# Learning Energy-based Spatial-Temporal Generative ConvNets for Dynamic Patterns

Jianwen Xie, Song-Chun Zhu, *Fellow, IEEE,* and Ying Nian Wu

**Abstract**—Video sequences contain rich dynamic patterns, such as dynamic texture patterns that exhibit stationarity in the temporal domain, and action patterns that are non-stationary in either spatial or temporal domain. We show that an energy-based spatial-temporal generative ConvNet can be used to model and synthesize dynamic patterns. The model defines a probability distribution on the video sequence, and the log probability is defined by a spatial-temporal ConvNet that consists of multiple layers of spatial-temporal filters to capture spatial-temporal patterns of different scales. The model can be learned from the training video sequences by an "analysis by synthesis" learning algorithm that iterates the following two steps. Step 1 synthesizes video sequences from the currently learned model. Step 2 then updates the model parameters based on the difference between the synthesized video sequences and the observed training sequences. We show that the learning algorithm can synthesize realistic dynamic patterns. We also show that it is possible to learn the model from incomplete training sequences with either occluded pixels or missing frames, so that model learning and pattern completion can be accomplished simultaneously.

**Index Terms**—Deep generative models; Energy-based models; Dynamic textures; Generative ConvNets; Spatial-temporal ConvNets.

## 1 Introduction

### 1.1 Background and motivation

There are a wide variety of dynamic patterns in video sequences, including dynamic textures [1] or textured motions [2] that exhibit statistical stationarity or stochastic repetitiveness in the temporal dimension, and action patterns that are non-stationary in either spatial or temporal domain. Recently we have witnessed tremendous advance in developing discriminative models for dynamic pattern recognition, e.g., [3], [4], [5], [6], [7], [8], [9], and [10], however, the progress in developing generative models of dynamic patterns for synthesis purpose has been lagging behind. Synthesizing dynamic patterns has been an interesting but challenging problem in computer vision and computer graphics. In this paper, we focus on the task of learning to synthesize dynamic patterns via generative modeling of dynamic patterns with a generative version of the convolutional neural network (ConvNet or CNN), or more specificcally, an energy-based model with ConvNet parametrization of the energy function.

The ConvNet [11], [12] has proven to be an immensely successful discriminative learning machine. The convolution operation in the ConvNet is particularly suited for signals such as images, videos and sounds that exhibit translation invariance either in the spatial domain or the temporal domain or both. Recently, researchers have become increasingly interested in the generative aspects of ConvNet, for the purpose of visualizing the knowledge learned by the ConvNet, or synthesizing realistic signals, or developing generative models that can be used for unsupervised learning.

In terms of synthesis, various approaches based on the ConvNet have been proposed for synthesizing realistic static images [13], [14], [15], [16], [17]. However, there has not been much work in the literature on modeling and synthesizing dynamic patterns based on the ConvNet, and this is the focus of the present paper.

In the pattern theory [18], [19] of Grenander, a visual pattern is represented by a probability distribution. Grenander used Gibbs distributions or energy-based models to approximate probability densities for image patterns. In this paper, we continue this paradigm and adopt spatial-temporal convolutional neural networks (ConvNet or CNN) to parametrize the energy functions of the energy-based models that are capable of synthesizing realistic videos of many dynamic patterns.

In terms of generative modeling, generative adversarial networks (GAN) [20] and variational autoencoder (VAE) [21] have emerged as two most popular approaches for unsupervised learning of complex distributions. However, neither GAN nor VAE provides explicit probability densities of the data that they model, since they only focus on generating data by learning a mapping from an easily sampled low dimensional distribution (e.g., Gaussian distribution) to the target data distribution. Moreover, both GAN and VAE rely on auxiliary models for training. For example, GAN adopts a discriminator to train the generator in a minimax two-player game, but eventually discards the discriminator after the generator is trained. VAE recruits an encoder as the inference model to approximate the inference process based on the posterior distribution, which may cause a gap between the VAE and the maximum likelihood estimator. This paper proposes a different model for dynamic patterns. It can be learned without recruiting an auxiliary model.

### 1.2 Overview of model and algorithm

We propose to model dynamic patterns by generalizing the energy-based generative ConvNet model recently proposed by [22]. The energy-based generative ConvNet can be derived from the discriminative ConvNet. It is a random field model, a Gibbs distribution, or an energy-based model [23], [24] that is in the

- *J. Xie is with Hikvision Research Institute, Santa Clara, USA. E-mail: jianwen@ucla.edu.*
- *S.-C. Zhu is with the Department of Statistics, University of California, Los Angeles, USA. E-mail: sczhu@stat.ucla.edu.*
- *Y. N. Wu is with the Department of Statistics, University of California, Los Angeles, USA. E-mail: ywu@stat.ucla.edu.*

*Manuscript received Aug, 2017; revised Aug, 2018; accepted Aug, 2018.*

form of exponential tilting of a reference distribution such as the Gaussian white noise distribution or the uniform distribution. The exponential tilting is parametrized by a ConvNet that involves multiple layers of linear filters and rectified linear units (ReLU) [12], which seek to capture features or patterns at different scales. The log probability density or the energy function of the model is the sum of a ConvNet term that is piecewise linear due to the ReLU non-linearity [25] and the $\ell_2$ norm of the signal that comes from the Gaussian white noise reference distribution. As a result, the energy function is piecewise quadratic, and the energy-based generative ConvNet model is piecewise Gaussian. Moreover, the local modes of the distribution are auto-encoding, where the auto-encoding process involves a bottom-up pass that computes the filter responses followed by a top-down pass that reconstructs the signal where the multiple layers of filters in the bottom-up pass serve as the basis functions in the top-down pass. Such an explicit representation is unique among energy-based models [23], [24], and is a result of the fusion between the ReLU piecewise linear structure and the $\ell_2$ norm term from the Gaussian white noise reference distribution.

The energy-based generative ConvNet can be sampled by the Langevin dynamics. Because of the aforementioned auto-encoding structure of the energy-based generative ConvNet, the Langevin dynamics is driven by the reconstruction error, i.e., the difference between the current sample and its reconstruction by the above auto-encoding process. The model can be learned by the stochastic gradient algorithm [26]. It is an “analysis by synthesis” scheme that seeks to match the synthesized signals generated by the Langevin dynamics to the observed training signals. Specifically, the learning algorithm iterates the following two steps after initializing the parameters and the synthesized signals. Step 1 updates the synthesized signals by the Langevin dynamics that samples from the currently learned model. Step 2 then updates the parameters based on the difference between the synthesized data and the observed data in order to shift the density of the model from the synthesized data towards the observed data. It is shown by [22] that the learning algorithm can synthesize realistic spatial image patterns such as textures and objects.

In this article, we generalize the energy-based spatial generative ConvNet by adding the temporal dimension, so that the resulting ConvNet consists of multiple layers of spatial-temporal filters that seek to capture spatial-temporal patterns at various scales. For dynamic textures, these spatial-temporal filters are convolutional in the temporal domain, reflecting the statistical stationarity or stochastic repetitiveness of the patterns in the temporal domain. If the dynamic textures also exhibit spatial stationarity, we also make the spatial-temporal filters convolutional in the spatial domain. For action and motion patterns that are non-stationary in the temporal domain, the top layer filters are fully connected in the temporal domain. We provide a mode seeking and mode shifting interpretation and an adversarial interpretation of the learning and sampling algorithm. We show that the learning algorithm for training the model can synthesize realistic dynamic patterns. We also show that it is possible to learn the model from incomplete video sequences with either occluded pixels or missing frames, so that model learning and pattern completion can be accomplished simultaneously.

### 1.3 Related work

In this section, we provide a comprehensive review of related work.

*Dynamic pattern recognition*. Recognizing dynamic patterns (e.g., actions) with ConvNets has been extensively studied in the past few years. [4] applied ConvNets with deep structures on a large-scale video dataset for classification. [5] designed a two-stream ConvNet structure, which incorporates spatial network trained on still image frames and temporal network trained on motion in the form of dense optical flow, for action recognition. [6] trained 3D ConvNets [3] on the realistic and large-scale video datasets to learn both appearance and motion features with 3D convolution operations. [7] studied modeling long-range temporal structure with ConvNets and LSTM [27]. [28] and [9] generalized the residual networks (ResNets) for the spatiotemporal domain for video action recognition and dynamic scene recognition. [10] proposed the Inception 3D (I3D) model by inflating all the 2D convolution filters and pooling kernels used by the Inception V1 architecture [29] into 3D convolutions and pre-training the model on the large-scale Kinetics human action video dataset [30]. Instead of learning discriminative spatial-temporal ConvNets of dynamic patterns, our paper mainly focuses on generative modeling of spatial-temporal ConvNets for dynamic pattern synthesis.

*FRAME models*. The FRAME (Filters, Random fields, And Maximum Entropy) model [31], [32], [33] is a Markov random field model (or a Gibbs distribution, or an energy-based model) that defines a probability distribution on the data space. The probability distribution is the maximum entropy distribution that reproduces the statistical properties of filter responses in the observed data. The original FRAME model is a stationary model developed for modeling texture patterns. A non-stationary version of FRAME model designed for object patterns was proposed in [34], [35]. The filters used in the above two versions of FRAME models are the Gabor filters and the isotropic Difference of Gaussian filters. These are linear filters that capture simple local image features, such as edges and blobs. Inspired by the recent successes of deep convolutional neural networks (CNNs or ConvNets) [11], [12], the deep FRAME model [17] replaces the linear filters by the non-linear filters at a certain convolutional layer of a pre-trained deep ConvNet. Such filters can capture more complex patterns, and the deep FRAME model built on such filters can be more expressive. Our model is a spatial-temporal (3D) generalization of the deep FRAME model with the non-linear filters in the deep ConvNet trained by maximum likelihood from the observed data.

*Energy-based generative ConvNet*. Recently, a deep generative model directly derived from the discriminative ConvNet model was proposed in [22]. The resulting model is an energy-based generative ConvNet model. The maximum likelihood learning of the model involves Markov chain Monte Carlo (MCMC) approximation of the gradient of the data log-likelihood. To address the inefficiency of MCMC sampling, [36] developed a multi-grid sampling and learning method for the energy-based generative ConvNet, and [37] proposed to recruit a top-down generator [38] serving as an approximate sampler of the energy-based generative ConvNet model. [22] did not work on dynamic patterns such as those in the video sequences. [39] is a generalization of [22] for dynamic patterns by adopting spatial-temporal ConvNets [3] to capture spatial and temporal features of the video sequences. Recently, [40] proposed a volumetric version of the energy-based generative ConvNet for modeling 3D shape patterns. This paper is an expanded version of our conference paper in [39].

*Dynamic textures*. Generating dynamic textures or textured motions have been studied by [1], [2], [41], [42]. For instance, [1] proposed a vector auto-regressive model coupled with framewise dimension reduction by single value decomposition. It is a linear model with Gaussian innovations. [2] proposed a dynamic model based on sparse linear representation of frames. See [43] for a recent review of dynamic textures. The spatial-temporal generative ConvNet is a non-linear and non-Gaussian model and is expected to be more flexible in capturing complex spatial-temporal patterns in dynamic textures with multiple layers of non-linear spatial-temporal filters. Recently some researcher have started to study dynamic texture synthesis by matching feature statistics, which are extracted by pre-trained convolutional networks, between synthesized and observed examples, e.g., [44] adopted a pre-trained ConvNet for object recognition, while [45] used two pre-trained ConvNets trained for object recognition and optical flow prediction separately. Even though a ConvNet structure is also included in our model, it serves as the energy function of the energy-based model and is learned from scratch by maximizing the log-likelihood of the observed data, without relying on other pre-trained networks for assistance.

*Generative adversarial networks of videos*. Recently, multiple video generation frameworks using generative adversarial network (GAN) [20] were proposed. For example, one can generalize the existing image-based generative adversarial networks framework to video generation by using a single generator consisting of 3D deconvolutional layers. [46] proposed generative adversarial networks for video with a spatial-temporal convolutional architecture that disentangles the scene's foreground from the background. TGAN [47] exploited a 1D temporal generator and a 2D image generator for video generation. The temporal generator takes a single latent variable as input and outputs a set of latent variables, while the image generator transforms these latent variables provided by the temporal generator into video frames. MoCoGAN [48] proposed the motion and content decomposed generative adversarial networks for video generation. All of the above methods need to recruit a discriminator with appropriate convolutional architecture to evaluate whether the generated videos are from the training data or the video generator. Different from GAN-based methods, our model is a deep 3D convolutional energy-based model with only one single bottom-up 3D ConvNet architecture as the energy function. Our model generates video clips by directly sampling from an explicit distribution by MCMC, such as Langevin dynamics, while the GAN-based methods cannot provide explicit distributions of the videos that they seek to model. Our model has an adversarial interpretation. See section 3.1 for details.

