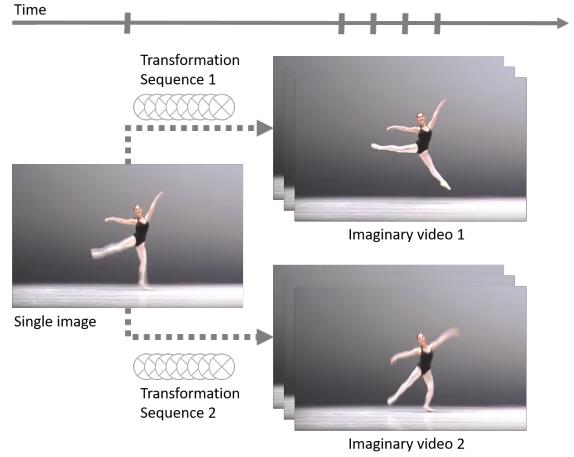
## KEYWORDS

Transformation Generation, Generative Models, Adversarial Training, Video Synthesis

## 1 INTRODUCTION

Given an static image, humans can think of various scenes of what will happen next using their imagination. For example, considering the ballerina in Figure 1, one can easily picture the scene of the dancer jumping higher or landing softly. In this work, we clarify the task as intimating human capability of ***Video Imagination***: synthesizing imaginary videos from single static image. This requires synthesized videos to be diverse and plausible. Although this study is still in its infancy, we believe video prediction and image reconstruction area can draw inspiration from it.

Compared to related tasks, e.g. video anticipation and prediction, there are more challenges for video imagination . Video imagination means to produce real high-dimension pixel values unlike low-dimension vectors in semantic anticipation. In addition, videos that are not identity to each other can all be reasonable, like imaginary video 1 and imaginary video 2 in Figure 1. So there is no precise ground truth as in common video prediction task. This intrinsic ambiguity makes regular criterion like MSE fails in evaluating whether the synthesized video is plausible. Moreover, compared to image generation, video synthesis needs to additionally model the temporal dependency that makes consecutive frames seem realistic.



**Figure 1: Synthesizing multiple imaginary videos from one single image.** For instance, given an image of a dancing ballerina, the videos of the dancer jumping higher or landing softly are both plausible imaginary videos. Those videos can be synthesized through applying a sequence of transformations to the original image.

Pioneers make attempts. Dense trajectory [34] and optical flow [22] have been used to model scene dynamics. Variational auto-encoder [34] and stochastic Markov-chain [24] have been introduced to form generative model. However, those models still struggle in high dimension space where manifold is hardly tractable, and is unsatisfying in terms of criterion

In this work, we present an end-to-end unsupervised framework with transformation generation for video imagination . Our key intuition is that we can model in transformation space instead of pixel space. Since scenes in frames are usually consistent, we assume that the major motions between frames can be modeled by transformations. If we reconstruct frame based on the original image and corresponding transformations, both scene dynamic and invariant appearance can be preserved well. In addition, we draw inspiration from image generation works [23] that use adversarial training. We believe an elaborate critic network that understands both spatial and temporal dependency would serve as reasonable criterion.

Based on the intuition and inspiration above, we design our framework focusing on model distributions in transformation space implicitly, and train it in adversarial way. In this framework, we generate transformation conditioned on the given image. Then we reconstruct each frame by applying the generated transformation to the given image. Latent variable is also introduced to enable

diverse sampling. Casting this into an adversarial architecture, we train our framework in a fully end-to-end fashion.

We believe this framework is a promising way to overcome existing challenges. As we build generation model in transformation space, it is more tractable to implicitly model the distribution of transformation. Conditioned on image makes generated transformation reasonable. The procedure of applying transformation to original image is similar to the insight of highway connection [10], and this helps the synthesized video maintaining sharp and clear. Also, the latent variable enables diverse imagination through sampling different transformations corresponding to different imaginary videos. Furthermore, there is nearly infinite resource for this unsupervised training. No label is needed, so every video clip can serve as a training sample.

For evaluation, since there is no general evaluation metrics for this task, we employ image quality assessment method to evaluate the quality of reconstructed frames and present a relative image quality assessment (*RQIA*) to eliminate the scene difference. In experiments, we evaluate our idea on three datasets, including two artificial video datasets with simple motions and one natural scene video dataset with complex motions. The synthesized 4-frames video results show that our framework can produce diverse sharp videos with plausible motions. We compare our framework with some related methods and two custom baselines. The quantitative evaluation results suggest that our framework outperforms others including those methods that are given more prior information, and the qualitative comparison also shows the advance of our synthesized videos.

The primary contribution of this paper is developing a new end-to-end unsupervised framework to synthesize imaginary videos from single image. We also make brave attempt on new evaluation method. In section 2, we review related work. In section 3, we present our *Video Imagination* video synthesis framework in details. In section 4, we illustrate new evaluation method *RQIA* and show experiments and comparison.

## 2 RELATED WORK

Although the works of future video synthesis from single image are rather little, our task shares common techniques with two related tasks: video prediction [27] and image reconstruction [32], where researchers have made impressive progress. In the following, we regard them as a universal visual prediction task, and review related works from different perspectives of approaches.

**Reconstruction in pixel space.** Early works of visual prediction focus on modeling and estimation in pixel space [38] [28] [37]. These methods reconstruct images by calculating pixel values directly. With recent resurgence of deep networks, researchers tend to replace standard machine learning models with deep networks. In particular, [14] proposes a video pixel network and estimates the discrete joint distribution of the raw pixel values. [26] uses LSTM network to learn representations of video and predict future frames from it. [33] employs adversarial training and generates video from scratch with deconvolution method [43]. A key issue in pixel-level prediction is the criterion metrics. A recent work [20] argues that standard mean squared error (MSE) criterion may fail with the inherently blurry predictions. They replace MSE in pixel

space with a MSE on image gradients, leveraging prior domain knowledge, and further improves using a multi-scale architecture with adversarial training.

**Mid-level tracking and matching.** To overcome the challenge of high dimensionality and ambiguity in pixel space, the prediction framework of mid-level elements gradually becomes popular. [19] explores a variation on optical flow that computes paths in the source images and copies pixel gradients along them to the interpolated images. [35] combines the effectiveness of mid-level visual elements with temporal modeling for video prediction. [24] defines a recurrent network architecture inspired from language modeling, predicting the frames in a discrete space of patch clusters. The input in [34] is a single image just like us, where the authors predict the dense trajectory of pixels in a scene with conditional variational autoencoder.

