DOMAIN-AGNOSTIC VIDEO PREDICTION FROM MOTION SELECTIVE KERNELS

Véronique Prinet

The Hebrew University of Jerusalem, Israel

ABSTRACT

Existing conditional video prediction approaches train a network from large databases and generalise to previously unseen data. We take the opposite stance, and introduce a model that learns from the first frames of a given video and extends its content and motion, to, e.g., double its length. To this end, we propose a dual network that can use in a flexible way both dynamic and static convolutional motion kernels, to predict future frames. We demonstrate experimentally the robustness of our approach on challenging videos in-the-wild and show that it is competitive w.r.t related baselines.

Index Terms— Frame Synthesis, Video Prediction, Motion Representation, Dynamic Kernels, Deep Learning, Neural Network

1. INTRODUCTION

We consider the problem of motion prediction for future frame synthesis. While the vast majority of the recent literature in the field is dedicated to learning forecasting models from (relatively) large databases, we focus our attention on learning from few samples. Being able to learn efficiently from a small dataset, exploiting a *good motion representation*, opens the door to a variety of new applications.

We explore for the first time predictive models that are *domain-agnostic* but *data-specific*. Our aim is to learn a model of a dynamic scene in the wild from a single video clip, and to extend/extrapolate its content and motion to, e.g., double its length. We are interested in any natural repetitive motions, such as a bird flapping wings or human walking (see figure 2).

Learning a predictive model from a single video in the wild is challenging: 1) the generic nature of the natural motion we are seeking to model is not suitable for loss-specific or architecture-specific networks of most existing methods; 2) the choice of videos-in-the-wild implies a model capable of robust background-foreground decomposition, to be able to recover large background regions occluded by the foreground in previous frames –something that no work of our knowledge so far has demonstrated; 3) learning from a short clip requires a quick and efficient convergence of the model at training time.

Our model is related to two different lines of work tackling the issue of motion prediction. The first one is concerned with dynamic filters, i.e., methods that infer input-dependent weights of a convolutional or LSTM network at each time-step, and apply these filters to a frame to predict the next one (e.g., [12, 5]). The second one refers to disentangled representations from unsupervised learning (i.e., separating the causes from the effect) –approaches that usually implicitly assume simple background or semi-rigid motion [9, 14, 26].

Future frames are generated one by one, in a recursive fashion. The model takes as input a context (i.e., historical frames) and predicts the next frame. We propose a motion representation based on a dual network: one stream applies a transformation to an input and generate the next frame; the second stream selects dynamically a subset of the kernels of the transformation network which will effectively be active for a given input. Our contributions can be summarized as follows:

1. We propose the first DNN-based model that attempts to extrapolate the content and motion of a given videoclip.
2. We introduce a predictive network that jointly learns dynamic and static elementary convolutional kernels. Its architecture makes it robust enough to learn efficiently from a few data.
3. We validate our approach on natural challenging videos with cluttered background, multiple and complex motions, and no particular semantic domain, with mid-range (10-30 frames) prediction.

2. RELATED WORK

Motion representation

Motion representation is a long-standing open problem in visual perception studies and computer vision [7, 18, 4, 13, 31]. Visual illusions show strong evidence that the perceived motion between consecutive images strongly depends on the image structure itself [31]. The first attempt to develop a parametric statistical model that explicitly captures the conditional dependence between the flow field and the input image structure, might be attributed to Sun etal. [23]. More recently, variational auto-encoders (VAE) have been shown to be an efficient means to modelling motion with learned prior [10].

Our model also learns input-dependent constraints on the flow field, albeit in a non-stochastic manner.

Video texture synthesis Dynamic texture (or textured motion) are sequences of images of moving scenes or objects that exhibit certain harmonic or stationary properties in time, often encountered in natural scenes (eg, fluid flow, clouds). Early parametric [21, 28] and non-parametric [2] approaches were mostly suitable to model global dynamic systems. Layered representations [7, 8] and deep non-linear dynamics [34, 32] were then introduced to improve the expressive power of these models. Our work took inspiration from deep non-linear auto-regressive models, in a similar fashion to [34]. However, in contrast to those cited approaches, our model can take advantage, but is not limited to, dynamic textures patterns.