*Recurrent neural network*. For temporal data, a popular model is the recurrent neural network [27], [49]. It is a causal model and it requires a starting frame. In contrast, our model is non-causal, and does not require a starting frame. Compared to the recurrent network, our model is more convenient and direct in capturing temporal patterns at multiple time scales.

Subsection 3.3 presents a detailed comparison between various generative models of dynamic patterns that are based on deep neural networks.

### 1.4 Contributions

The following are the main contributions of this paper. (1) We propose an energy-based spatial-temporal generative ConvNet for modeling video sequences by combining the 3D ConvNets [3] and the energy-based generative ConvNets [22]. (2) We show that the model is capable of synthesizing realistic dynamic patterns. (3) We show that it is possible to learn the model from incomplete data. (4) We present a mode seeking and mode shifting interpretation of the learning and sampling algorithm, and we also present an adversarial interpretation of the algorithm.

## 2 Energy-based Spatial-temporal generative ConvNet

### 2.1 Spatial-temporal filters

To fix notation, let $\mathbf{I}(x, t)$ be an image sequence of a video defined on the square (or rectangular) image domain $\mathcal{D}$ and the time domain $\mathcal{T}$, where $x = (x_1, x_2) \in \mathcal{D}$ indexes the coordinates of pixels, and $t \in \mathcal{T}$ indexes the frames in the video sequence. We can treat $\mathbf{I}(x, t)$ as a three dimensional function defined on $\mathcal{D} \times \mathcal{T}$. For a spatial-temporal filter $F$, we let $F * \mathbf{I}$ denote the filtered image sequence or feature map, and let $[F * \mathbf{I}](x, t)$ denote the filter response or feature at pixel $x$ and time $t$.

The spatial-temporal ConvNet is a composition of multiple layers of linear filtering and ReLU non-linearity, as expressed by the following recursive formula:

$$\begin{aligned}[F_k^{(l)} * \mathbf{I}](x,t) = h\Bigg( & \sum_{i=1}^{N_{l-1}} \sum_{(y,s)\in\mathcal{S}_l} w_{i,y,s}^{(l,k)} \\ & \times [F_i^{(l-1)} * \mathbf{I}](x+y, t+s) + b_{l,k}\Bigg),\end{aligned} \tag{1}$$

where $l \in \{1, 2, ..., L\}$ indexes the layers. $\{F_k^{(l)}, k = 1, ..., N_l\}$ are the filters at layer $l$, and $\{F_i^{(l-1)}, i = 1, ..., N_{l-1}\}$ are the filters at layer $l - 1$. $k$ and $i$ are used to index filters at layers $l$ and $l - 1$ respectively, and $N_l$ and $N_{l-1}$ are the numbers of filters at layers $l$ and $l - 1$ respectively. The filters are locally supported, so the range of $(y, s)$ is within a local support $\mathcal{S}_l$ (such as a $7 \times 7 \times 3$ box of image sequence). The weight parameters $(w_{i,y,s}^{(l,k)}, (y, s) \in \mathcal{S}_l, i = 1, ..., N_{l-1})$ define a linear filter that operates on $(F_i^{(l-1)} * \mathbf{I}, i = 1, ..., N_{l-1})$. The $\{b_{l,k}\}$ are bias parameters. The linear filtering operation is followed by ReLU $h(r) = \max(0, r)$. At the bottom layer, $[F_k^{(0)} * \mathbf{I}](x, t) = \mathbf{I}_k(x, t)$, where $k \in \{\mathrm{R}, \mathrm{G}, \mathrm{B}\}$ indexes the three color channels. Sub-sampling may be implemented so that in $[F_k^{(l)} * \mathbf{I}](x, t)$, $x \in \mathcal{D}_l \subset \mathcal{D}$, and $t \in \mathcal{T}_l \subset \mathcal{T}$. For example, if the sub-sampling size is $n$, we only keep the first pixel and then every $n$-th pixel after the first. The sub-sampling operation can be applied to both spatial and temporal domains.

The spatial-temporal filters at multiple layers are expected to capture the spatial-temporal patterns at multiple scales. It is possible that the top-layer filters are fully connected in the spatial domain as well as the temporal domain (e.g., the feature maps are $1 \times 1$ in the spatial domain) if the dynamic pattern does not exhibit spatial or temporal stationarity.

### 2.2 Energy-based spatial-temporal generative ConvNet

The spatial-temporal generative ConvNet is an energy-based model or a random field model defined on the image sequence $\mathbf{I} = (\mathbf{I}(x, t), x \in \mathcal{D}, t \in \mathcal{T})$. It is in the form of exponential tilting of a reference distribution $q(\mathbf{I})$:

$$p(\mathbf{I}; \theta) = \frac{1}{Z(\theta)} \exp\left[f(\mathbf{I}; \theta)\right] q(\mathbf{I}), \tag{2}$$

where the scoring function $f(\mathbf{I};\theta)$ is

$$f(\mathbf{I};\theta) = \sum_{k=1}^{K} \sum_{x \in \mathcal{D}_L} \sum_{t \in \mathcal{T}_L} [F_k^{(L)} * \mathbf{I}](x,t), \tag{3}$$

where $\theta$ consists of all the weight and bias terms that define the filters ($F_k^{(L)}, k = 1, ..., K = N_L$) at layer $L$, and $q$ is the Gaussian white noise model, i.e.,

$$q(\mathbf{I}) = \frac{1}{(2\pi\sigma^2)^{|\mathcal{D}\times\mathcal{T}|/2}} \exp\left[-\frac{1}{2\sigma^2}||\mathbf{I}||^2\right], \tag{4}$$

where $|\mathcal{D} \times \mathcal{T}|$ counts the number of pixels in the domain $\mathcal{D} \times \mathcal{T}$. Without loss of generality, we shall assume $\sigma^2 = 1$. $Z(\theta) = \int \exp[f(\mathbf{I};\theta)]q(\mathbf{I})d\mathbf{I}$ is the normalizing constant or partition function that is analytically intractable.

The scoring function $f(\mathbf{I};\theta)$ in (3) tilts the Gaussian reference distribution into a non-Gaussian model. In fact, the purpose of $f(\mathbf{I};\theta)$ is to identify the non-Gaussian spatial-temporal features or patterns. In the definition of $f(\mathbf{I};\theta)$ in (3), we sum over the filter responses at the top layer $L$ over all the filters, positions and times. The spatial and temporal pooling reflects the fact that we assume the model is stationary in spatial and temporal domains. If the dynamic texture is non-stationary in the spatial or temporal domain, then the top layer filters $F_k^{(L)}$ are fully connected in the spatial or temporal domain, e.g., $\mathcal{D}_L$ is $1 \times 1$.

A simple but consequential property of the ReLU non-linearity is that $h(r) = \max(0, r) = 1(r > 0)r$, where $1()$ is the indicator function, so that $1(r > 0) = 1$ if $r > 0$ and 0 otherwise. As a result, the scoring function $f(\mathbf{I};\theta)$ is piecewise linear [25], and each linear piece is defined by the multiple layers of binary activation variables $\delta_{k,x,t}^{(l)}(\mathbf{I};\theta) = 1\left([F_k^{(l)} * \mathbf{I}](x,t) > 0\right)$, which tells us whether a local spatial-temporal pattern represented by the $k$-th filter at layer $l$, $F_k^{(l)}$, is detected at position $x$ and time $t$. Let $\delta(\mathbf{I};\theta) = \left(\delta_{k,x,t}^{(l)}(\mathbf{I};\theta), \forall l, k, x, t\right)$ be the activation pattern of $\mathbf{I}$. Then $\delta(\mathbf{I};\theta)$ divides the image space into a large number of pieces according to the value of $\delta(\mathbf{I};\theta)$. On each piece of image space with fixed $\delta(\mathbf{I};\theta)$, the scoring function $f(\mathbf{I};\theta)$ is linear, because with fixed value of the indicator function $1(r > 0)$, the ReLU $h(r)$ reduces to a linear mapping. Specifically, according to [22], we can write

$$f(\mathbf{I};\theta) = a_{\theta,\delta(\mathbf{I};\theta)} + \langle \mathbf{I}, \mathbf{B}_{\theta,\delta(\mathbf{I};\theta)} \rangle, \tag{5}$$

where both $a$ and $\mathbf{B}$ are defined by $\delta(\mathbf{I};\theta)$ and $\theta$. In fact, $\mathbf{B} = \partial f(\mathbf{I};\theta)/\partial \mathbf{I}$, and can be computed by back-propagation, with $h'(r) = 1(r > 0)$. The back-propagation process defines a top-down deconvolution process [50], where the filters at multiple layers become the basis functions at those layers, and the activation variables at different layers in $\delta(\mathbf{I};\theta)$ become the coefficients of the basis functions in the top-down deconvolution.

$p(\mathbf{I};\theta)$ in (2) is an energy-based model [23], [24], whose energy function is a combination of the $\ell_2$ norm $||\mathbf{I}||^2$ that comes from the reference distribution $q(\mathbf{I})$ and the piecewise linear scoring function $f(\mathbf{I};\theta)$, i.e.,

$$\begin{aligned}\mathcal{E}(\mathbf{I};\theta) &= -f(\mathbf{I};\theta) + \frac{1}{2}||\mathbf{I}||^2 \\ &= \frac{1}{2}||\mathbf{I}||^2 - \left(a_{\theta,\delta(\mathbf{I};\theta)} + \langle \mathbf{I}, \mathbf{B}_{\theta,\delta(\mathbf{I};\theta)}\rangle\right) \\ &= \frac{1}{2}||\mathbf{I} - \mathbf{B}_{\theta,\delta(\mathbf{I};\theta)}||^2 + \text{const},\end{aligned} \tag{6}$$

where $\text{const} = -a_{\theta,\delta(\mathbf{I};\theta)} - ||\mathbf{B}_{\theta,\delta(\mathbf{I};\theta)}||^2/2$, which is constant on the piece of image space with fixed $\delta(\mathbf{I};\theta)$.

Since $\mathcal{E}(\mathbf{I};\theta)$ is a piecewise quadratic function, $p(\mathbf{I};\theta)$ is piecewise Gaussian. On the piece of image space $\{\mathbf{I} : \delta(\mathbf{I};\theta) = \delta\}$, where $\delta$ is a fixed value of $\delta(\mathbf{I};\theta)$, $p(\mathbf{I};\theta)$ is $\text{N}(\mathbf{B}_{\theta,\delta}, \mathbf{1})$ truncated to $\{\mathbf{I} : \delta(\mathbf{I};\theta) = \delta\}$, where we use $\mathbf{1}$ to denote the identity matrix. If the mean of this Gaussian piece, $\mathbf{B}_{\theta,\delta}$, is within $\{\mathbf{I} : \delta(\mathbf{I};\theta) = \delta\}$, then $\mathbf{B}_{\theta,\delta}$ is also a local mode, and this local mode $\mathbf{I}$ satisfies a hierarchical auto-encoder, with a bottom-up encoding process $\delta = \delta(\mathbf{I};\theta)$, and a top-down decoding process $\mathbf{I} = \mathbf{B}_{\theta,\delta}$. In general, for an image sequence $\mathbf{I}$, $\mathbf{B}_{\theta,\delta(\mathbf{I};\theta)}$ can be considered a reconstruction of $\mathbf{I}$, and this reconstruction is exact if $\mathbf{I}$ is a local mode of $\mathcal{E}(\mathbf{I};\theta)$.

Our model in (2) directly corresponds to a classifier, specifically a spatial-temporal discriminative ConvNet, in the following sense [22], [51], [52], [53], [54], [55]. Suppose we have $C$ categories of video sequences. Let $p_c(\mathbf{I};\theta)$ be the video sequence distribution of category $c$, for $c = 1, ..., C$, which is defined according to model (2). The ConvNet structures $f_c(\mathbf{I})$ may share common lower layers of parameters. Let $\rho_c$ be the prior probability of category $c$ for $c = 1, ..., C$. The posterior distribution $p(c|\mathbf{I})$ for classifying an example $\mathbf{I}$ to the category $c$ is a softmax multi-class classifier (Spatial-temporal discriminative ConvNet) given by

$$p(c|\mathbf{I};\theta) = \frac{\exp[f_c(\mathbf{I};\theta) + b_c]}{\sum_{c=1}^{C}\exp[f_c(\mathbf{I};\theta) + b_c]}, \tag{7}$$

where the category-specific bias term $b_c = \log\rho_c - \log Z_c(\theta) +$ constant. The correspondence to discriminative ConvNet justifies the energy-based spatial-temporal generative ConvNet model.