**Existing pixels utilization.** A insightful idea of improving the quality of prediction image is to utilize existing pixels. [18] synthesizes video frames by flowing pixel values from existing ones through voxel flow. [41] outputs the difference image, and produces the future frame by sum up the difference image and raw frame. [5] and [7] share a similar methods with us of applying filters to raw frames to predict new frames, and they provide the validation of gradients flow through filters.

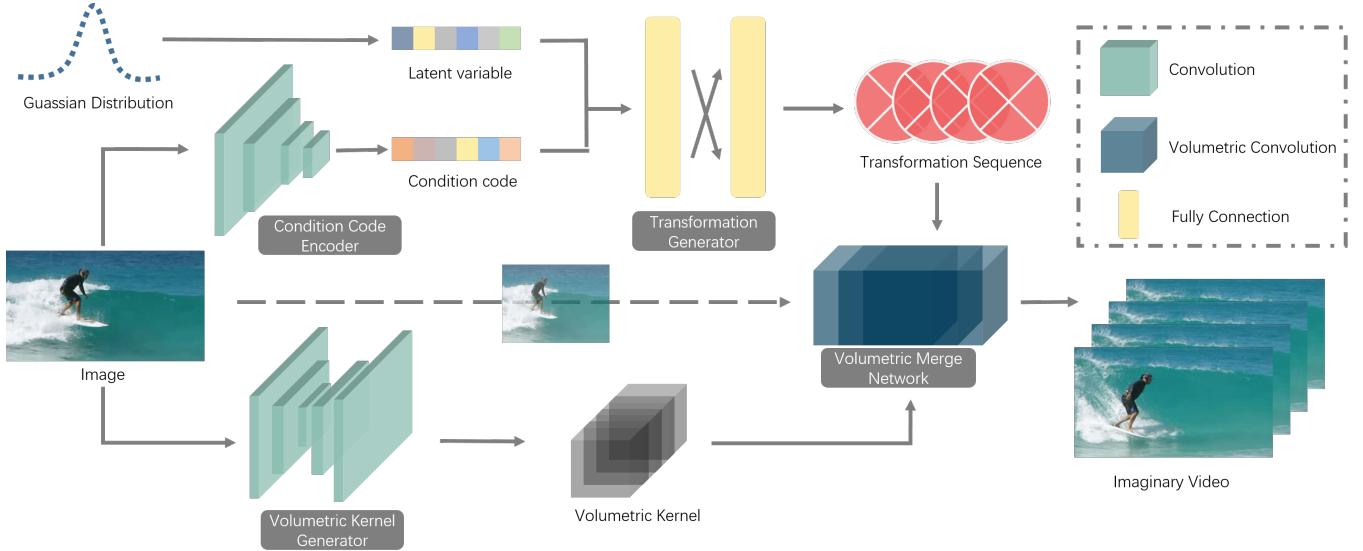
**Generation model evolutions.** Traditional works treat visual prediction as a regression problem. They often formulate prediction tasks with machine learning techniques to optimize the correspondence estimation [11, 16, 17]. With the development of deep networks, community of visual prediction has begun to produce impressive results by training variants of neural network structures to produce novel images and videos [9, 39, 40, 44]. The probabilistic models become popular again. More recently, generative adversarial networks (GANs) [8] and variational autoencoders [15] have been used to model and sample from distributions of natural images and videos [6, 23, 42]. Our proposed algorithm is based on GAN, but unlike previous works starting with a simple noise, we force our generation model conditioned on the given image, which benefits to generate reasonable transformation.

To the best of our knowledge there are no existing model that can produce multiple videos given one single image. Perhaps the most similar works to our task are [34, 41], where both works aim to build a probabilistic model of future given an image. But [41] only outputs one frame and [34] just produce optical flows.

Also, note a concurrent work that learns to predict in transformation space is [31], where the authors predict the new frames by predicting the following affine transformations. But their task is to generate frames from sequence of frames while ours is to synthesize imaginary videos given a single image. In addition, our work differs in that there methods are close to a regression problem as to predict precise future frames, but our task requires a probabilistic view and aims at generating multiple videos.

## 3 APPROACH

Rather than struggle in high-dimension pixel space, our idea is to model in transformation space for video imagination: to take one single image as input and synthesize imaginary videos that picture



**Figure 2: Pipeline of video imagination from single image.** In our framework, to produce one imaginary video, the input image is first encoded into a condition code and sent to transformation generator together with a latent variable. The generated transformation sequence is applied to input image later in volumetric merge network where frames are reconstructed with transformed images and volumetric kernels. Those four frames form one imaginary video. By sampling different latent variable from gaussian distribution, our framework can produce diverse imaginary videos.

multiple plausible scene change of that image. Figure 2 shows the pipeline of output one imaginary video in our framework.

In our framework, firstly, we send latent variable and condition code encoded from image into **transformation generator**, which outputs a group of transformation sequences. Secondly, we apply those transformation sequences to the original image and reconstruct frames through a **volumetric merge network**. Finally, we combine frames as an imaginary video then use **video critic network** to achieve adversarial training. In the following subsections, we firstly give a problem description; then we describe the details of those crucial parts and its implementation.

### 3.1 Problem definition

Firstly we use formulations to describe this task: given an image  $X$ , outputs  $m$  imaginary videos  $\hat{V}$  corresponding to different reasonable motions. Each imaginary video contains  $T$  consecutive frames  $f_T$ .