Video frame prediction Recent years have sparked huge interest in conditional video prediction [12, 17, 26, 19, 33, 22, 9, 10, 25, 27, 3, 1, 16]. The goal is to generate future frames given a few frames history –a 'context'. Most closely related to our work, some approaches represent motion using a set of input-dependent convolution filters, that operate on an image pyramid or at image full resolution [33, 12, 5, 27]. Amongst those, [27] is the only one of our knowledge which proposes a predictive model for domain-agnostic immediate future video frame generation. The authors use a sole adversial loss to constrain the network, thus accounting for the uncertainty of the future. It is however restricted to short-term prediction, while we aim at exploring long-range solutions. Besides, a large body of work address the issue of disentangling video content from motion, with some applications to video synthesis [17, 26, 9, 14]. Most of those share a same basic principle: a dedicated network architecture, which hard-codes the decomposition between motion/pose and content, by the means of two distinct encoders, and the use of LSTM. In contrast, we propose a soft mechanism which simply learns how to distinguish the moving foreground from the background. This enables us, in particular, to recover occluded background regions, something not possible from existing techniques.

3. METHOD

3.1. Overview

We aim at learning an auto-regressive sequence model P_ζ, to predict $T-\delta$ future frames, given δ observed ones, $\mathbf{x}_{<\delta}$. Applying the product rule, the conditional likelihood over the future frames, $\mathbf{x}_{\delta:T}$, can factorized as:

$$\mathcal{L}(\zeta) = P_\zeta(\mathbf{x}_{\delta:T}|\mathbf{x}_{<\delta}) = \prod_{t'=\delta}^{T-1} P_\zeta(\mathbf{x}_{t'+1}|\tilde{\mathbf{x}}_{t'-\delta:t'}), \qquad (1)$$

where the first frames of the time-series are observed, i.e., $\tilde{\mathbf{x}}_{0:\delta} = \mathbf{x}_{0:\delta} = \mathbf{x}_{<\delta}$ ($\mathbf{x}$ refers to the ground-truth image, and $\tilde{\mathbf{x}}$ is a generated one). We use a δ-order markovian assumption: predictions are independent conditionally of the past few frames. Future frames can be generated recursively one by one, each newly generated frame, $\tilde{\mathbf{x}}_t$, feeding the model for the next time step. The set of parameters $\zeta = \{\Phi, \theta\}$ defines the model. We learn P_ζ by minimizing the negative logarithm of equation (1), so that: $\zeta = -\arg\min_\zeta \log L(\zeta) = \arg\min_\zeta E(\zeta)$.

Fig. 1. Sample frames from our video-clip dataset.

Our predictive model, P_ζ, is based on two nested modules: (i) a transformation model G_θ (or *transformer*), which generates the next frame $\tilde{\mathbf{x}}_t$, by transforming the previous ones, $\mathbf{x}_{t-\delta:t}$, via a series of elementary motion kernels, $W^l_{.,n}$. The size, orientation and activation amplitude of those kernels determine the transformation to be applied to the input. This encompasses both object displacement (similar to local image warping), and new pixel generation (that uncover occluded regions). (ii) a selection model S_Φ (or *selector*), whose role is to choose, at each time step, which subset amongst the available motion kernels of G_θ is the most efficient to perform the desired transformation, conditioned on the input data. Specifically, the selection model outputs a probability mass function over the kernels indices of the transformation model.

Our image prediction model can thus be written as follows:

$$\begin{aligned}\mathcal{T}_\zeta : \mathbf{x}_{t-\delta:t} \mapsto \tilde{\mathbf{x}}_{t+1} &= G_{\theta,S_\Phi}(\mathbf{x}_{t-\delta:t}) \\ &= G_\theta(\mathbf{x}_{t-\delta:t}; S_\Phi(\mathbf{x}_{t-\delta:t})).\end{aligned} \qquad (2)$$

3.2. Direction selective motion kernels

The transformation model G_θ and selection model S_Φ are nested deep networks, represented by a U-shaped encoder-decoder for the former [15], an encoder for the latter.