## 2.3 Sampling and learning algorithm

One can sample from $p(\mathbf{I};\theta)$ of model (2) by the Langevin dynamics:

$$\begin{aligned}\mathbf{I}_{\tau+1} &= \mathbf{I}_\tau - \frac{\epsilon^2}{2}\frac{\partial}{\partial \mathbf{I}}\mathcal{E}(\mathbf{I};\theta) + \epsilon Z_\tau \\ &= \mathbf{I}_\tau - \frac{\epsilon^2}{2}\left[\mathbf{I}_\tau - \frac{\partial}{\partial \mathbf{I}}f(\mathbf{I};\theta)\right] + \epsilon Z_\tau \\ &= \mathbf{I}_\tau - \frac{\epsilon^2}{2}\left[\mathbf{I}_\tau - \mathbf{B}_{\theta,\delta(\mathbf{I}_\tau;\theta)}\right] + \epsilon Z_\tau,\end{aligned} \tag{8}$$

where $\tau$ indexes the time steps, $\epsilon$ is the step size, and $Z_\tau \sim \text{N}(0, \mathbf{1})$ is the Gaussian white noise term.

The Langevin dynamics consists of a deterministic part and a stochastic part. The former is a gradient descent that attempts to find the minimum of the energy function $\mathcal{E}(\mathbf{I};\theta)$, while the latter is a Brownian motion $Z_r$ that prevents the gradient descent from falling into local minimum traps.

The dynamics is driven by the reconstruction error $\mathbf{I} - \mathbf{B}_{\theta,\delta(\mathbf{I};\theta)}$. The finiteness of the step size $\epsilon$ can be corrected by a Metropolis-Hastings acceptance-rejection step. The Langevin dynamics can be extended to Hamiltonian Monte Carlo [56] or more sophisticated versions [57].

The learning of $\theta$ from training image sequences $\{\mathbf{I}_m, m = 1, ..., M\}$ can be accomplished by the maximum likelihood estimation (MLE), which follows an "analysis by synthesis" scheme. Let $L(\theta) = \sum_{m=1}^{M}\log p(\mathbf{I}_m;\theta)/M$, with $p(\mathbf{I};\theta)$ defined in (2),

$$\frac{\partial L(\theta)}{\partial\theta} = \frac{1}{M}\sum_{m=1}^{M}\frac{\partial}{\partial\theta}f(\mathbf{I}_m;\theta) - \text{E}_\theta\left[\frac{\partial}{\partial\theta}f(\mathbf{I};\theta)\right], \tag{9}$$

where the $\text{E}_\theta$ denotes the expetation with respect to $p(\mathbf{I};\theta)$, and the expectation term is due to $\frac{\partial}{\partial\theta}\log Z(\theta) = \text{E}_\theta[\frac{\partial}{\partial\theta}f(\mathbf{I};\theta)]$, which is analytically intractable and has to be approximated by the Monte Carlo samples [26] produced by the Langevin dynamics. Suppose we run $\tilde{M}$ parallel chains of Langevin dynamics according to (8) to synthesize $\tilde{M}$ Monte Carlo samples $\{\tilde{\mathbf{I}}_m, m = 1, ..., \tilde{M}\}$, then the Monte Carlo approximation of the gradient of the log-likelihood function in (9) is given by

$$\frac{\partial L(\theta)}{\partial \theta} \approx \frac{1}{M} \sum_{m=1}^{M} \frac{\partial}{\partial \theta} f(\mathbf{I}_m; \theta) - \frac{1}{\tilde{M}} \sum_{m=1}^{\tilde{M}} \frac{\partial}{\partial \theta} f(\tilde{\mathbf{I}}_m; \theta). \quad (10)$$

$\theta$ can be updated by a stochastic gradient ascent algorithm [26] with learning rate $\eta$:

$$\theta^{(t+1)} = \theta^{(t)} + \eta_t \left[ \frac{1}{M} \sum_{m=1}^{M} \frac{\partial}{\partial \theta} f(\mathbf{I}_m; \theta^{(t)}) - \frac{1}{\tilde{M}} \sum_{m=1}^{\tilde{M}} \frac{\partial}{\partial \theta} f(\tilde{\mathbf{I}}_m; \theta^{(t)}) \right] \quad (11)$$

See Algorithm 1 for a description of the learning and sampling algorithm. The algorithm keeps synthesizing image sequences from the current model, and updating the model parameters in order to match the synthesized image sequences to the observed image sequences. This is an "analysis by synthesis" scheme. The convergence of the algorithm has been studied by [26], [58].

**Algorithm 1** Learning and sampling algorithm

```
Input:
    (1) training image sequences $\{\mathbf{I}_m, m = 1, ..., M\}$
    (2) number of synthesized image sequences $\tilde{M}$
    (3) number of Langevin steps $l$
    (4) number of learning iterations $T$
Output:
    (1) estimated model parameters $\theta$
    (2) synthesized image sequences $\{\tilde{\mathbf{I}}_m, m = 1, ..., \tilde{M}\}$

Let $t \leftarrow 0$, initialize $\theta^{(0)}$.
Initialize $\tilde{\mathbf{I}}_m$, for $m = 1, ..., \tilde{M}$, by sampling from $q(\mathbf{I})$.
repeat
    For each $m$, run $l$ steps of Langevin dynamics to update $\tilde{\mathbf{I}}_m$,
    i.e., starting from the current $\tilde{\mathbf{I}}_m$, each step follows equation (8).
    Calculate $H^{\text{obs}} = \sum_{m=1}^{M} \frac{\partial}{\partial \theta} f(\mathbf{I}_m; \theta^{(t)})/M$, and $H^{\text{syn}} = \sum_{m=1}^{\tilde{M}} \frac{\partial}{\partial \theta} f(\tilde{\mathbf{I}}_m; \theta^{(t)})/\tilde{M}$.
    Update $\theta^{(t+1)} \leftarrow \theta^{(t)} + \eta_t(H^{\text{obs}} - H^{\text{syn}})$, with step size $\eta_t$.
    Let $t \leftarrow t + 1$
until $t = T$
```

In algorithm 1, the Langevin sampling step involves the computation of $\partial f(\mathbf{I}; \theta)/\partial \mathbf{I}$, and the parameter updating step involves the computation of $\partial f(\mathbf{I}; \theta)/\partial \theta$. Because of the ConvNet structure of $f(\mathbf{I}; \theta)$, both gradients can be computed efficiently by back-propagation, and the two gradients share most of their chain rule computations in back-propagation. The whole learning and sampling algorithm can be considered an alternating back-propagation algorithm: (1) Sampling back-propagation: changing the synthesized examples via Langevin dynamics or gradient descent. (2) Learning back-propagation: changing the model parameters according to the synthesized examples and the observed examples by gradient ascent. In term of MCMC sampling, the Langevin dynamics samples from an evolving distribution because $\theta^{(t)}$ keeps changing. Thus the learning and sampling algorithm runs non-stationary chains.

## 3 Interpretations and Related Models

This section presents interpretations of the learning algorithms, as well as related models of dynamic patterns.

### 3.1 Adversarial interpretation

Our spatial-temporal generative ConvNet model is an energy-based model

$$p(\mathbf{I}; \theta) = \frac{1}{Z(\theta)} \exp[-\mathcal{E}(\mathbf{I}; \theta)], \quad (12)$$

where the energy function $\mathcal{E}(\mathbf{I}; \theta) = -f(\mathbf{I}; \theta) + \frac{1}{2}||\mathbf{I}||^2$.

The update of $\theta$ is based on $\partial L(\theta)/\partial \theta$, which can be approximated by

$$\begin{aligned} \frac{\partial L(\theta)}{\partial \theta} &\approx \frac{1}{\tilde{M}} \sum_{m=1}^{\tilde{M}} \frac{\partial}{\partial \theta} \mathcal{E}(\tilde{\mathbf{I}}_m; \theta) - \frac{1}{M} \sum_{m=1}^{M} \frac{\partial}{\partial \theta} \mathcal{E}(\mathbf{I}_m; \theta) \\ &= \frac{\partial}{\partial \theta} \left[ \frac{1}{\tilde{M}} \sum_{m=1}^{\tilde{M}} \mathcal{E}(\tilde{\mathbf{I}}_m; \theta) - \frac{1}{M} \sum_{m=1}^{M} \mathcal{E}(\mathbf{I}_m; \theta) \right], \end{aligned} \quad (13)$$

where $\{\tilde{\mathbf{I}}_m, m = 1, ..., \tilde{M}\}$ are the synthesized image sequences that are generated by the Langevin dynamics. At the zero temperature limit, the Langevin dynamics becomes gradient descent:

$$\tilde{\mathbf{I}}_{\tau+1} = \tilde{\mathbf{I}}_\tau - \frac{\epsilon^2}{2} \frac{\partial}{\partial \tilde{\mathbf{I}}} \mathcal{E}(\tilde{\mathbf{I}}_\tau; \theta). \quad (14)$$

The algorithm has an adversarial interpretation where the learning and sampling steps play a minimax game. Consider the value function $V(\tilde{\mathbf{I}}_m, m = 1, ..., \tilde{M}; \theta)$:

$$\frac{1}{\tilde{M}} \sum_{m=1}^{\tilde{M}} \mathcal{E}(\tilde{\mathbf{I}}_m; \theta) - \frac{1}{M} \sum_{m=1}^{M} \mathcal{E}(\mathbf{I}_m; \theta). \quad (15)$$

The updating of $\theta$ is to increase $V$ by shifting the low energy regions from the synthesized image sequences $\{\tilde{\mathbf{I}}_m\}$ to the observed image sequences $\{\mathbf{I}_m\}$, whereas the updating of $\{\tilde{\mathbf{I}}_m, m = 1, ..., \tilde{M}\}$ is to decrease $V$ by moving the synthesized image sequences towards the low energy regions (i.e., decreasing $\frac{1}{\tilde{M}} \sum_{m=1}^{\tilde{M}} \mathcal{E}(\tilde{\mathbf{I}}_m; \theta)$). Therefore, the resulting algorithm approximately solves the minimax problem below

$$\max_{\theta} \min_{\{\tilde{\mathbf{I}}_m\}} V(\{\tilde{\mathbf{I}}_m\}; \theta). \quad (16)$$

This is an adversarial interpretation of the learning and sampling algorithm, but the value function is different from that of the generative adversarial learning [20]. It can also be considered a generalization of the herding method [59] from exponential family models to general energy-based models.

In our work, we let $-\mathcal{E}(\mathbf{I}; \theta) = f(\mathbf{I}; \theta) - ||\mathbf{I}||^2/2\sigma^2$. We can also let $-\mathcal{E}(\mathbf{I}; \theta) = f(\mathbf{I}; \theta)$ by assuming a uniform reference distribution $q(\mathbf{I})$. Our experiments show that the model with the uniform $q$ can also synthesize realistic dynamic patterns.

### 3.2 Mode seeking and mode shifting

The data distribution $P_{\text{data}}$ might have many local modes, if the observed data are highly varied. Our model parametrized by a ConvNet structure $f(\mathbf{I};\theta)$ can be flexible and expressive enough to fit such a $P_{\text{data}}$ by creating modes to encode the observed examples with highly diverse patterns. The $f(\mathbf{I};\theta)$ or equivalently the energy function $\mathcal{E}(\mathbf{I};\theta)$ should be learned such that the energy function places lower values on the observed examples than the unobserved examples. This can be achieved by the sampling and learning algorithm presented in Algorithm 1, which can be interpreted as a process that alternates mode seeking and mode shifting.

The sampling step can be interpreted as mode seeking where the Langevin dynamics searches for low energy (high probability) modes in the landscape defined by $\mathcal{E}(\mathbf{I};\theta)$ via stochastic gradient descent and settles the synthesized image sequences $\{\tilde{\mathbf{I}}_m\}$ around the low energy (high density) regions of $\mathcal{E}(\mathbf{I};\theta)$.

The learning step can be interpreted as mode shifting by shifting the low energy (high density) regions from the synthesized image sequences $\{\tilde{\mathbf{I}}_m\}$ towards the observed image sequences $\{\mathbf{I}_m\}$, or shifting the major modes (or basins) of the energy function $\mathcal{E}(\mathbf{I};\theta)$ from the synthesized image sequences towards the observed image sequences, until the observed image sequences stay in the major modes of the energy landscape.

The learning algorithm will create and sharpen the major modes to concentrate on the observed image sequences, if those modes are too diffused around the observed image sequences. The learned energy landscape might have some major modes that are not occupied by the observed examples, and these modes will create examples that are similar to the observed examples.

This mode seeking and mode shifting interpretation is related to Hopfield network [60] that uses local modes of energy landscape to memorize the observed examples. This is also related to attractor network [61] with the Langevin dynamics serving as the attractor dynamics.

### 3.3 Comparison with other models of dynamic patterns

The energy-based model is based on a bottom-up ConvNet $f_\theta(\mathbf{I})$ defined on the video sequence $\mathbf{I}$. There are other models based on top-down ConvNets.