Ideally, we would like to model the distribution  $P(V | X)$  of all possible imaginary  $V$  given  $X$ . Practically, we aim to train a neural network  $T_\theta$  with parameters  $\theta$ , which implicitly models a distribution  $P_T(V | X)$ . Through training, we expect  $P_T(V | X)$  to converge to a good estimate of  $P(V | X)$ .  $T_\theta(X)$  yields a sample  $\hat{V}$  drawn from  $P_T(V | X)$ , so we have

$$\hat{V} = T_\theta(X) \sim P_T(V | X) \quad (1)$$

Instead of directly modeling in pixel space, we choose to model distribution in transformation space. We build this model based on a key assumption that the major motions between frames can be modeled by transformations. That means letting  $M_T$  denote motion between  $X$  and  $f_T$ ,  $M_T$  can be represented by a transformation sequence  $\Phi_T$  containing  $p$  transformations. Letting  $\odot$  denote the

operation of apply transformation sequence to image, we have  $f_T = \Phi_T \odot X$ . Letting  $\Phi$  represent the group of transformation sequences of all videos, we have  $V = \Phi \odot X$ . By introducing  $G$  implicitly modeling  $P_G(\Phi | X)$  in transformation space, we have a new description of target:

$$\hat{V} = G_\theta(X) \odot X \sim P_T(\Phi \odot X | X) = P_G(\Phi | X) \odot X \quad (2)$$

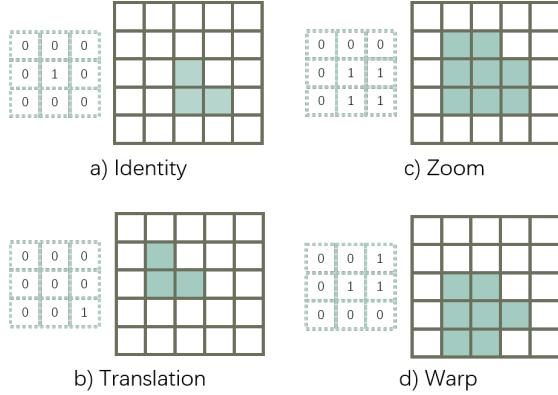
To make diversity samplings of  $V$  feasible, we introduce latent variable  $z$  that follows a specific distribution (e.g. Guassian Distribution). Hence, we can modify target of  $G_\theta$  from modeling the distribution  $P_G(\Phi | X)$  to modeling  $P_G(\Phi | X, z)$ . This implicit distribution allow us to sample different imaginary videos  $\hat{V}$  through sampling different  $z$ . Therefore, everything reduces to the following target:

$$\hat{V} = G_\theta(X, z) \odot X \sim P_G(\Phi | X, z) \odot X \quad (3)$$

### 3.2 Transformation Generator

The job of transformation generator is implicitly modeling  $P_G(\Phi | X, z)$  so that it can generate transformation conditioned on image. Given the condition code of a static image  $X$ , together with a latent variable  $z$ , the goal of transformation generator is learning to generate a transformation group  $\Phi$ .

To be specific, transformation generator outputs  $T$  transformation sequences  $\{\Phi_1, \Phi_2, \dots, \Phi_T\}$  corresponding to transformations between  $X$  and  $\{f_1, f_2, \dots, f_T\}$ . Each transformation sequence  $\Phi_T$  contains  $P$  transformations formed by  $K$  parameters. Transformations are generated in a sequential fashion in hope of a better description of warp motion, because motion can often be decomposed in a layer-wise manner.



**Figure 3: Different convolution kernels result in different motions.** The dotted square denotes convolution kernel and the right side image shows the result of applying the kernel. One simple kernel can model motion like b) translation c) Zoom d) Warp.

We use two kinds of transformations to model motion. Since in adversarial training, the gradient back-propagation starts from critic network then flows to the frames, the transformation type we choose needs to allow gradient propagating from transformed images to transformation generator. Fortunately, prior works in [5, 12] revealed that there is a group of transformations having this adorable attribution. We build two distinct models to form  $\Phi$  based on prior works.

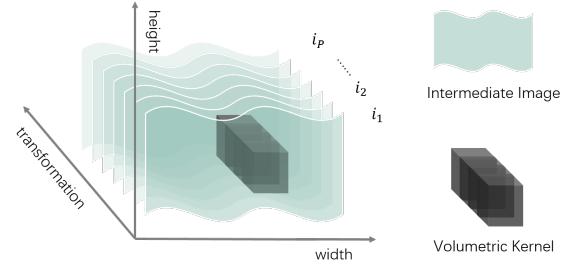
**Affine Transformation:** Simply formed by 6 parameters, affine transformation can model motions including translation, rotation, zoom, and shear. Works in [3, 36] have shown that affine transformation provides a good approximation of 3-D moving objects motion. Affine transformation works on coordinates, which would raise a problem of undefined pixel locations. In practice, we use differentiable bilinear interpolation to complete those pixels.

**Convolutional Transformation:** A convolution kernel can naturally model simple motions like translation, zoom and warp as shown in Figure 3. The kernel size can vary with application scene. For example, a  $5 \times 5$  kernel allows pixels translating over a distance of 2 pixels. A sequence of kernel would raise the size of receptive field and allow more complex or intenser motions.

### 3.3 Volumetric Merge Network

Volumetric merge network is responsible for reconstructing frames  $\{f_1, f_2, \dots, f_T\}$  based on the generated transformation  $\Phi$  and image  $X$ . The transformation group  $\Phi$  is finally applied to image  $X$ , producing an intermediate image group  $I$  consisting of  $T$  intermediate image sequences  $\{I_1, I_2, \dots, I_T\}$  that will be used to reconstruct  $\{f_1, f_2, \dots, f_T\}$  accordingly. Combining frames temporally, volumetric merge network outputs imaginary video  $\hat{V}$ .

Since the transformation is generated in a sequential fashion, it is intuitive to take the sequence of intermediate images as an extended dimension representing transformation. That is, we consider each transformed sequence  $I_T$  as one entity  $I_T \in \mathbb{R}^{W \times H \times P}$  that has 3 dimensions as width  $W$ , height  $H$ , and transformations  $P$ . This 3-D entity, as shown in Figure 4, allows us to reconstruct frame by



**Figure 4: Intermediate image sequence  $I_T$  as 3-D entity.** A volumetric kernel can take both neighbor pixel values and intermediate image differences into consideration.

merging it in a volumetric way. Each pixel is reconstructed through volumetric kernels. The kernels can take both neighbor pixel values and intermediate image differences into consideration.

Parameters in volumetric kernels can be obtained either from clipping a crop of generated transformations or through a specific volumetric kernel generator as shown in Figure 2. Volumetric kernel generator (a full convolution network) concentrates more on capturing the dependency in spatial domain, while generated transformations can give volumetric kernel better understanding of correlation between intermediate images.