Given an input clip $\mathbf{x}_\tau$, $\tau = [t-\delta, t]$, lets define the matrix $\hat{\boldsymbol{\alpha}}(\mathbf{x}_\tau) = S_\Phi(\mathbf{x}_\tau)$ so that $\hat{\boldsymbol{\alpha}} \in [0,1]^{L/2\times N}$ and $\int_{n=1}^{N} \hat{\alpha}^l_n = 1$. Each element $\hat{\alpha}^l_n \in \hat{\boldsymbol{\alpha}}$, inferred from S_Φ, is a weighting scalar that will act upon the convolutional filters of G_θ. L and N are respectively the total number of hidden layers and the number of channels per layer of the decoder in G_θ. Hence, at each layer l of the *decoder* of the transformation model, we

define the linear transformation applied to an hidden feature map $\mathcal{Y}^{l-1}$, as follows:

$$\begin{aligned}\alpha_n^{l-1} &\leftarrow N\hat{\alpha}_n^{l-1}(\mathbf{x}_\tau)\\ \mathcal{Z}_{n'}^{l} &= \sum_{n=0}^{2N-1}[\mathcal{Y}^{L-l};\ \alpha^{l-1}\,\mathcal{Y}^{l-1}]_n * W_{n,n'}^{l} \qquad (3)\\ &= \sum_{n=0}^{N-1}\mathcal{Y}_n^{L-l} * W_{n,n'}^{l} + \sum_{n=N}^{2N-1}\mathcal{Y}_n^{l-1} * W_{n,n'}^{l}\,\alpha_n^{l-1},\end{aligned}$$

where $*$ denotes the convolution operation, $l \in \{L/2, ..., L-1\}$ and $n' \in \{0, ..., N-1\}$. $[A^a; B^b]$ refers to the concatenation of feature maps A originating from the encoder at layer a of the net, via the skip-connection, with feature maps B at layer b, along the depth/channel dimension. $W_{n,n'}^{l} \in \mathbb{R}^{f^2}$ is the weights matrix of the $f \times f$ motion kernels. In the above equations and the subsequent ones, we omit the bias term (here $+b_n$ on the RHS), for the sake of compactness and simplicity. The input- and time-dependent behaviour of the transformer's kernels appears in the second term of the RHS of equation (3): $W_{n,n'}^{l}\alpha_n^{l-1} = N\,W_{n,n'}^{l}\hat{\alpha}_n^{l-1}(\mathbf{x}_\tau)$. The scalars $\hat{\alpha}_n^{l-1}(\mathbf{x}_\tau)$ modulate and modify the behaviour of the convolutional kernels at each time step, hence confer to the network a greater flexibility.

For the sake of completeness, we finally write down the expression of the very first and very last building blocks of the transformer network:

$$\begin{aligned}\mathcal{Z}_{n'}^{0} &= \sum_{t'=t-\delta}^{t}\mathbf{x}_{t'} * W_{t',n'}^{0}, \quad \mathcal{Y}_{n'}^{0} = \rho_0(\mathcal{Z}_{n'}^{0}),\\ \mathcal{Z}_{n'}^{L} &= \sum_{n=0}^{2N-1}[\mathcal{Y}^{0};\ \alpha^{L}\,\mathcal{Y}^{L-1}]_n * W_{n,n'}^{L}, \ \ \tilde{\mathbf{x}}_{t+1} = \rho_L(\mathcal{Z}^{L}).\end{aligned}$$

where $\rho()$ is the non-linearity function.

3.3. Loss function

We use the L_1 norm as reconstruction loss, in addition to which we introduce a motion loss, minimizing for the total variation in the time domain:

$$\ell_{L_1}(\mathbf{x}_t) = |\tilde{\mathbf{x}}_t - \mathbf{x}_t| \qquad (4)$$

$$\ell_{motion}(\mathbf{x}_t) = ||\tilde{\mathbf{x}}_t - \tilde{\mathbf{x}}_{t-1}| - |\mathbf{x}_t - \mathbf{x}_{t-1}|| \qquad (5)$$

The motion loss explicitly forces the network to account for the temporal changes between consecutive frames. Hence the total per-batch loss function can be written:

$$E(\zeta) = \sum_{t=t'}^{t'+K}\big(\ell_{L_1}(\mathbf{x}_t) + \mu_{motion}\,\mathbb{1}_{t>t'}\ell_{motion}(\mathbf{x}_t)\big), \qquad (6)$$

where μ_{motion} is a factor weighting the two terms, $\mathbb{1}$ is the indicator function and K is the time-range prediction in the future at training time. Note that we do not impose any direct constraint on the output of the selector, $\boldsymbol{\alpha}(\mathbf{x}) = S_\Phi(\mathbf{x})$.