One generator model [62] assumes a hidden noise vector $z \sim \mathrm{N}(0, I_d)$, where $I_d$ is the identity matrix of dimension $d$. Then $\mathbf{I} = g(z) + \epsilon$, where $g$ is parametrized by a top-down spatial-temporal ConvNet, and $\epsilon$ is Gaussian white noise.

The other generator model is a non-linear version of the state space model or hidden Markov model, where $s_t = r(s_{t-1}, u_t)$, and $\mathbf{I}_t = g(s_t) + \epsilon_t$, where $s_t$ is the state vector, and $u_t$ is the noise vector, and the transition model $r$ is parametrized by a forward network, and the emission model $g$ is parametrized by a top-down network. This model is called dynamic generator model [63]. It has a causal structure. It is a non-linear generalization of the dynamic texture model [1].

The above generator models can be learned by maximum likelihood. They can also be learned jointly with inference models as in VAE, or discriminator models as in GAN, or the energy-based models. In the joint training of the dynamic generator model and the energy-based model, the former plays the role of actor and the latter plays the role of evaluator or critic.

In comparison with classical mechanics, the energy-based model is similar to the Lagrangian formulation which is based on the action defined on the whole trajectory, whereas the dynamic generator model is similar to the Hamiltonian formulation which unfolds the trajectory over time.

In the Gibbs distribution in statistical mechanics, the energy function is defined on the spatial domain, and the system evolves over time and converges to the Gibbs distribution [64]. In the energy-based model studied in this paper, the energy function is defined on the spatial-temporal domain, where the time dimension is treated in the same way as the spatial dimensions. The Langevin dynamics evolves this spatial-temporal model over a virtue time (denoted by $\tau$). This treatment is similar to the lattice field theory in physics, which is also based on a Gibbs distribution whose energy is defined on the spatial-temporal domain, and the MCMC sampler evolves the system over a virtue time [64].

In the context of image-based control or intuitive physics, we can jointly model the actions and the video sequences, using the energy-based model and the dynamic generator model. The energy function in the energy-based model can be interpreted as the cost function for optimal control, and the dynamic generator model consists of the policy and the dynamics [65].

Beside the energy-based models and the generator models, there are also flow-based models [66], where the log-likelihoods can be computed in closed form. Specifically, $\mathbf{I} = g(z)$, where $z$ is a noise vector that is of the same dimensionality as $\mathbf{I}$, and $g$ is a composition of a sequence of invertible transformations that are designed such that the inverse and the Jacobian of $g$ can be computed in closed form.

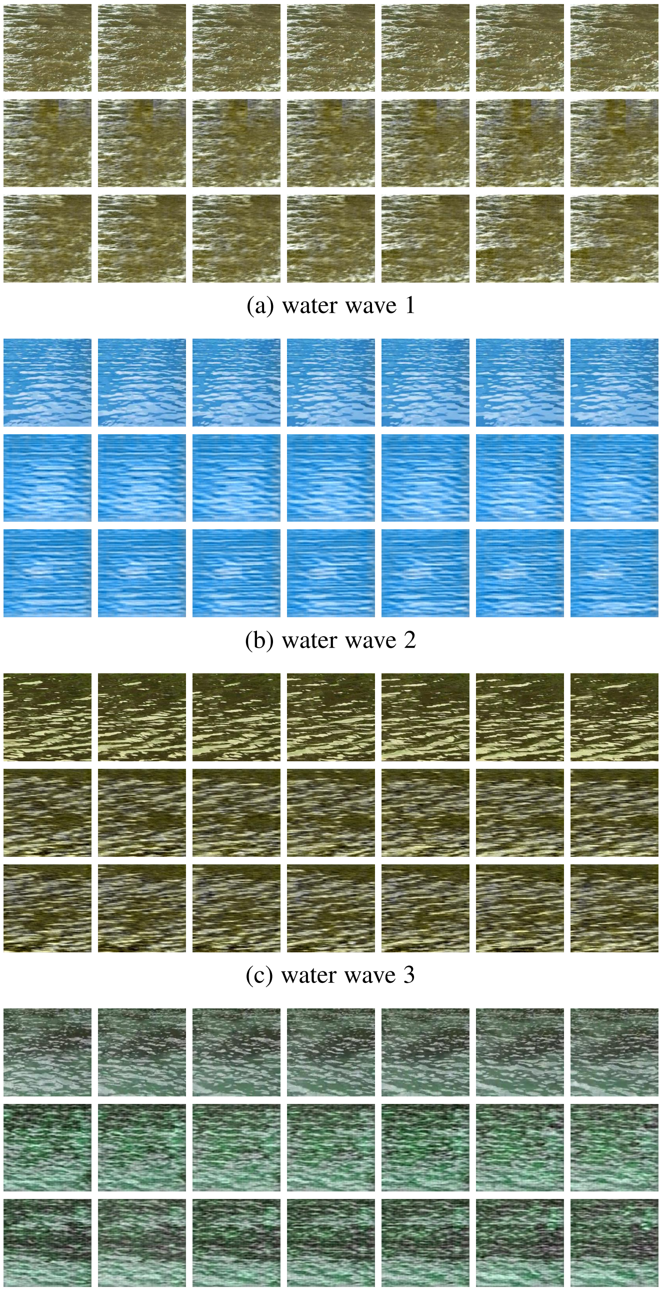
## 4 EXPERIMENTS

We learn the energy-based spatial-temporal generative ConvNet from video clips collected from DynTex++ dataset of [67] and the Internet. We show the synthesis results by displaying the frames in the video sequences. We have posted the synthesis results on the project page http://www.stat.ucla.edu/~jxie/STGConvNet/STGConvNet.html, so that the reader can watch the videos.

### 4.1 Experiment 1: Generating dynamic textures with both spatial and temporal stationarity

We first learn the model from dynamic textures that are stationary in both spatial and temporal domains. Because these dynamic textures exhibit spatial-temporal homogeneity, we use spatial-temporal filters that are convolutional in both spatial and temporal domains to design our network $f(\mathbf{I};\theta)$, which consists of three layers of spatial-temporal convolution, ReLU nonlinearity, and sub-sampling. The first layer has 120 $15 \times 15 \times 15$ filters with sub-sampling size of 7 pixels and frames. The second layer has 40 $7 \times 7 \times 7$ filters with sub-sampling size of 3. The third layer has 20 $3 \times 3 \times 2$ filters with sub-sampling size of $2 \times 2 \times 1$. The sub-sampling operation is useful to reduce the size of the output feature map of each layer for the sake of efficient computation.

Figure 1 displays 4 results of learning to synthesize water waves. For each category, the first row displays 7 frames of the observed sequence, while the second and third rows show the corresponding frames of two synthesized sequences generated by the learning algorithm. The qualitative results displayed in Figure 1 clearly show that our model can learn to generate realistic examples of dynamic textures with both spatial and temporal stationarity, such as water waves. The synthesized examples are similar, but not identical, to the observed video sequence. Generally, when the number of chains of synthesis increases, the diversity of the synthesized sequences will also increase.



(a) water wave 1

(b) water wave 2

(c) water wave 3

(d) water wave 4

Fig. 1. Synthesizing dynamic textures with both spatial and temporal stationarity. For each category, the first row displays the frames of the observed sequence, and the second and third rows display the corresponding frames of two synthesized sequences generated by the learning algorithm. The observed and the synthesized videos are of the size 224 × 224 pixels × 50 or 70 frames.

We use the layer-by-layer learning scheme. Starting from the first layer, we sequentially add the layers one by one. Each time we learn the model and generate the synthesized image sequence using Algorithm 1. While learning the new layer of filters, we refine the lower layers of filters with back-propagation. Due to the limitation of the GPU memory and for the sake of computational efficiency, we use $\tilde{M} = 3$ chains for Langevin sampling. Even with a single chain, the statistics can still be estimated because of stationarity in the temporal and spatial domains. We learn an energy-based spatial-temporal generative ConvNet for each category from one observed video that is prepared to be of the size 224 × 224 pixels × 50 or 70 frames. The range of pixel intensities is [0, 255]. Mean subtraction is used as pre-processing. The number of Langevin iterations between every two consecutive updates of parameters, $l = 20$. The number of learning iterations $T = 1200$, where we add one more layer every $400$ iterations. We use layer-specific learning rates, where the learning rate at the higher layer is less than that at the lower layer, in order to obtain stable convergence.

### 4.2 Experiment 2: Generating dynamic textures with only temporal stationarity

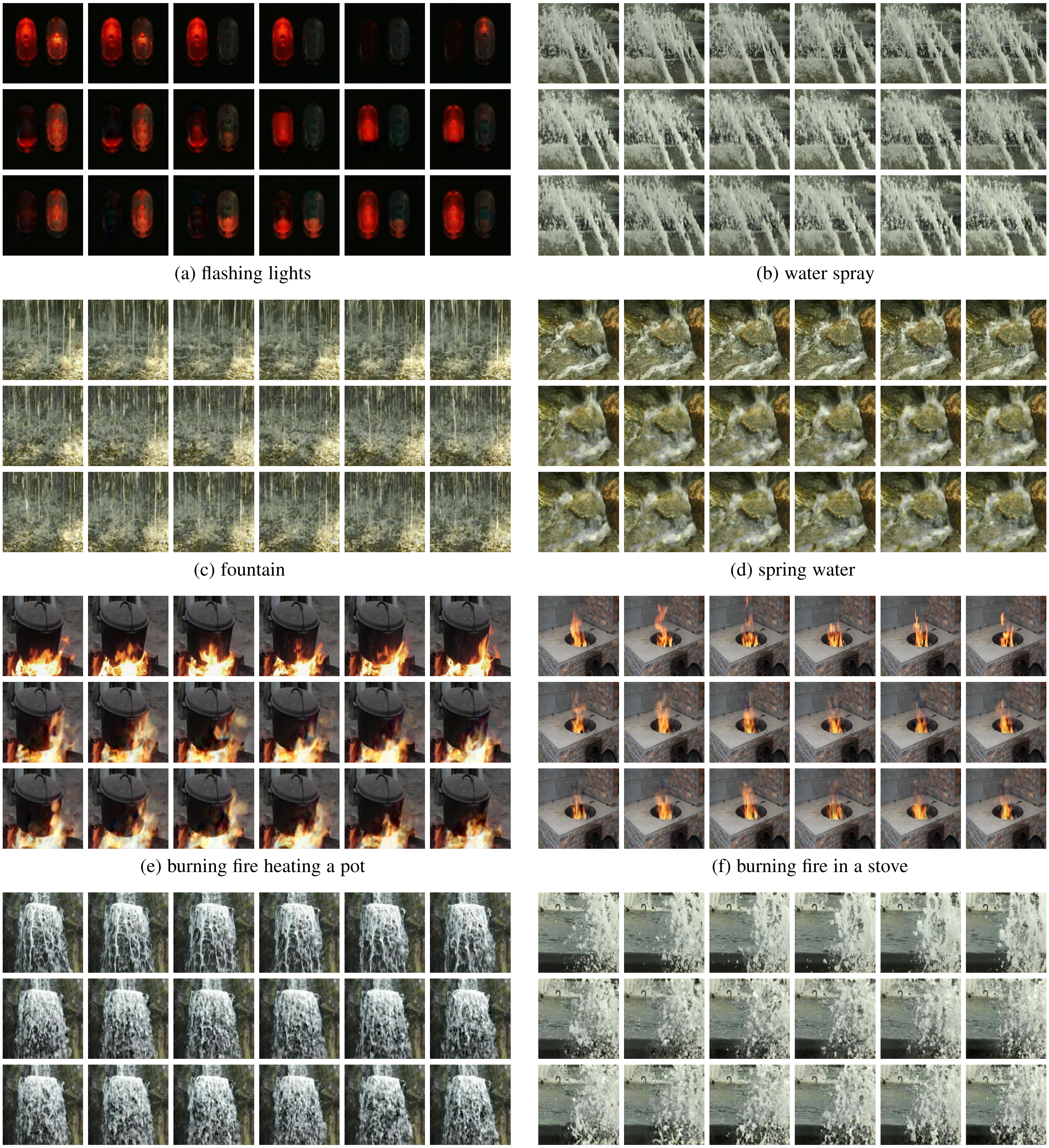
Many dynamic textures have structured background and objects that are not stationary in the spatial domain. In this case, the network used in Experiment 1 may fail. However, we can modify the network in Experiment 1 by using filters that are fully connected in the spatial domain at the second layer to account for spatial non-homogeneity of the dynamic textures. Specifically, the first layer has 120 $7 \times 7 \times 7$ filters with sub-sampling size of 3 pixels and frames. The second layer is a spatially fully connected layer, which contains 30 filters that are fully connected in the spatial domain but convolutional in the temporal domain. The temporal size of the filters is 4 frames with sub-sampling size of 2 frames in the temporal dimension. Due to the spatial full connectivity at the second layer, the spatial domain of the feature maps at the third layer is reduced to $1 \times 1$. The third layer has 5 $1 \times 1 \times 2$ filters with sub-sampling size of 1 in the temporal dimension.