### 3.4 Video Critic Network

To meet the requirement of a better criterion, we design a video critic network *Critic* to achieve adversarial training. Video critic network *Critic* receives synthesized video  $\hat{V}$  and real video  $V$  as input alternatively, and outputs criticism judging how convincing the input is.

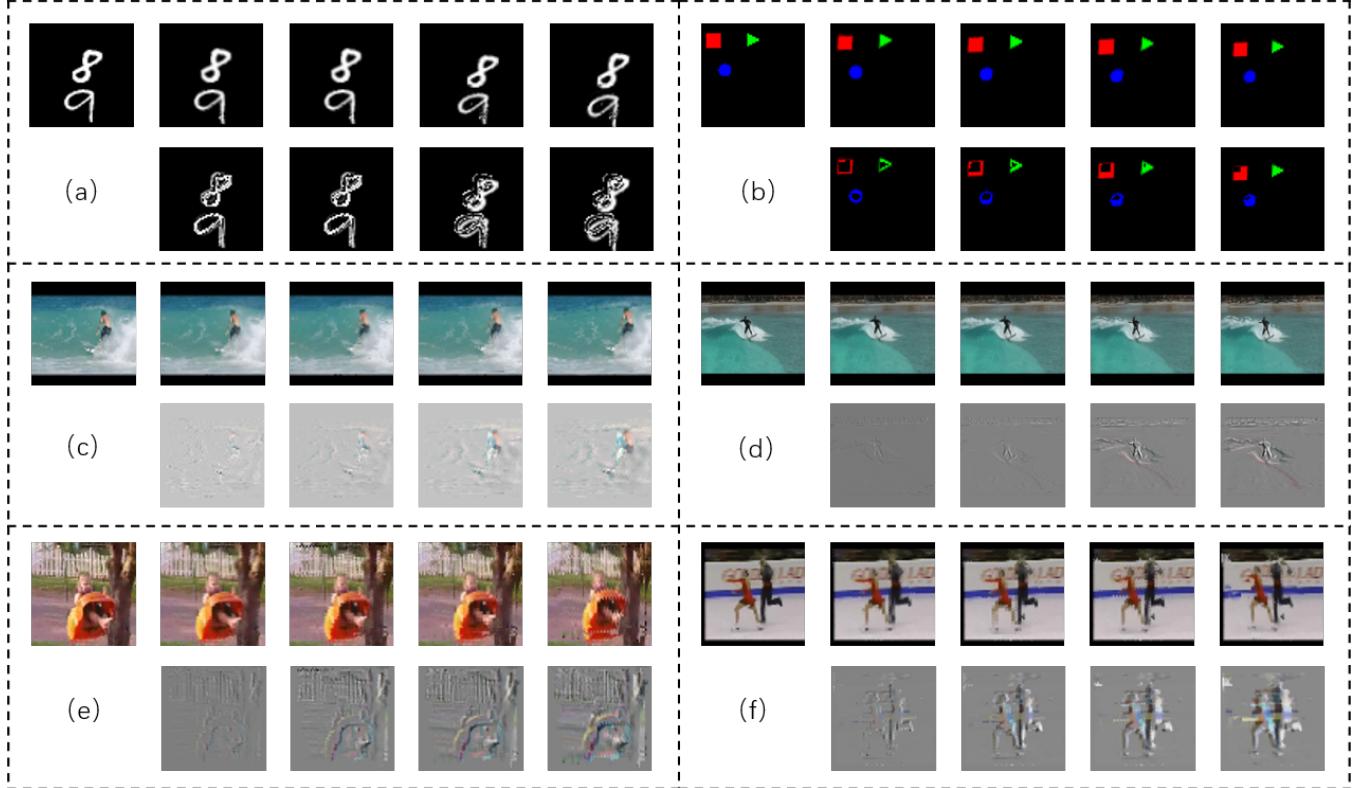
A convincing video means that the frame looks clear and the motion between frames seems consecutive and reasonable. Video critic network *Critic* needs to give reference of whether the input is plausible and realistic, which requires understanding of both static appearance and dynamic scene. The similar requirement can be found in action recognition task, where lately researchers have made progress [30]. We draw inspiration from those works, and design *Critic* to have the structure of spatial-temporal convolution networks [13].

### 3.5 Learning and Implementations

Our framework consists of fully feed-forward networks. The transformation generator consists of 4 fully connected layers. The latent code sampled from a gaussian distribution has 100 dimensions, and the condition code has 512 dimensions. We can encode  $X$  into condition code either through refined AlexNet or a 5 layer convolutional network. The volumetric merge network consists of 3 volumetric convolutional layers, while the last layer uses element-wise kernel. We use a five-layer spatio-temporal convolutional network as the critic network.

We employ Wasserstein GAN [2] to train our framework. The generator loss  $L_g$  is defined as:

$$L_g = -\mathbb{E}_{v \sim P_T(V|X)} \text{Critic}(v) \quad (4)$$



**Figure 5: Quality Performance of our framework.** In each dotted box, the first shows the synthesized imaginary videos given the fist frame as input. The second row shows the difference images of synthesized frames and input. (a)(b) demonstrate the results experiment on moving MNIST and 2D shapes dataset. (c)(d) shows the result of surfing class on UCF101 dataset in different resolution as c)  $64 \times 64$  and d)  $128 \times 128$ . (e)(f) shows the results given image from swing and ice-dancing categories in UCF101 dataset. The synthesized frames are sharp and clear. Difference images illustrate plausible motions. Results of different resolutions and different image categories on UCF101 dataset suggest our framework shows scale to the complexity of high-resolution videos.

The critic loss  $L_c$  is defined as:

$$L_c = \mathbb{E}_{v \sim P_T(V|X)} C(v) - \mathbb{E}_{v \sim P(V|X)} Critic(v) \quad (5)$$

Alternatively, we minimize the loss  $L_g$  once after minimizing the loss  $L_d$  5 times until a fixed number of iterations. Ultimately, the optimal video critic network  $C$  is hoped to produce good estimate of Earth-Mover (EM) distance between  $P(V | X)$  and  $P_T(V | X)$ . We use the RMSProp optimizer and a fixed learning rate of 0.00005. ReLU activation functions and batch normalization are also employed.

We use a Tesla K80 GPU and implement the framework in TensorFlow [1]. Our implementation is based on a modified version of [23], and the code can be found at the project page<sup>1 2</sup>. Since we model in relatively small transformation space, the model converges faster than others. Training procedure typically takes only a few days even hours depending on datasets.