We learn the model end-to-end in a fully unsupervised manner. We employ training in stages with tasks of increasing difficulty [30].

4. EXPERIMENTAL RESULTS

Videos and additional results are available online[1]. Please refer to our website and long version arXiv paper [20] for the complete specifications regarding architecture, training procedure and data.

We quantitatively and qualitatively evaluate our approach on several video-clips in-the-wild. The accuracy of the reconstruction is measured in terms of PSNR and SSIM [29], averaged over the length of the predicted sequence. We run and compare:

- B0 *Baseline-0*. Reference baseline, no prediction. We compute the error between the last input frame, and the next frame.
- B1 *Baseline-1*. Encoder-encoder. The sole transformation model $G_\theta()$ is trained, the selection model is inactive; we set $\mu_{motion} = 0$.
- M1 *DN w/o motion loss*. Our dual net model —$G_\theta()$ and $S()_\Phi$ are trained jointly; we set $\mu_{motion} = 0$.
- M2 *FDN*. Our dual net model, trained with motion loss. We set $\mu_{motion} = 10$, unless specified otherwise.

Quantitative results are given in Table 1. Visual illustrations are shown on Figure 2 for the Bird sequence, on our website for the other clips.

Bird. The sequence was downloaded from Youtube, cropped and resized to 256×256 pixels. It comprises 80 frames, 50 of which being used for training. Motion is learned with a conditioning of four frames. For testing, we input to the net four frames that it has not seen at training, and predict 25 future frames. Figure 2 shows that our approach synthesises correctly motion and appearance, while the baseline tends to introduce color artefact.

Boy on a bicycle. This example illustrates a sequence with cluttered textured background. The video was acquired with a Canon EOS camera, with a resolution of 1280×720, then cropped and resized to 100×320 pix. It comprises 57 frames, 30 of which being used for training. Motion is learned with a conditioning of three frames. To illustrate the results, we feed the net with three frames of the sequence unseen at training time, and predict future frames until the boy leaves the camera's field of view. While competitive methods reproduce correctly the translational motion, our approach tends to be more faithful in terms of foreground details and object shape contours.