We use end-to-end learning scheme to learn the above 3-layer energy-based spatial-temporal generative ConvNet for dynamic textures with only temporal stationarity. At each iteration, the 3 layers of filters are updated with 3 different layer-specific learning rates. The learning rate at the higher layer is much less than that at the lower layer to avoid the issue of large gradients.

We learn an energy-based spatial-temporal generative ConvNet for each category from one training video. We synthesize $\tilde{M} = 3$ videos using the Langevin dynamics. Figure 2 displays the results. For each category, the first row shows 6 frames of the observed sequence (224 × 224 pixels × 70 frames), and the second and third rows show the corresponding frames of two synthesized sequences generated by the learning algorithm. We use the same set of parameters for all the categories without tuning. Figure 3 compares our method to that of [1], which is a linear dynamic system model. The image sequence generated by this model appears more blurred than the sequence generated by our method.

The computational time for each iteration including 20 steps of Langevin dynamics is roughly 120 seconds on a PC with an Intel i7-6700k CPU and a Titan Xp GPU. It takes roughly 5 seconds for each step of Langevin dynamics sampling.

Quantitative evaluation for dynamic texture synthesis is a particularly challenging task, because there is no unique correct output when synthesizing new samples of an observed dynamic texture. The structural similarity (SSIM) index [68] is designed to provide a perceptual judgment on the similarity between two videos, and ranges from -1 to 1, with a larger score indicating greater similarity. We can calculate the SSIM between the synthesized image sequence and the original image sequence to examine the synthesis quality. A larger SSIM indicates a better synthesis quality due to higher perceptual similarity between the synthesized and original sequences. We compare our method with some baseline methods for dynamic textures, such as HOSVD [69], FFT-LDS [70], and Doretto et al [1], in terms of SSIM on 17 dynamic texture videos in Table 1. Our method outperforms the baseline methods.



(a) flashing lights

(b) water spray

(c) fountain

(d) spring water

(e) burning fire heating a pot

(f) burning fire in a stove

(g) waterfall in a mountain

(h) water spray in a fountain

Fig. 2. Synthesizing dynamic textures with only temporal stationarity. For each category, the first row displays 6 frames of the observed sequence, and the second and third rows display the corresponding frames of two synthesized sequences generated by the learning algorithm. The observed and the synthesized videos are of the size 224 $\times$ 224 pixels $\times$ 70 frames. (a) flashing lights. (b) water spray. (c) fountain. (d) spring water. (e) burning fire heating a pot. (f) burning fire in a stove. (g) waterfall in a mountain. (h) water spray in a fountain.

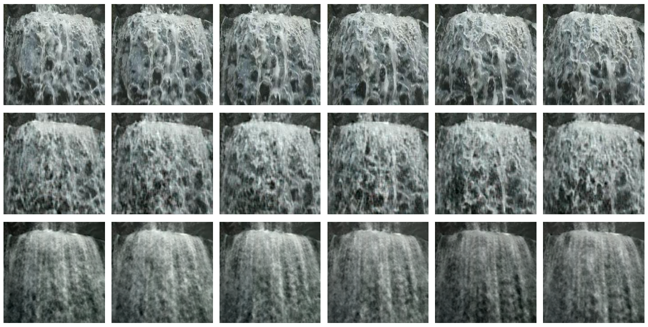


Fig. 3. Comparison on synthesizing dynamic texture of waterfall. From top to bottom: segments of the observed sequence, synthesized sequence by our method, and synthesized sequence by the method of [1].

TABLE 1
A comparison of different dynamic texture models on SSIM

| | ours | HOSVD [69] | FFT-LDS [70] | Doretto [1] |
|---|---|---|---|---|
| boiling water | **0.8890** | 0.4777 | 0.4719 | 0.8628 |
| falling water | **0.4044** | 0.2246 | 0.2421 | 0.3458 |
| fire flames | **0.4893** | 0.2180 | 0.1491 | 0.4086 |
| fire in a stove | **0.9259** | 0.4646 | 0.4895 | 0.9075 |
| fire heating a pot | **0.7969** | 0.4350 | 0.4401 | 0.7853 |
| flashing lights | **0.7960** | 0.2368 | 0.1982 | 0.7561 |
| fountain | **0.3487** | 0.1572 | 0.1114 | 0.3128 |
| fountain spray | **0.5422** | 0.2366 | 0.1948 | 0.4981 |
| rocky waterfall | **0.6245** | 0.1919 | 0.2281 | 0.6183 |
| round fountain | **0.6662** | 0.2082 | 0.2160 | 0.6617 |
| spring water | **0.5703** | 0.1587 | 0.2771 | 0.5561 |
| washing machine | **0.9175** | 0.4520 | 0.6026 | 0.9032 |
| water spray | **0.3528** | 0.1615 | 0.0937 | 0.3365 |
| water spray in a pool | **0.5610** | 0.2171 | 0.1560 | 0.3442 |
| water stream | **0.5030** | 0.1667 | 0.1808 | 0.4492 |
| waterfall | **0.3045** | 0.1473 | 0.1001 | 0.2566 |
| waterfall in a mountain | **0.5415** | 0.2272 | 0.2156 | 0.5206 |
| Avg. | **0.6020** | 0.2577 | 0.2569 | 0.5602 |

Human perception, which is also an important and reliable measure to evaluate synthesis quality, has been used in [71], [45], [72], [63]. Following the protocol of [63], we conduct a human perceptual study to evaluate the perceived realism of the synthesized dynamic textures. Twenty different human observers are randomly chosen to participate in the perceptual experiment. Each participant performs 36 (12 categories × 3 examples) pairwise comparisons between a synthesized dynamic texture and its real version. Participants are asked to select which one looks more realistic after viewing each pair of dynamic textures for a specified exposure time. The dynamic textures are all shown at the same resolution (each image frame is of the size $64 \times 64$ pixels) in the form of video. The comparisons are randomized across the left-right order of two videos in each pair and the order of presenting different video pairs. The exposure time is chosen from discrete durations between 0.3 and 3.6 seconds. This evaluates how quickly the difference between two dynamic textures can be perceived.

We record the "fooling" rate, which is the participant error rate in distinguishing synthesized dynamic textures from real ones, for each participant. The higher the "fooling" rate, the more realistic the synthesized dynamic textures. Note that "perfectly" synthesized results would lead to a "fooling" rate of $50\%$ , indicating that the participants are unable to differentiate between the synthesized and real examples, and thus make random guesses.

Our model is compared with four baseline methods, such as LDS (linear dynamic system) [1], dynamic generator [63], TwoStream [45] and MoCoGAN [48], on 12 dynamic texture video sequences (e.g., waterfall, burning fire, waving flag, etc) that have been adopted in [63]. We briefly introduce them as follows. LDS simulates dynamic textures by a first-order auto-regressive moving average model driven by white zero-mean Gaussian noise; Dynamic generator is a non-linear generalization of the linear state space model where the non-linear transformations in the transition and emission models are parameterized by neural networks, and its training is based on maximum likelihood, which is accomplished by the alternating back-propagation through time algorithm; TwoStream method synthesizes dynamic textures by matching the feature statistics that are extracted from two pre-trained convolutional neural networks between synthesized and observed examples, where the pre-trained convolutional neural networks are trained for two independent tasks: object recognition, and optical flow prediction; and MoCoGAN is a motion and content decomposed generative adversarial network for video generation, where extra networks, e.g., image and video discriminators, are recruited to train the video generator in an adversarial manner.

Figure 4 shows the comparison results at different exposure times, where the "fooling" rate decreases as exposure time increases, and then remains at the same level for longer exposures. This means that as the given exposure time becomes longer and longer, it becomes easier for the participants to tell "fake" examples from real ones. In summary, the dynamic textures generated by our proposed method are more realistic than other baseline methods. Besides, since the dynamic generator is a non-linear generalization of LDS, it is undoubtedly more expressive than LDS. The result also shows that the LDS outperforms those methods using sophisticated deep network structures (i.e., TwoStream and MoCoGAN). This is not surprising because when learning from a small number of training data (in this experiment, we only have one single training example), the MoCoGAN, with a large number of parameters that need to be estimated, may not fit the training data very well by using an unstable and complicated adversarial training scheme, while the TwoStream method has a limitation that it cannot synthesize dynamic textures not being spatially homogeneous (i.e., dynamic textures with structured background, e.g., burning fire heating a pot). Our model is simple yet powerful, relying on neither extra networks nor extra labeled data for pre-training.

The learning of our model can be scaled up. We learn the fire pattern from 30 training videos, with mini-batch implementation. The size of each mini-batch is 10 videos. Each video contains 30 frames ($100 \times 100$ pixels). For each mini-batch, $\tilde{M} = 13$ parallel chains for Langevin sampling is used. The gradient to update the model parameters is averaged over all mini-batches. For this experiment, we slightly modify the network by using 120 $11 \times 11 \times 9$ filters with sub-sampling size of 5 pixels and 4 frames at the first layer, and 30 spatially fully connected filters with temporal size of 5 frames and sub-sampling size of 2 at the second layer, while keeping the setting of the third layer unchanged. The number of learning iterations $T = 1300$. Figure 5 shows the result of learning fire patterns by displaying one frame for each of 30 observed sequences, the corresponding frame for each of the synthesized sequences, and three examples of synthesized sequences. Figure 6 shows another example of learning flowing cloud patterns from 30 training videos.

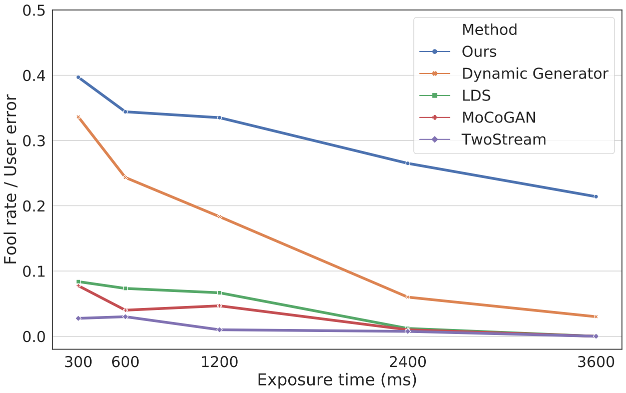
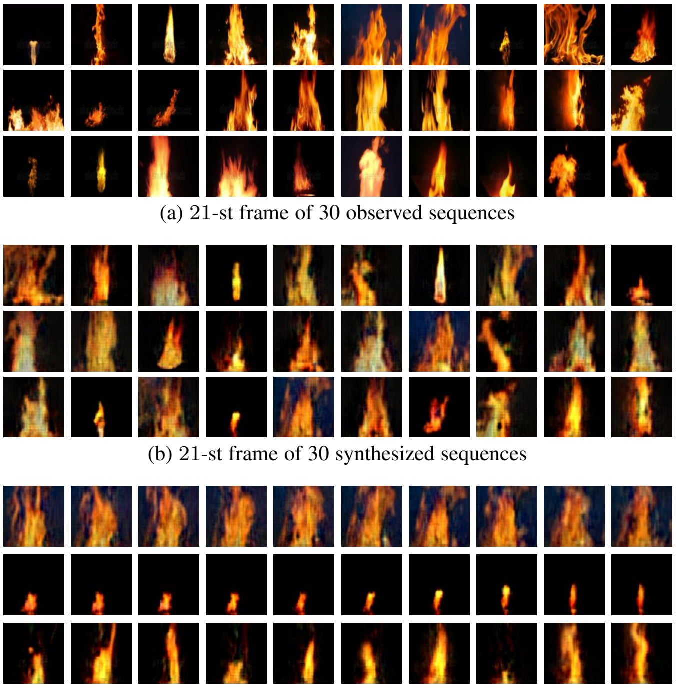


Fig. 4. Limited time pairwise comparison results. Each curve shows the "fooling" rate / user error (realism) over different exposure times. The number of pairwise comparisons is 36. The number of participants is 20.