## 4 EXPERIMENT

In this section, we experiment our framework on 3 video datasets: Moving MNIST [26], 2D shape [41] and UCF101 [25]. For evaluations, we perform qualitative inspection and novel quantitative assessment RIQA to measure the objective quality of the imaginary video.

### 4.1 Baselines and Competing Methods

Current work about this task is quiet limited. To find out whether our framework outperforms those methods that do not involve our crucial components, we develop two simple but reasonable baselines for this task. For the first one, **Baseline 1**, the transformation generator and volumetric merge network in our original framework are replaced by a generator network that directly outputs flatten pixels. For the second one, **Baseline 2**, the whole adversarial training procedure including critic network is removed, and the network is trained minimizing  $l_2$  loss function. Those two baselines can also be considered as a form of ablation experiments.

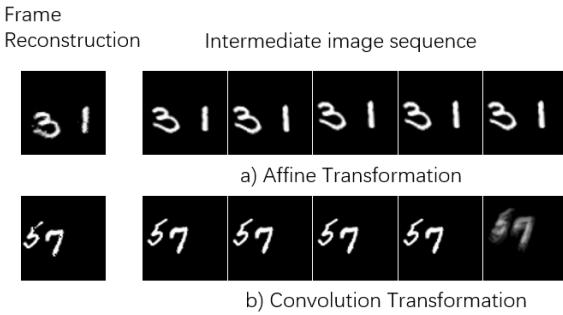
We also consider several latest works as competing methods as shown in 1. The task setting is distinct, so it is difficult to find evaluation metrics that can fairly compare all those works together,

<sup>1</sup><https://github.com/gipub327/VideoImagination>

<sup>2</sup>This page contains no information about the authors

**Table 1: Task setting comparison of related work.** Multiple output means that the method build a probabilistic model and can sample different results. \* indicates the methods can also experiment on natural scenes like in UCF101 dataset.

Model	Input	Output
Ours *	image	5 frames(multiple)
Visual Dynamic [5]	image	1 frame(multiple)
Scene Dynamic [33]	image	32 frames
Dynamic Filter [5]	4 frames	1 frame
Beyond MSE [20] *	4 frames	1 frame
Video Sequences [31] *	4 frames	4 frames



**Figure 6: Intermediate image sequences visualization.** Different transformation models result in different intermediate image sequences  $I_T$ . Each intermediate image represent one mode of simple transformations. A sequence of these intermediate image can form more complex motion.

but we make brave attempt later in Section 4.5 to compare our framework against some of those works.

## 4.2 Moving MNIST Dataset

**Dataset:** We first experiment on a synthetic grey video dataset: moving MNIST dataset [26]. It consists of videos where two MNIST digits move in random directions with constant speed inside a  $64 \times 64$  frame. The 64,000 training video clips and 320 testing clips are generated on-the-fly. Each video clip consists of 5 frames. Taking the first frame as input, the goal is to synthesize multiple imaginary 5-frames videos.

**Setup:** There is barely no pre-processing in our work except for normalizing all videos to be in range  $[0, 1]$ . We experiment on two transformation models. For convolutional transformation we set kernel size as  $9 \times 9$ , and the transformation sequence length  $P$  is set as 5 for both models. We generate 4 transformation sequences  $\{\Phi_1, \Phi_2, \Phi_3, \Phi_4\}$  corresponding to 4 consecutive frames  $\{f_1, f_2, f_3, f_4\}$  at once.

**Result:** Figure 5 (a) illustrates the qualitative performance in moving MNIST dataset. As we can see, frames are sharp and clear while the shape information of digits is well preserved as we expect. The difference images show that the generated transformations successfully model one motion mode so that the synthesized

imaginary video has plausible consecutive motion. Figure 6 shows reconstructed frames and the corresponding intermediate image sequences in different transformation models.

## 4.3 Synthetic 2D Shapes Dataset

**Dataset:** We experiment our framework using a synthetic RGB video dataset: Synthetic 2D Shapes Dataset [41]. There are only three types of objects in this dataset moving horizontally, vertically or diagonally with random velocity in  $[0, 5]$ . 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. There are 20,000 clips for training and 500 for testing just like settings in [41]. We aim at synthesizing multiple imaginary videos each containing five consecutive frames.

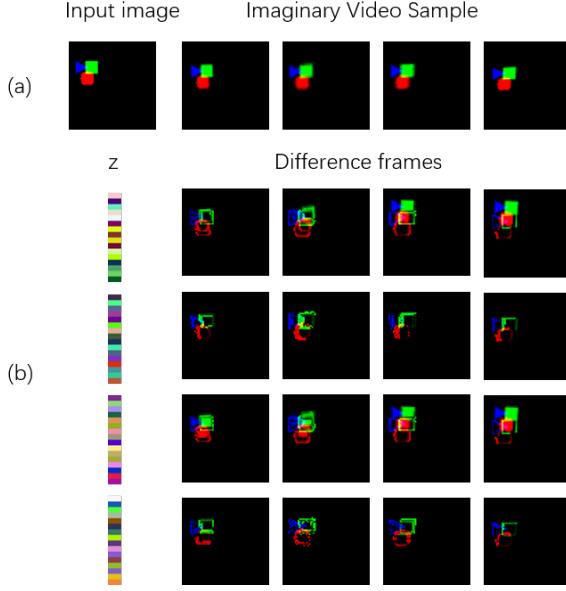
**Setup:** The input image size is set as  $64 \times 64$  so that we can inherit the network architecture and settings in section 4.2. The transformations applied to each color channel are set to be identical for the consistent of RGB channels.

**Result:** Figure 5 (b) illustrates the qualitative performance in 2D shape dataset. Appearance information including color and shape is reconstructed at a satisfying level, and the motion is plausible and non-trivial. Multiple sampling results are shown in Figure 7. It is clear that sampling different  $z$ s lead to different imaginary videos with the same input image. Motions in those videos are notably dissimilar. Figure 8 gives an perception comparison among our framework and two baselines. The three methods are trained in same iteration. Obviously, generation from scratch as Baseline 2 needs much longer training time and  $l_2$  loss criterion as Baseline 1 not only make the result lacking of diversity, but also leads to blur because of intrinsic ambiguity of image.