[1] https://cs.huji.ac.il/~prinet/project_pages/VideoPredict/

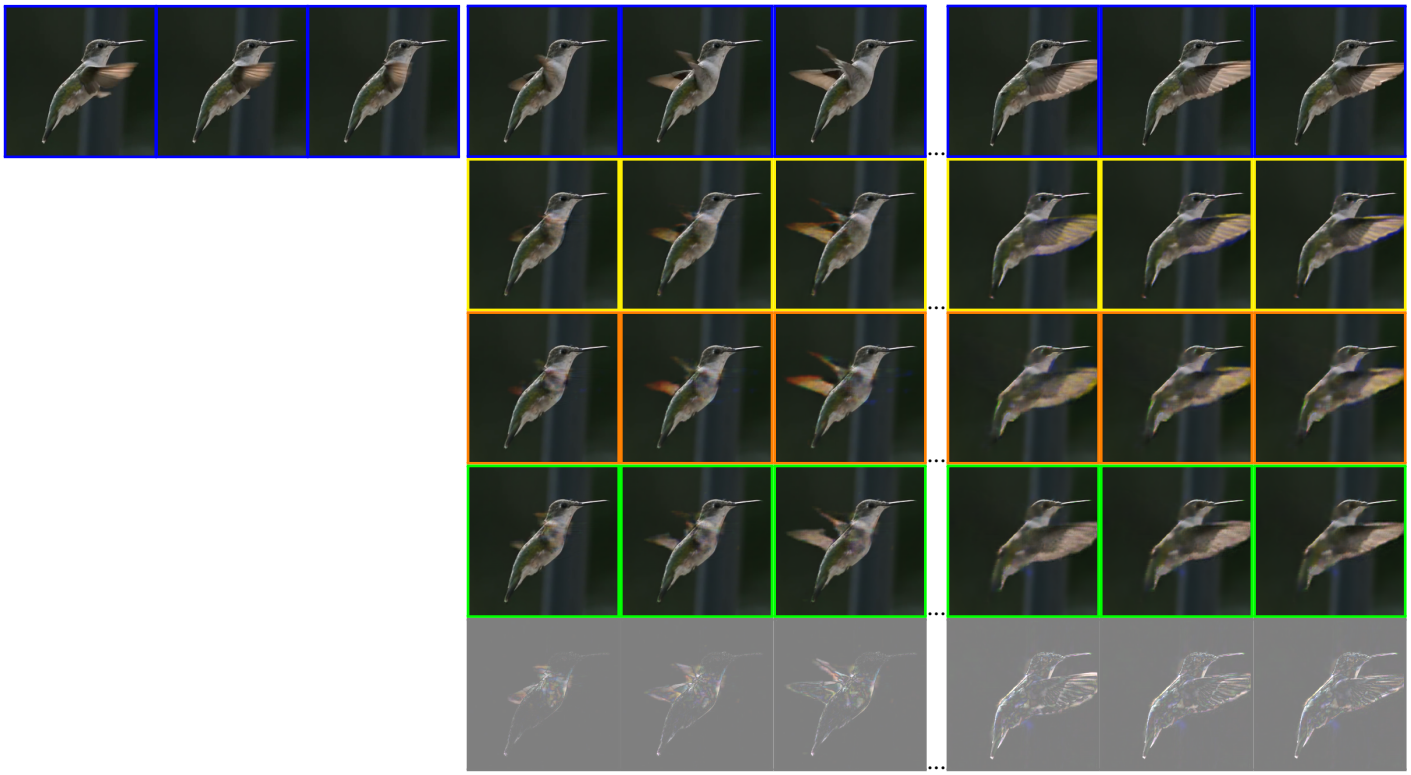

Fig. 2. Bird sequence. Blue frames: input conditioning frames and six frames (three first and three last frames, out of 25) ground-truth (that the model has not seen at training). Yellow frames: prediction results from B1. Orange frames: prediction results from M1. Green frames: prediction results from M2 (our FDN). Gray frames: L_2 error between ground-truth and our full model (FDN) prediction.

	B0	B1	M1	M2 (FDN)
Bird	22.2	23.1/0.913	23.6/0.922	**24.2/0.923**
Garden	19.5	20.3/0.682	**20.5/0.70**	20.42/0.695
Ocean	25.6	26.1/0.943	27.06/**0.963**	**27.7** /0.955

Table 1. Quantitative analysis (average PSNR/SSIM over the predicted sequence length), for the Bird, Garden and Ocean.

Ocean sequence. We selected a sequence from the YUP++ dataset [11] depicting ocean waves and a boat moving (static camera # 28, Ocean category), that we cropped to 200^2 pix and down-sampled in the time domain, to eventually get a sequence of 50 frames. The boat displacement is uniform, while the waves are characterized by harmonic oscillations. The colors are tern, without good contrast between the boat's hull and the sea. We learn the model from 20 frames, using three frames for conditioning. We predict over the next 26 frames. The motion loss, accounted for only in our full model, allows our model to distinguish correctly the sea from the boat's hull.

Cat sequence. The Cat sequence was downloaded from Youtube, and subsampled in time and space by a factor of two. It comprises 32 frames (105×320 pixels). The motion reflects the global translational displacement and the local movement of the cat's legs. To illustrate the results, we feed the net with three frames that have been seen during training (no 25 and onward), and predict over thirty frames (with no ground truth available for most of the predicted sequence). The visual results (see our project web page) shows sharper contour and better motion forecast from our model, in comparison to the baselines (B1, M1).

5. CONCLUSION

We have introduced a model for future frame synthesis from a single video-clip in-the-wild. Inspired initially by the mechanism of Direction Selective cells in the retina (see [6, 24]), our motion representation is based on a dual network: one that learns kernels, and a second one which dynamically selects the best subset for next frame prediction. Our approach compares favourably with respect to baseline methods on challenging videos. Future research directions include video manipulation and motion (de)composition.