(a) 21-st frame of 30 observed sequences

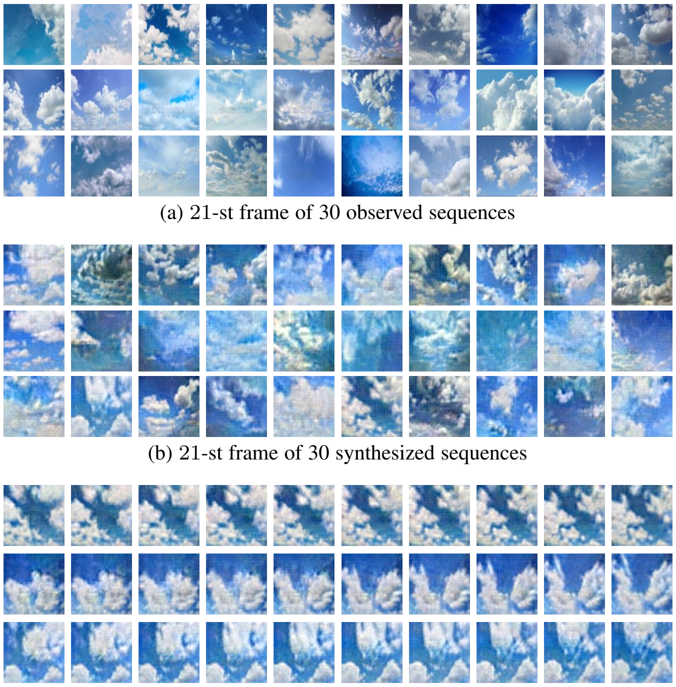
(b) 21-st frame of 30 synthesized sequences

(c) 3 examples of synthesized sequences

Fig. 5. Learning from 30 observed burning fire videos with mini-batch implementation. The batch size is 10. For each mini-batch, the number of parallel chains for synthesis is 13. The observed and synthesized videos are of the size 100 ×100 pixels × 30 frames. (a) displays one frame for each of 30 observed sequences. (b) displays the corresponding frame for each of the synthesized sequences. (c) shows three examples (one example per row) of synthesized video sequences.

### 4.3 Experiment 3: Generating action patterns without spatial or temporal stationarity

Experiments 1 and 2 show that the energy-based generative spatial-temporal ConvNet can learn from video sequences without alignment. We can also specialize it to learning roughly aligned video sequences of action patterns, which are non-stationary in either spatial or temporal domain, by using a single top-layer filter that covers the whole video sequence.



(a) 21-st frame of 30 observed sequences

(b) 21-st frame of 30 synthesized sequences

(c) 3 examples of synthesized sequences

Fig. 6. Learning from 30 observed flowing cloud videos with mini-batch implementation. The batch size is 10. For each mini-batch, the number of parallel chains for synthesis is 13. The observed and synthesized videos are of the size 100 ×100 pixels × 30 frames. (a) displays one frame for each of 30 observed sequences. (b) displays the corresponding frame for each of the synthesized sequences. (c) shows three examples (one example per row) of synthesized video sequences.

The action patterns are spatial-temporally aligned in the sense that (1) for each time step, the target objects in different videos possess the same locations, shapes, and poses (i.e., spatially aligned), and (2) the start and end times of the actions in different videos are the same (i.e., temporally aligned).

We learn a 2-layer energy-based spatial-temporal generative ConvNet from video sequences of roughly aligned actions. The first layer has 200 $7 \times 7 \times 7$ filters with sub-sampling size of 3 pixels and frames. The second layer is a fully connected layer with a single filter that covers the whole sequence. The observed sequences are of the size $100 \times 200$ pixels $\times 70$ frames.

Figure 7 displays three results of modeling and synthesizing actions from roughly aligned video sequences. We learn a model for each category, where the number of training sequences is 5 for the running cow example, 2 for the running tiger example, and 2 for the running llama example. The videos are collected from the Internet and each has 70 frames, with an animal running at the center of a black background. For each example, Figure 7 displays segments of 2 observed sequences, and segments of 2 synthesized action sequences generated by the learning algorithm. We run $\tilde{M} = 8$ paralleled chains for the experiment of running cows, 4 paralleled chains for the experiment of running tigers, and 5 paralleled chains for the experiments of running llamas.

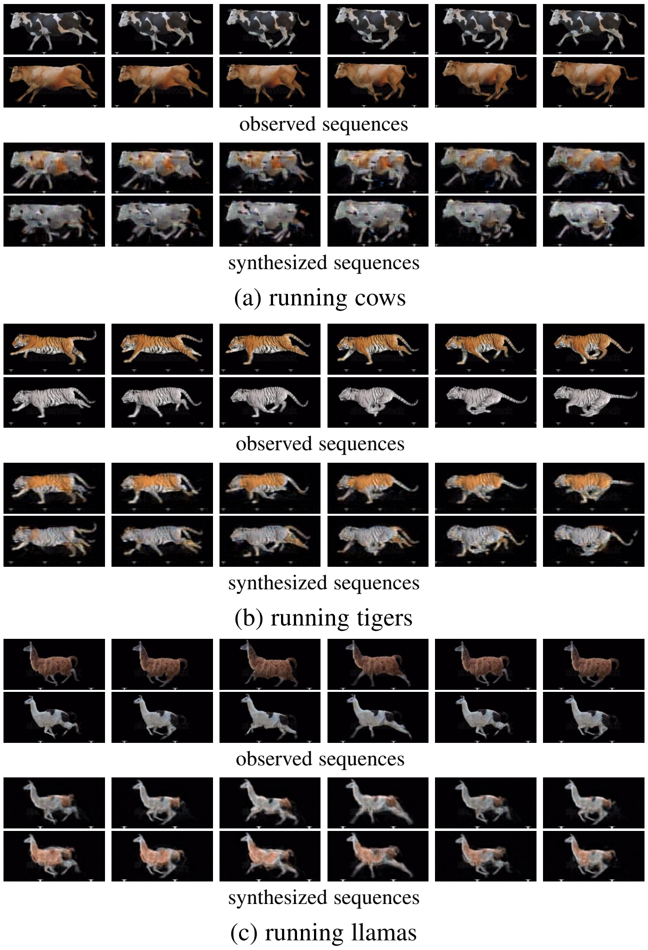
The experiments show that our model can capture non-stationary action patterns. In general, both appearances and motions of the animals in the synthesized video sequences are physically plausible. We also notice that the synthesized action patterns contain some artifacts (e.g., the appearance of additional limbs of the animals in the synthesized video sequences), which is due to the fact that the action patterns in the collected training video sequences are not perfectly aligned (e.g., at a specific time, the poses of the animals in different videos are not the same). Note that synthesizing action patterns can be difficult for the methods of [1] and [45].

One limitation of our model is that it does not involve explicit tracking of the objects and their parts, therefore the model can only learn descriptive features of an action pattern for the sake of synthesis, instead of an explainable action decomposition for the purpose of understanding. The other limitation is that a single model can only be used for single category of action pattern. Unsupervised learning from actions of mixed categories requires fitting a mixture of energy-based spatial-temporal generative ConvNet models. However, this would require the estimation of the intractable normalizing constants $Z(\theta)$ of the models, because the EM-type algorithm to fit mixtures of energy-based spatial-temporal generative ConvNet models involves computation of the log-likelihood of each video sequence under each model. Our previous works [34] and [35] have studied this issue. Nevertheless, the dynamic generator [63] and MoCoGAN [48] adopt top-down generators, with explicit latent variables accounting for action categories, to learn from actions of mixed categories in an unsupervised manner.

### 4.4 Experiment 4: Learning from incomplete data

Our model can learn from video sequences with occluded pixels. The task is inspired by the fact that most of the videos contain occluded objects. Our learning method can be adapted to this task with minimal modification. The modification involves, for each iteration, running $k$ steps of Langevin dynamics to recover the occluded regions of the observed sequences. The additional Langevin dynamics for recovery is initialized by the observed occluded sequences, and it only synthesizes the occluded regions, while keeping the visible parts in the observed sequences unchanged. At each iteration, we use the completed observed sequences and the synthesized sequences to compute the gradient of the log-likelihood and update the model parameters. Our method simultaneously accomplishes the following tasks: (1) recover the occluded pixels of the training video sequences, (2) synthesize new video sequences from the learned model, (3) learn the model by updating the model parameters using the recovered sequences and the synthesized sequences. See Algorithm 2 for the description of the learning, sampling, and recovery algorithm.

We design 3 types of occlusions: (1) Type 1: salt and pepper occlusion, where we randomly place $7 \times 7$ masks on the $150 \times 150$ image domain to cover $50\%$ of the pixels of the videos. (2) Type 2: single region mask occlusion, where we randomly place a $60 \times 60$ mask on the $150 \times 150$ image domain. (3) Type 3: missing frames, where we randomly block $50\%$ of the image frames from each video. Figure 8 displays two examples of the recovery results for each type of occlusion. Each video has 70 frames. We can see that our model can recover the incomplete video sequences, while learning from them.


observed sequences

synthesized sequences

(a) running cows

observed sequences

synthesized sequences

(b) running tigers

observed sequences

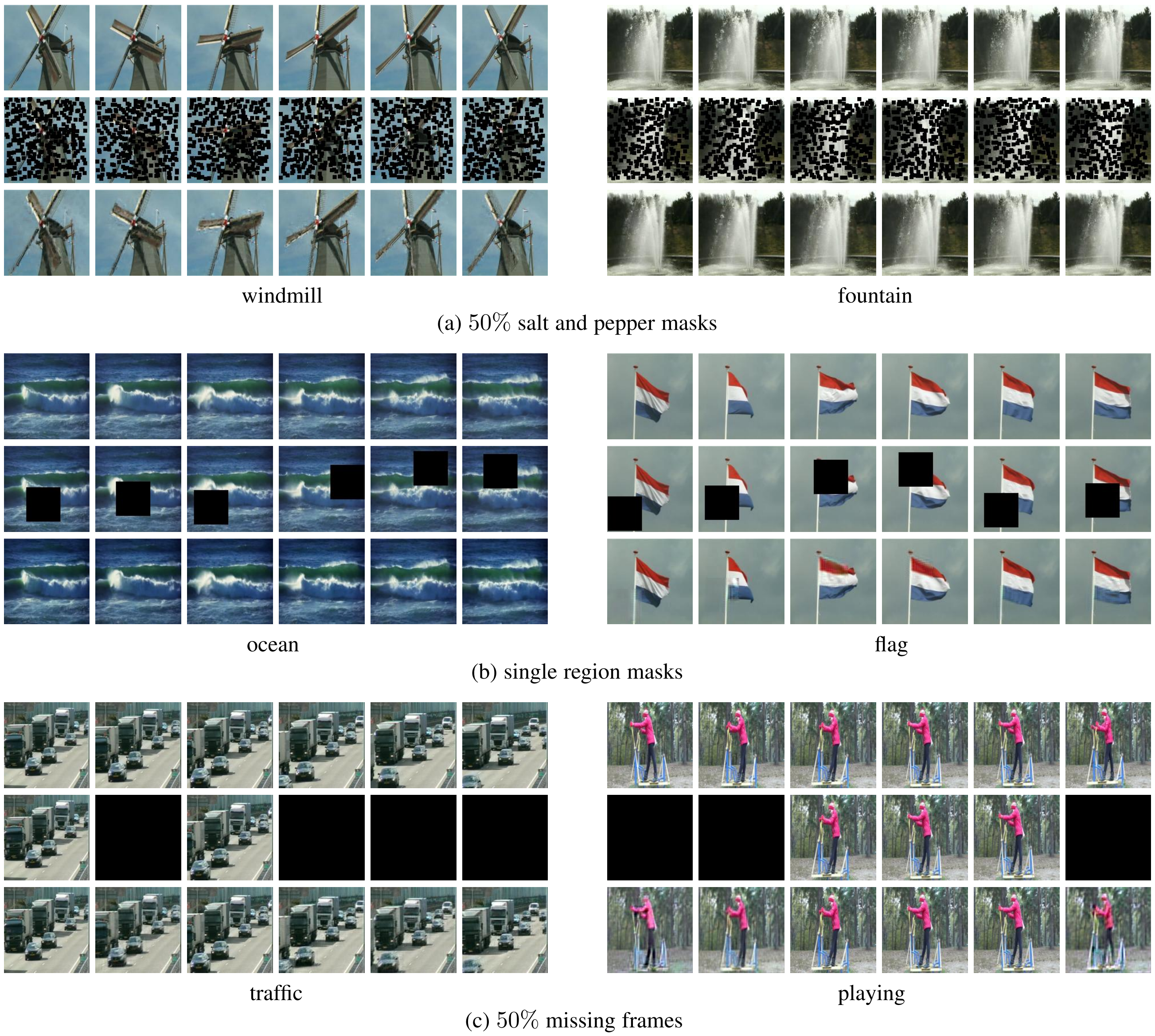
synthesized sequences

(c) running llamas

Fig. 7. Synthesizing action patterns without spatial or temporal stationarity. For each action video sequence, 6 continuous frames are shown. (a) running cows. Frames of 2 of 5 training sequences are displayed. The corresponding frames of 2 of 8 synthesized sequences generated by the learning algorithm are displayed. (b) running tigers. Frames of 2 observed training sequences are displayed. The corresponding frames of 2 of 4 synthesized sequences are displayed. (c) running llamas. Frames of 2 observed training sequences are displayed. The corresponding frames of 2 of 5 synthesized sequences are displayed. The observed sequences are of the size $100 \times 100$ pixels $\times 70$ frames.