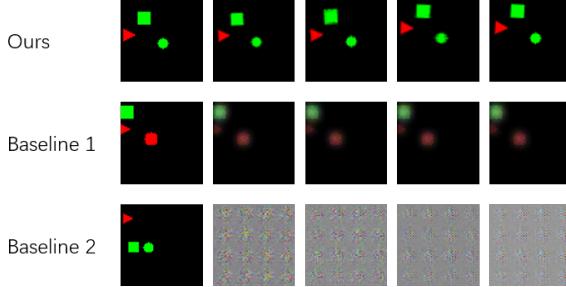
## 4.4 UCF 101 Dataset

**Dataset and setup:** The former datasets are both synthetic datasets. For natural scene, we experiment on UCF101 dataset [25]. The dataset contains 13,320 videos with an average length of 6.2 seconds belonging to 101 different action categories. The original dataset are labeled for action recognition, but we do not employ those labels and instead we use the dataset in an unsupervised way. Videos with an average length of 6.2 seconds are cut into clips that each consists of five frames. We prepare 15,680 video clips for each category as training samples and 1,000 unseen image as testing samples. The video frames are reshaped to  $128 \times 128$  and  $64 \times 64$  for different resolution experiments. The convolutional kernel size is set to 16 and 9 accordingly.

**Result:** Figure 5 (c)(d) illustrate the qualitative performance in surfing class of different resolutions. Obviously our framework produce fairly sharp frames. It successfully escapes from appearance deformation of surfer and wave. The difference images suggest that our framework can model plausible waving and surfing motions. The dynamic results seem rather realistic, so we strongly recommend a quick look at the small gif demo in supplementary material. Figure 9 shows the convergence curve of EM distance. We can see the curve decrease with training and converge to a small constant.



**Figure 7: Diverse video imagination: multiple imaginary videos from same input image.** (a) denotes the input image and one imaginary video sample as reference. The first column of (b) indicates different input  $z$ s, the rest columns shows the difference frames of imaginary video samples minus the reference. Each row of (b) illustrates a unique imaginary video and its unique  $z$ .



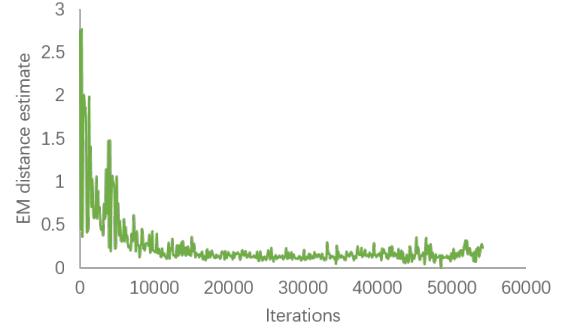
**Figure 8: Synthesis result of custom Baselines** With same fixed training iterations, our framework produce obviously better result.  $l_2$  loss in baseline 1 brings blur. Baseline 2 that reconstruct pixel from noise needs much longer training time and cannot produce recognizable frames.

The absolute of the constant is meaningless because the scale of EM distance varies with architecture of critic network.

#### 4.5 Evaluation and Comparison:

As shown in Table 1, there is no existing work shares the completely same task settings as ours. To make fair comparison to other works and baseline, we perform both qualitative inspection and novel quantitative evaluation.

**Frame quality assessment.** Quantitative evaluation of generative models is a difficult, unsolved problem [29]. The video imagination task is a multi-modality problem. But traditional full reference



**Figure 9: Curve of EM distance estimate at different steps of training.** The estimation of EM distance is done by a video critic network  $C$  that is well trained. We can see that the EM distance decrease and converge with training.

**Table 2: quantitative evaluation comparison among related visual prediction work.** The lower RIQA indicates better frame reconstruction quality. The BRISQUE score obviously varies with scenes and resolutions. RIQA points out the decreasing proportion between input and output, hence successfully reflects the reconstruction quality.

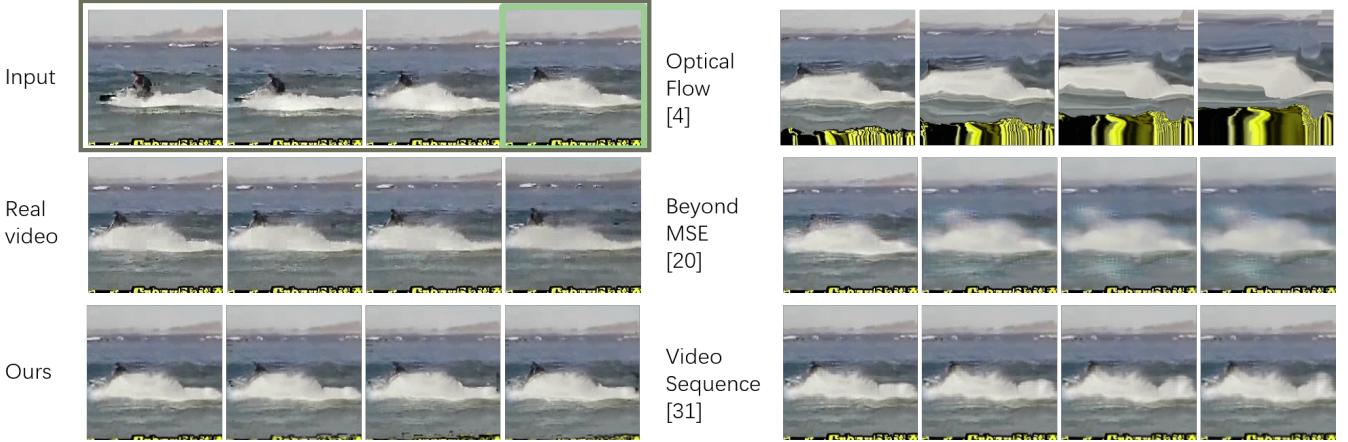
Methods	Input BRISQUE	Output BRISQUE	RIQA
Ours 64 × 64	45.2164	47.0168	3.98%
Ours 128 × 128	35.9809	36.7120	<b>2.03%</b>
Baseline 1	45.2164	50.7681	12.28%
Baseline 2	45.2164	89.2315	97.34%
Optical Flow [4]	39.3708	40.8481	3.75%
Beyond MSE [20]	46.3219	50.0637	9.24%
Video Sequences [31]	39.3708	42.8834	8.92%

image quality assessment methods (*FIQA*) requires a precise ground truth image as reference hence they are no longer appropriate. We employ popular Blind Image Quality Assessment(*BIQA*) method *BRISQUE*[21] as our non-reference quantitative evaluation metric.

Since *BRISQUE* is based on natural scene statistic, it is not applicable in synthetic image. we implement it on those methods that can synthesize natural scene images in UCF101 dataset [4, 20, 31]. A key problem of employing this metric is that the scenes and resolutions of the synthesized videos may be varied, so it is unfair to make comparison among those samples directly. Fortunately, the quality of the input image can be a solid quality reference. We calculate the decreasing proportion of quality score between inputs and outputs, and take it as our assessment metric: Relative image quality assessment (*RIQA*).

$$RIQA = \frac{BRISQUE(\text{Input}) - BRISQUE(\text{Output})}{BRISQUE(\text{Input})} \quad (6)$$

It is fair and reasonable because *RIQA* eliminates the natural quality differences between scenes and resolutions while have the ability of reflecting the crucial reconstruction quality well.



**Figure 10: Perceptual Comparison among related works using UCF101 dataset.** The input frames are from Skijet class. The output frames are reshaped to same size for a fair visual inspection. Notice that our framework only takes one frame as input (the green square) while the rest methods take four frames as input (the grey rectangle). Our result are sharp and relatively clear while the motions of rider and skijet are recognizable and plausible.

**Table 3: Analysis of the settings of models and hyper-parameters.**  $K$  refers to the number of parameters forming transformation.  $P$  refers the sequence length of transformation for reconstructing one frame.

Settings	RIQA
affine transformation with $K = 6$ and $P = 5$	<b>2.03%</b>
affine transformation with $K = 6$ and $P = 10$	4.79%
convolutional transformation with $K = 8 \times 8$ and $P = 5$	4.03%
convolutional transformation with $K = 16 \times 16$ and $P = 5$	4.01%

As shown in Table 2, diversity of the scenes and resolutions makes the raw BRISQUE score not comparable, but the RIQA tells the reconstruction quality change. We can see that our framework outperform other methods, and the poor performances of baselines suggest the architecture of our framework is reasonable. In addition, our framework and Video Sequence[31], that are based on transformation space, do produce images with better qualities than [20], which reconstruct frames from scratch.

Table 3 shows the results when we change the hyper-parameters and some model settings, including the number of parameters  $K$  forming transformation, the sequence length of transformation  $P$  for reconstructing one frame, and the type of transformations. The results demonstrate that our framework is overall robust to those choice. It seems that affine transformation model with transformation sequence length  $P = 5$  can achieve the best performance.

**Qualitative inspection.** Figure 10 shows the perceptual comparison between our framework and three competing methods [4, 20, 31] that also experimenting on UCF101 dataset. Our framework produces four frames conditioned on one frame while other

methods take a sequence of four frames as inputs. The simple optical method [4] fails due to the strong assumption of constant flow speed, yet it perform relatively better in quantitative evaluation because the image get weird but still maintain sharp. Beyond MSE [20] maintains some appearance but still struggles in deformation and blur. The transformation-based model [31] provides fairly recognizable result but also gets blurry. Considering [31] actually takes four frames as input and aims to predict future frames, the motion looks less consecutive and convincing. Our framework synthesizes sharp and recognizable frames, and the dynamic scene looks realistic and plausible. The motion in our result (wave raising) is not identity to motion in the real video (wave falling), this is because the intrinsic ambiguity of one single image. Notice that the yellow symbols on the bottom turn to pieces in our framework while in [31] it remains still. We believe this is because [31] splits frame into patches so gains better description of patch variance.

**Failure Case.** A typical failure case in affine transformation model is that the motions between frames are plausible yet unexpected black pixels appear somewhere in the frames. We think this is caused by the empty pixels in intermediate images after applying affine transformations. In convolution model, one common failure mode is that some part of the objects lack resolution while the silhouettes remain recognizable. We believe a more powerful merge network would be a promising solution in both cases, and we leave this for future work.

## 5 CONCLUSION

In this paper, we have presented a new framework to synthesize multiple videos from one single image. Specifically, our framework uses transformation generation to model the motions between frames, and reconstructs frames with those transformations in a volumetric merge network. We also present a novel evaluation metric to assess the reconstruction quality. We have demonstrated that our framework can produce plausible videos with state-of-the-art image quality on different datasets.