To quantitatively evaluate the qualities of the recovered videos, we test our method on 7 video sequences, which are collected from DynTex++ dataset of [67], with 3 types of occlusions mentioned above. We use the same model structure as the one used in Experiment 3. The number of Langevin steps for recovery is set to be equal to the number of Langevin steps for synthesizing, which is 20. We run a single Langevin chain for synthesis. For each experiment, we report the recovery errors measured by the average per pixel difference between the original image sequence and the recovered image sequence on the occluded pixels. The range of pixel intensities is $[0, 255]$. We compare our results with the results obtained by a baseline method, which is a generic Markov random field model defined on the video sequence. The model is a 3D (spatial-temporal) Markov random field, whose potentials are pairwise $\ell_1$ or $\ell_2$ differences between nearest neighbor pixels, where the nearest neighbors are defined in both the spatial and temporal domains. The image sequences are recovered by sampling the intensities of the occluded pixels conditional on the observed pixels using the Gibbs sampler. Table 2 shows the comparison results for 3 types of occlusions. Our model significantly outperforms the baseline methods in terms of recovery errors. Note that the recovery errors are not training errors, because the intensities of the occluded pixels of the image frames are not observed in training.

We want to emphasize that (1) our models differ from denoising auto-encoders [73], where the training data are fully observed, and noises are added as a matter of regularization; (2) our experiments are different from in-painting or de-noising, where the prior model



windmill

fountain

(a) $50\%$ salt and pepper masks

ocean

flag

(b) single region masks

traffic

playing

(c) $50\%$ missing frames

Fig. 8. Learning energy-based spatial-temporal generative ConvNets from occluded video sequences. For each experiment, the first row shows a segment of the observed sequence, the second row shows a segment of the occluded sequence with black masks, and the third row shows the corresponding segment of the recovered sequence. The observed occluded sequences are of the size $150 \times 150$ pixels $\times 70$ frames. (a) type 1: salt and pepper mask. (b) type 2: single region mask. (c) type 3: $50\%$ missing frames.

or regularization has already been learned from fully observed data or provided; (3) Learning from incomplete data can be difficult for GAN and VAE.

### 4.5 Experiment 5: Background inpainting

If a moving object in the video is occluded in each frame, it turns out that the recovery algorithm will become an algorithm for background inpainting of videos, where the goal is to remove the undesired moving object from the video. We use the same model as the one in Experiment 2 for Figure 2. Figure 9 shows two examples of removals of (a) a moving boat and (b) a walking person respectively. The videos are collected from [74]. For each example, the first column displays 4 frames of the original video. The second column shows the corresponding frames with masks occluding the target to be removed. The third column presents the inpainting result by our algorithm. The video size is $130 \times 174$ pixels $\times 150$ frames in example (a) and $130 \times 230$ pixels $\times 104$ frames in example (b). The experiment is different from the video inpainting by interpolation. We synthesize image patches to fill in the empty regions of the video by running Langevin dynamics. We run a single Langevin chain for synthesis. As shown in Figure 9, our method successfully removes the target objects and inpaints the regions of the removed objects with reasonable backgrounds. Both appearance and motion of the inpainted video sequences are physically plausible.

## 5 Conclusion

In this paper, we propose an energy-based spatial-temporal generative ConvNet model for dynamic patterns, such as dynamic textures and action patterns. The model is in the form of deep spatial-temporal convolutional energy-based model where the energy function is defined by a bottom-up spatial-temporal ConvNet. The

**Algorithm 2** Learning, sampling, and recovery algorithm

```
Input:
    (1) training image sequences with occluded pixels $\{\mathbf{I}_m, m = 1, ..., M\}$
    (2) binary masks $\{O_m, m = 1, ..., M\}$ indicating the locations of the occluded pixels in the training image sequences
    (3) number of synthesized image sequences $\tilde{M}$
    (4) number of Langevin steps $l$ for synthesizing image sequences
    (5) number of Langevin steps $k$ for recovering the occluded pixels
    (6) number of learning iterations $T$
Output:
    (1) estimated model parameters $\theta$
    (2) synthesized image sequences $\{\tilde{\mathbf{I}}_m, m = 1, ..., \tilde{M}\}$
    (3) recovered image sequences $\{\mathbf{I}'_m, m = 1, ..., M\}$

Let $t \leftarrow 0$, initialize $\theta^{(0)}$.
Initialize $\tilde{\mathbf{I}}_m$, for $m = 1, ..., \tilde{M}$, by sampling from $q(\mathbf{I})$.
Initialize $\mathbf{I}'_m = \mathbf{I}_m$, for $m = 1, ..., M$.
repeat
    For each $m$, run $k$ steps of Langevin dynamics to recover the occluded region of $\mathbf{I}'_m$, i.e., starting from the current $\mathbf{I}'_m$, each step follows equation (8), but only the occluded pixels in $\mathbf{I}'_m$ (specified by $O_m$) are updated in each step.
    For each $m$, run $l$ steps of Langevin dynamics to update $\tilde{\mathbf{I}}_m$, i.e., starting from the current $\tilde{\mathbf{I}}_m$, each step follows equation (8).
    Calculate $H^{\text{obs}} = \sum_{m=1}^{M} \frac{\partial}{\partial\theta} f(\mathbf{I}'_m; \theta^{(t)})/M$, and $H^{\text{syn}} = \sum_{m=1}^{\tilde{M}} \frac{\partial}{\partial\theta} f(\tilde{\mathbf{I}}_m; \theta^{(t)})/\tilde{M}$.
    Update $\theta^{(t+1)} \leftarrow \theta^{(t)} + \eta(H^{\text{obs}} - H^{\text{syn}})$, with step size $\eta$.
    Let $t \leftarrow t + 1$
until $t = T$
```

model corresponds to a spatial-temporal discriminative ConvNet classifier in the sense that the latter can be directly derived from the former. This property makes our model natural for modeling dynamic patterns. The learning of the model is achieved by an "analysis by synthesis" scheme: we sample the synthesized examples from the current model, usually by Markov chain Monte Carlo (MCMC), and then update the model parameters based on the difference between the observed training examples and the synthesized examples. We show that the learning algorithm can be interpreted as an alternating mode seeking and mode shifting process, as well as an adversarial minimax optimization process.

Our experiments show that the model can synthesize different types of realistic dynamic patterns, with well designed spatial-temporal ConvNet structures serving as energy functions. Besides, it is possible to learn the model from videos with occluded pixels or missing frames. This can be achieved by adopting an extra Langevin dynamics starting from the corrupted training video sequences to recover the missing information. The resulting learning, sampling, and recovery algorithm is useful for unsupervised video inpainting, which includes video recovery (e.g., recovering missing pixels of frames of a video) and background inpainting (e.g., removing undesired moving object from a video). We show that learning from incomplete training data can provide an objective criterion to evaluate generative models of dynamic patterns.

TABLE 2
Recovery errors in occlusion experiments

(a) salt and pepper masks

| | ours | MRF-$\ell_1$ | MRF-$\ell_2$ |
|---|---|---|---|
| flag | **3.7923** | 6.6211 | 10.9216 |
| fountain | **5.5403** | 8.1904 | 11.3850 |
| ocean | **3.3739** | 7.2983 | 9.6020 |
| playing | **5.9035** | 14.3665 | 15.7735 |
| sea world | **5.3720** | 10.6127 | 11.7803 |
| traffic | **7.2029** | 14.7512 | 17.6790 |
| windmill | **5.9484** | 8.9095 | 12.6487 |
| Avg. | **5.3048** | 10.1071 | 12.8272 |

(b) single region masks

| | ours | MRF-$\ell_1$ | MRF-$\ell_2$ |
|---|---|---|---|
| flag | **8.1636** | 10.6586 | 12.5300 |
| fountain | **6.0323** | 11.8299 | 12.1696 |
| ocean | **3.4842** | 8.7498 | 9.8078 |
| playing | **6.1575** | 15.6296 | 15.7085 |
| sea world | **5.8850** | 12.0297 | 12.2868 |
| traffic | **6.8306** | 15.3660 | 16.5787 |
| windmill | **7.8858** | 11.7355 | 13.2036 |
| Avg. | **6.3484** | 12.2856 | 13.1836 |

(c) 50% missing frames

| | ours | MRF-$\ell_1$ | MRF-$\ell_2$ |
|---|---|---|---|
| flag | **5.5992** | 10.7171 | 12.6317 |
| fountain | **8.0531** | 19.4331 | 13.2251 |
| ocean | **4.0428** | 9.0838 | 9.8913 |
| playing | **7.6103** | 22.2827 | 17.5692 |
| sea world | **5.4348** | 13.5101 | 12.9305 |
| traffic | **8.8245** | 16.6965 | 17.1830 |
| windmill | **7.5346** | 13.3364 | 12.9911 |
| Avg. | **6.7285** | 15.0085 | 13.7746 |

## Acknowledgments

We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp GPU used for this research. The work is supported by ONR MURI N00014-16-1-2007, and DARPA ARO W911NF-16-1-0579.

## References

[1] G. Doretto, A. Chiuso, Y. N. Wu, and S. Soatto, "Dynamic textures," *International Journal of Computer Vision*, vol. 51, no. 2, pp. 91–109, 2003.

[2] Y. Wang and S.-C. Zhu, "A generative method for textured motion: Analysis and synthesis," in *European Conference on Computer Vision*, 2002, pp. 583–598.

[3] S. Ji, W. Xu, M. Yang, and K. Yu, "3D convolutional neural networks for human action recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 1, pp. 221–231, 2013.

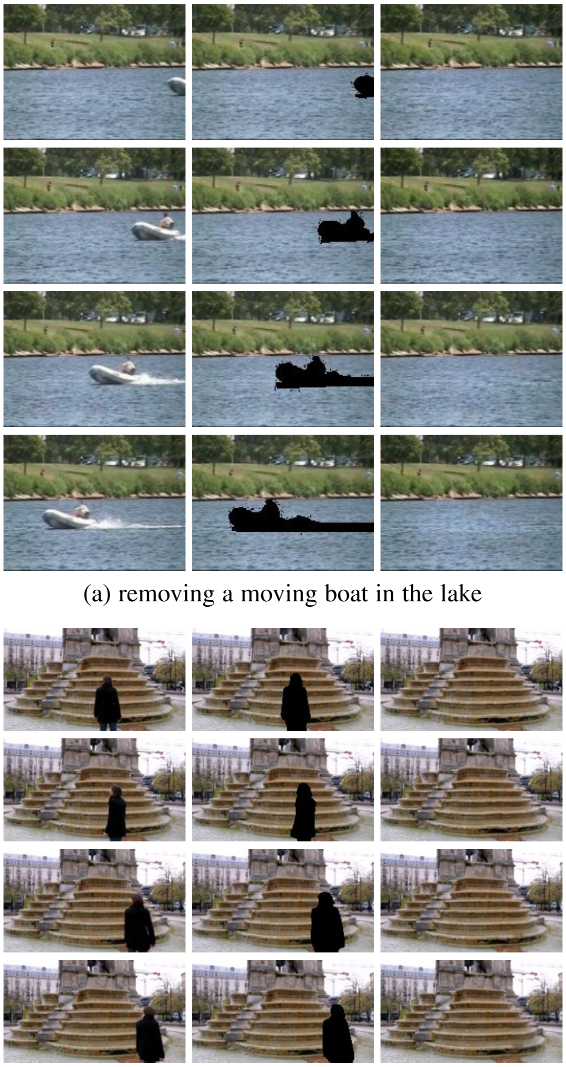
[4] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1725–1732.

[5] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in *Advances in Neural Information Processing Systems*, 2014, pp. 568–576.

[6] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3D convolutional networks," in *IEEE International Conference on Computer Vision*, 2015, pp. 4489–4497.

[7] J. Yue-Hei Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici, "Beyond short snippets: Deep networks for video classification," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 4694–4702.

[8] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Van Gool, "Temporal segment networks: Towards good practices for deep action recognition," in *European conference on computer vision*, 2016, pp. 20–36.



(a) removing a moving boat in the lake

(b) removing a walking person in front of fountain

Fig. 9. Background inpainting for videos. For each experiment, the first column displays 4 frames of the original video. The second column shows the corresponding frames with black masks occluding the target to be removed. The third column shows the inpainting result by our algorithm. (a) a moving boat in the lake ($130 \times 174$ pixels $\times 150$ frames). (b) a walking person in front of fountain ($130 \times 230$ pixels $\times 104$ frames).

[9] C. Feichtenhofer, A. Pinz, and R. P. Wildes, "Temporal residual networks for dynamic scene recognition," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 4728–4737.