## REFERENCES

- [1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, and others. 2016. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467* (2016).
- [2] Martin Arjovsky, Soumith Chintala, and Léon Bottou. 2017. Wasserstein gan. *arXiv preprint arXiv:1701.07875* (2017).
- [3] James R Bergen, Peter J Burt, Rajesh Hingorani, and Shmuel Peleg. 1990. Computing two motions from three frames. In *Computer Vision, 1990. Proceedings, Third International Conference on*. IEEE, 27–32.
- [4] Thomas Brox, Andrés Bruhn, Nils Papenberg, and Joachim Weickert. 2004. High accuracy optical flow estimation based on a theory for warping. *Computer Vision-ECCV 2004* (2004), 25–36.
- [5] Bert De Brabandere, Xu Jia, Tinne Tuytelaars, and Luc Van Gool. 2016. Dynamic filter networks. In *Neural Information Processing Systems (NIPS)*.
- [6] Emily L Denton, Soumith Chintala, Rob Fergus, and others. 2015. Deep Generative Image Models using a Laplacian Pyramid of Adversarial Networks. In *Advances in neural information processing systems*, 1486–1494.
- [7] Chelsea Finn, Ian Goodfellow, and Sergey Levine. 2016. Unsupervised learning for physical interaction through video prediction. In *Advances In Neural Information Processing Systems*. 64–72.
- [8] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Advances in neural information processing systems*. 2672–2680.
- [9] Karol Gregor, Ivo Danihelka, Alex Graves, Danilo Jimenez Rezende, and Daan Wierstra. 2015. DRAW: A recurrent neural network for image generation. *arXiv preprint arXiv:1502.04623* (2015).
- [10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Identity mappings in deep residual networks. In *European Conference on Computer Vision*. Springer, 630–645.
- [11] Minh Hoai and Fernando De la Torre. 2014. Max-margin early event detectors. *International Journal of Computer Vision* 107, 2 (2014), 191–202.
- [12] Max Jaderberg, Karen Simonyan, Andrew Zisserman, and others. 2015. Spatial transformer networks. In *Advances in Neural Information Processing Systems*. 2017–2025.
- [13] Shuiwang Ji, Wei Xu, Ming Yang, and Kai Yu. 2013. 3D convolutional neural networks for human action recognition. *IEEE transactions on pattern analysis and machine intelligence* 35, 1 (2013), 221–231.
- [14] Nal Kalchbrenner, Aaron van den Oord, Karen Simonyan, Ivo Danihelka, Oriol Vinyals, Alex Graves, and Koray Kavukcuoglu. 2016. Video pixel networks. *arXiv preprint arXiv:1610.00527* (2016).
- [15] Diederik P Kingma and Max Welling. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114* (2013).
- [16] Kris M Kitani, Brian D Ziebart, James Andrew Bagnell, and Martial Hebert. 2012. Activity forecasting. In *European Conference on Computer Vision*. Springer, 201–214.
- [17] Tian Lan, Tsung-Chuan Chen, and Silvio Savarese. 2014. A hierarchical representation for future action prediction. In *European Conference on Computer Vision*. Springer, 689–704.
- [18] Ziwei Liu, Raymond Yeh, Xiaou Tang, Yiming Liu, and Aseem Agarwala. 2017. Video Frame Synthesis using Deep Voxel Flow. *arXiv preprint arXiv:1702.02463* (2017).
- [19] Dhruv Mahajan, Fu-Chung Huang, Wojciech Matusik, Ravi Ramamoorthi, and Peter Belhumeur. 2009. Moving gradients: a path-based method for plausible image interpolation. *ACM Transactions on Graphics (TOG)* 28, 3 (2009), 42.
- [20] Michael Mathieu, Camille Couprie, and Yann LeCun. 2015. Deep multi-scale video prediction beyond mean square error. *arXiv preprint arXiv:1511.05440* (2015).
- [21] Anish Mittal, Anush Krishna Moorthy, and Alan Conrad Bovik. 2012. No-reference image quality assessment in the spatial domain. *IEEE Transactions on Image Processing* 21, 12 (2012), 4695–4708.
- [22] Silvia L Pintea, Jan C van Gemert, and Arnold WM Smeulders. 2014. Déjà vu. In *European Conference on Computer Vision*. Springer, 172–187.
- [23] Alec Radford, Luke Metz, and Soumith Chintala. 2015. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434* (2015).
- [24] MarcAurelio Ranzato, Arthur Szlam, Joan Bruna, Michael Mathieu, Ronan Collobert, and Sumit Chopra. 2014. Video (language) modeling: a baseline for generative models of natural videos. *arXiv preprint arXiv:1412.6604* (2014).
- [25] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. 2012. UCF101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402* (2012).
- [26] Nitish Srivastava, Elman Mansimov, and Ruslan Salakhutdinov. 2015. Unsupervised Learning of Video Representations using LSTMs. In *ICML*. 843–852.
- [27] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*. 3104–3112.
- [28] Thiow Keng Tan, Choong Seng Boon, and Yoshinori Suzuki. 2006. Intra prediction by template matching. In *Image Processing, 2006 IEEE International Conference on*. IEEE, 1693–1696.
- [29] Lucas Theis, Aäron van den Oord, and Matthias Bethge. 2015. A note on the evaluation of generative models. *arXiv preprint arXiv:1511.01844* (2015).
- [30] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. 2015. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*. 4489–4497.
- [31] Joost van Amersfoort, Anitha Kannan, Marc'Aurelio Ranzato, Arthur Szlam, Du Tran, and Soumith Chintala. 2017. Transformation-based models of video sequences. *arXiv preprint arXiv:1701.08435* (2017).
- [32] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. 2008. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*. ACM, 1096–1103.
- [33] Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. 2016. Generating videos with scene dynamics. In *Advances In Neural Information Processing Systems*. 613–621.
- [34] Jacob Walker, Carl Doersch, Abhinav Gupta, and Martial Hebert. 2016. An uncertain future: Forecasting from static images using variational autoencoders. In *European Conference on Computer Vision*. Springer, 835–851.
- [35] Jacob Walker, Abhinav Gupta, and Martial Hebert. 2014. Patch to the future: Unsupervised visual prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3302–3309.
- [36] John YA Wang and Edward H Adelson. 1993. Layered representation for motion analysis. In *Computer Vision and Pattern Recognition, 1993. Proceedings CVPR'93, 1993 IEEE Computer Society Conference on*. IEEE, 361–366.
- [37] Zhou Wang and Alan C Bovik. 2009. Mean squared error: Love it or leave it? A new look at signal fidelity measures. *IEEE signal processing magazine* 26, 1 (2009), 98–117.
- [38] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. 2004. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing* 13, 4 (2004), 600–612.
- [39] Junyuan Xie, Ross Girshick, and Ali Farhadi. 2016. Deep3d: Fully automatic 2D-to-3D video conversion with deep convolutional neural networks. In *European Conference on Computer Vision*. Springer, 842–857.
- [40] Jianwen Xie, Song-Chun Zhu, and Ying Nian Wu. 2016. Synthesizing Dynamic Textures and Sounds by Spatial-Temporal Generative ConvNet. *arXiv preprint arXiv:1606.00972* (2016).
- [41] Tianfan Xue, Jiajun Wu, Katherine Bouman, and Bill Freeman. 2016. Visual dynamics: Probabilistic future frame synthesis via cross convolutional networks. In *Advances in Neural Information Processing Systems*. 91–99.
- [42] Xincheng Yan, Jimei Yang, Kihyuk Sohn, and Honglak Lee. 2016. Attribute2image: Conditional image generation from visual attributes. In *European Conference on Computer Vision*. Springer, 776–791.
- [43] Matthew D Zeiler, Dilip Krishnan, Graham W Taylor, and Rob Fergus. 2010. Deconvolutional networks. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, 2528–2535.
- [44] Tinghui Zhou, Shubham Tulsiani, Weilun Sun, Jitendra Malik, and Alexei A Efros. 2016. View synthesis by appearance flow. In *European Conference on Computer Vision*. Springer, 286–301.