[10] J. Carreira and A. Zisserman, "Quo vadis, action recognition? a new model and the kinetics dataset," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 6299–6308.

[11] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[12] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, 2012, pp. 1097–1105.

[13] A. Dosovitskiy, J. Tobias Springenberg, and T. Brox, "Learning to generate chairs with convolutional neural networks," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1538–1546.

[14] K. Gregor, I. Danihelka, A. Graves, D. J. Rezende, and D. Wierstra, "DRAW: A recurrent neural network for image generation," in *International Conference on Machine Learning*, 2015, pp. 1462–1471.

[15] E. L. Denton, S. Chintala, R. Fergus *et al.*, "Deep generative image models using a Laplacian pyramid of adversarial networks," in *Advances in Neural Information Processing Systems*, 2015, pp. 1486–1494.

[16] T. D. Kulkarni, W. F. Whitney, P. Kohli, and J. Tenenbaum, "Deep convolutional inverse graphics network," in *Advances in Neural Information Processing Systems*, 2015, pp. 2539–2547.

[17] Y. Lu, S.-C. Zhu, and Y. N. Wu, "Learning FRAME models using cnn filters," in *The Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[18] U. Grenander, "A unified approach to pattern analysis," *Advances in Computers*, vol. 10, no. 175-216, p. 15, 1970.

[19] U. Grenander, M. I. Miller *et al.*, *Pattern theory: from representation to inference*. Oxford university press, 2007.

[20] 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*, 2014, pp. 2672–2680.

[21] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[22] J. Xie, Y. Lu, S.-C. Zhu, and Y. N. Wu, "A theory of generative ConvNet," in *International Conference on Machine Learning*, 2016.

[23] Y. LeCun, S. Chopra, R. Hadsell, M. Ranzato, and F. Huang, "A tutorial on energy-based learning," in *Predicting Structured Data*. MIT Press, 2006.

[24] J. Ngiam, Z. Chen, P. W. Koh, and A. Y. Ng, "Learning deep energy models," in *International Conference on Machine Learning*, 2011, pp. 1105–1112.

[25] G. F. Montufar, R. Pascanu, K. Cho, and Y. Bengio, "On the number of linear regions of deep neural networks," in *Advances in Neural Information Processing Systems*, 2014, pp. 2924–2932.

[26] L. Younes, "On the convergence of Markovian stochastic algorithms with rapidly decreasing ergodicity rates," *Stochastics: An International Journal of Probability and Stochastic Processes*, vol. 65, no. 3-4, pp. 177–228, 1999.

[27] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[28] C. Feichtenhofer, A. Pinz, and R. Wildes, "Spatiotemporal residual networks for video action recognition," in *Advances in neural information processing systems*, 2016, pp. 3468–3476.

[29] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1–9.

[30] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev *et al.*, "The kinetics human action video dataset," *arXiv preprint arXiv:1705.06950*, 2017.

[31] S. Zhu and D. Mumford, "Grade: Gibbs reaction and diffusion equations." in *International Conference on Computer Vision*, 1998.

[32] S. C. Zhu, Y. N. Wu, and D. Mumford, "Minimax entropy principle and its application to texture modeling," *Neural Computation*, vol. 9, no. 8, pp. 1627–1660, 1997.

[33] Y. N. Wu, S. C. Zhu, and X. Liu, "Equivalence of Julesz ensembles and FRAME models," *International Journal of Computer Vision*, vol. 38, no. 3, pp. 247–265, 2000.

[34] J. Xie, W. Hu, S.-C. Zhu, and Y. N. Wu, "Learning sparse FRAME models for natural image patterns," *International Journal of Computer Vision*, pp. 1–22, 2014.

[35] J. Xie, Y. Lu, S.-C. Zhu, and Y. N. Wu, "Inducing wavelets into random fields via generative boosting," *Journal of Applied and Computational Harmonic Analysis*, 2015.

[36] R. Gao, Y. Lu, J. Zhou, S.-C. Zhu, and Y. N. Wu, "Learning generative ConvNets via multi-grid modeling and sampling," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 9155–9164.

[37] J. Xie, Y. Lu, R. Gao, S.-C. Zhu, and Y. N. Wu, "Cooperative training of descriptor and generator networks," *arXiv preprint arXiv:1609.09408*, 2016.

[38] T. Han, Y. Lu, S.-C. Zhu, and Y. N. Wu, "Alternating back-propagation for generator network," in *The Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[39] J. Xie, S.-C. Zhu, and Y. N. Wu, "Synthesizing dynamic patterns by spatial-temporal generative ConvNet," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 7093–7101.

[40] J. Xie, Z. Zheng, R. Gao, W. Wang, S.-C. Zhu, and Y. N. Wu, "Learning descriptor networks for 3D shape synthesis and analysis," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8629–8638.

[41] Y. Wang and S.-C. Zhu, "Analysis and synthesis of textured motion: Particles and waves," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 10, pp. 1348–1363, 2004.

[42] Z. Han, Z. Xu, and S.-C. Zhu, "Video primal sketch: A unified middle-level representation for video," *Journal of Mathematical Imaging and Vision*, vol. 53, no. 2, pp. 151–170, 2015.

[43] X. You, W. Guo, S. Yu, K. Li, J. C. Príncipe, and D. Tao, "Kernel learning for dynamic texture synthesis," *IEEE Transactions on Image Processing*, vol. 25, no. 10, pp. 4782–4795, 2016.

[44] C. M. Funke, L. A. Gatys, A. S. Ecker, and M. Bethge, "Synthesising dynamic textures using convolutional neural networks," *arXiv preprint arXiv:1702.07006*, 2017.

[45] M. Tesfaldet, M. A. Brubaker, and K. G. Derpanis, "Two-stream convolutional networks for dynamic texture synthesis," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 6703–6712.

[46] C. Vondrick, H. Pirsiavash, and A. Torralba, "Generating videos with scene dynamics," in *Advances in Neural Information Processing Systems*, 2016, pp. 613–621.

[47] M. Saito, E. Matsumoto, and S. Saito, "Temporal generative adversarial nets with singular value clipping," in *IEEE International Conference on Computer Vision*, 2017, pp. 2830–2839.

[48] S. Tulyakov, M.-Y. Liu, X. Yang, and J. Kautz, "MoCoGAN: Decomposing motion and content for video generation," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1526–1535.

[49] R. J. Williams and D. Zipser, "A learning algorithm for continually running fully recurrent neural networks," *Neural Computation*, vol. 1, no. 2, pp. 270–280, 1989.

[50] M. D. Zeiler, G. W. Taylor, and R. Fergus, "Adaptive deconvolutional networks for mid and high level feature learning," in *International Conference on Computer Vision*, 2011, pp. 2018–2025.

[51] J. Dai, Y. Lu, and Y. N. Wu, "Generative modeling of convolutional neural networks," *arXiv preprint arXiv:1412.6296*, 2014.

[52] Z. Tu, "Learning generative models via discriminative approaches," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2007, pp. 1–8.

[53] J. Lazarow, L. Jin, and Z. Tu, "Introspective neural networks for generative modeling," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2774–2783.

[54] L. Jin, J. Lazarow, and Z. Tu, "Introspective classification with convolutional nets," in *Advances in Neural Information Processing Systems*, 2017, pp. 823–833.

[55] K. Lee, W. Xu, F. Fan, and Z. Tu, "Wasserstein introspective neural networks," *arXiv preprint arXiv:1711.08875*, 2017.

[56] R. M. Neal, "Mcmc using hamiltonian dynamics," *Handbook of Markov Chain Monte Carlo*, vol. 2, 2011.

[57] M. Girolami and B. Calderhead, "Riemann manifold Langevin and Hamiltonian Monte Carlo methods," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 73, no. 2, pp. 123–214, 2011.

[58] H. Robbins and S. Monro, "A stochastic approximation method," *The Annals of Mathematical Statistics*, pp. 400–407, 1951.

[59] M. Welling, "Herding dynamical weights to learn," in *International Conference on Machine Learning*, 2009, pp. 1121–1128.

[60] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," *Proceedings of the National Academy of Sciences*, vol. 79, no. 8, pp. 2554–2558, 1982.

[61] H. S. Seung, "Learning continuous attractors in recurrent networks," in *Advances in Neural Information Processing Systems*, 1998, pp. 654–660.

[62] T. Han, Y. Lu, X. Xing, and Y. N. Wu, "Learning generator networks for dynamic patterns," in *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2019.

[63] J. Xie, R. Gao, Z. Zheng, S.-C. Zhu, and Y. N. Wu, "Learning dynamic generator model by alternating back-propagation through time," in *The Thirty-Third AAAI Conference on Artificial Intelligence*, 2019.

[64] D. P. Landau and K. Binder, *A guide to Monte Carlo simulations in statistical physics*. Cambridge university press, 2014.

[65] B. D. Ziebart, A. Maas, J. A. Bagnell, and A. K. Dey, "Maximum entropy inverse reinforcement learning," in *The Twenty-Third AAAI Conference on Artificial Intelligence*, 2008.

[66] M. Kumar, M. Babaeizadeh, D. Erhan, C. Finn, S. Levine, L. Dinh, and D. Kingma, "Videoflow: A flow-based generative model for video," *arXiv preprint arXiv:1903.01434*, 2019.

[67] B. Ghanem and N. Ahuja, "Maximum margin distance learning for dynamic texture recognition," in *European Conference on Computer Vision*, 2010, pp. 223–236.

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

[69] R. Costantini, L. Sbaiz, and S. Susstrunk, "Higher order SVD analysis for dynamic texture synthesis," *IEEE Transactions on Image Processing*, vol. 17, no. 1, pp. 42–52, 2008.

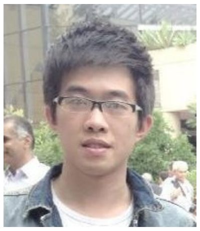
[70] B. Abraham, O. I. Camps, and M. Sznaier, "Dynamic texture with Fourier descriptors," in *Proceedings of the 4th International Workshop on Texture Analysis and Synthesis*, vol. 1, 2005, pp. 53–58.

[71] Q. Chen and V. Koltun, "Photographic image synthesis with cascaded refinement networks," in *International Conference on Computer Vision*, vol. 1, no. 2, 2017, pp. 1511–1520.

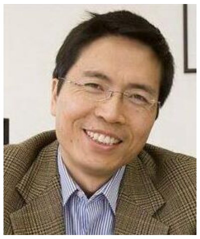
[72] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro, "High-resolution image synthesis and semantic manipulation with conditional gans," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8798–8807.

[73] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *Journal of machine learning research*, vol. 11, no. Dec, pp. 3371–3408, 2010.

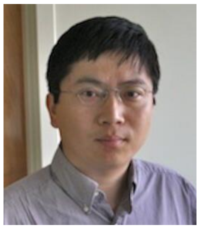
[74] A. Newson, A. Almansa, M. Fradet, Y. Gousseau, and P. Pérez. http://perso.telecom-paristech.fr/~gousseau/video_inpainting. [Online]. Available: http://perso.telecom-paristech.fr/~gousseau/video_inpainting/



**Jianwen Xie** received his Ph.D degree in statistics from University of California, Los Angeles (UCLA) in 2016. He received double M.S. degrees in statistics and computer science from UCLA in 2014 and 2012 respectively, and B.E. degree in software engineering from Jinan University, China in 2008. He is currently a senior research scientist at Hikvision Research Institute, USA. Before joining Hikvision, he was a staff research associate and postdoctoral researcher in the Center for Vision, Cognition, Learning, and Autonomy (VCLA) at UCLA from 2016 to 2017. His research interests focus on generative modeling and learning with applications in computer vision.



**Song-Chun Zhu** received his Ph.D. degree from Harvard University. He is currently professor of Statistics and Computer Science at UCLA. His research interests include vision, statistical modeling, learning, cognition, situated dialogues, robot autonomy and AI. He received a number of honors, including the Helmholtz Test-of-time award in ICCV 2013, the Aggarwal prize from the IAPR in 2008, the David Marr Prize in 2003 with Z. Tu et al. for image parsing, twice Marr Prize honorary nominations with Y. Wu et al. in 1999 for texture modeling and 2007 for object modeling respectively. He received the Sloan Fellowship in 2001, a US NSF Career Award in 2001, and an US ONR Young Investigator Award in 2001. He is a Fellow of IEEE since 2011.



**Ying Nian Wu** received his Ph.D. degree in statistics from Harvard in 1996. He was an assistant professor in the Department of Statistics, University of Michigan from 1997 to 1999. He joined University of California, Los Angeles (UCLA) in 1999, and is currently a professor in UCLA Department of Statistics. His research interests include statistical modeling, computing and learning, with applications in computer vision. He received Honorable Mention for the David Marr Prize with S. C. Zhu et al. in 1999 for texture modeling and 2007 for object modeling respectively.