6. REFERENCES

[1] M. Babaeizadeh, C. Finn, D. Erhan, R. Campbell, and S. Levine. Stochastic variational video prediction. In *ICLR*, 2018.

[2] Z. Bar-Joseph, R. El-Yaniv, D. Lischinski, and M. Werman. Texture mixing and texture movie synthesis using statistical learning. *IEEE Transactions on Visualization and Computer Graphics archive*, 7(2):120–135, 2002.

[3] P. Bhattacharjee and S. Das. Temporal coherency based criteria for predicting video frames using deep multi-stage generative adversarial networks. In *NIPS*, 2017.

[4] M. Black and P. Anandan. The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields. *Computer Vision and Image Understanding*, 63, 1996.

[5] B. D. Brabandere, X. Jia, T. Tuytelaars, and L. V. Gool. Dynamic filter networks. In *NIPS*, 2016.

[6] K. L. Briggman, M. Helmstaedter, and W. Denk. Wiring specificity in the direction-selectivity circuit of the retina. *Nature*, 471:183–188, March 2011.

[7] A. Chan and N. Vasconcelos. Layered dynamic texture. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 2009.

[8] Y. Chuang, D. Goldman, K. Zheng, B. Curless, D. Salesin, and R. Szeliski. Animating pictures with stochastic motion texture. *ACM Transactions on Graphics*, 2005.

[9] E. Denton and V. Birodkar. Unsupervised learning of disentangled representations from video. In *NIPS*, 2017.

[10] E. Denton and R. Fergus. Stochastic video generation with a learned prior. In *ICML*, 2018.

[11] K. Derpanis, M. Lecce, K. Daniildis, and R. Wildes. Dynamic scene understanding: The role of orientation features in space and time in scene classification. In *CVPR*, 2012.

[12] C. Finn, I. Goodfellow, and S. Levine. Unsupervised learning for physical interaction through video prediction. In *NIPS*, 2016.

[13] D. Fleet. Design and use of linear models for image motion analysis. *International Journal of Computer Vision*, 36, 2000.

[14] J. Hsieh, B. Liu, D. Huang, L. Fei-Fei, and J.-C. Niebles. Learning to decompose and disentangle representations for video prediction. https://arxiv.org/abs/1806.04166, 2018.

[15] P. Isola, J.-Y. Zhu, J. Zhou, and A. Efros. Image-to-image translation with conditional adversarial nets. In *CVPR*, 2017.

[16] A. Lee, R. Zhang, F. Ebert, P. Abbeel, C. Finn, and S. Levine. Stochastic adversarial video prediction. lhttps:arXiv:1804.01523v1.

[17] W. Lotter, G. Kreiman, and D. Cox. Deep predictive coding networks for video prediction and unsupervsed learning. In *ICLR*, 2017.

[18] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Imaging Understanding Workshop*, 1981.

[19] M. Mathieu, C. Couprie, and Y. LeCun. Deep multiscale video prediction beyond mean square error. In *ICLR*, 2016.

[20] V. Prinet. Motion selective prediction for video frame synthesis. *CoRR*, http://arxiv.org/abs/1812.10157, 2018.

[21] S. Soatto, G. Doretto, and Y. Wu. Dynamic textures. In *ECCV*, 2001.

[22] N. Srivastava, E. Mansimov, and R. Salakhutdinov. Unsupervised learning of video representations using lstms. In *ICML*, 2015.

[23] D. Sun, S. Roth, J. Lewis, and M. Black. Learning optical flow. In *ECCV*, 2008.

[24] W. Sun, Q. Deng, W. Levick, and S. He. On direction-selective ganglion cells in the mouse retina. *The Journal of Physiology*, pages 197–202, 2006.

[25] S. Tulyakov, M. Liu, X. Yang, and J. Kautz. Mocogan: Decomposing motion and content for video generation. In *CVPR*, 2018.

[26] R. Villegas, J. Yang, S. Hong, X. Lin, and H. Lee. Decomposing motion and content for natural video sequence prediction. In *ICLR*, 2017.

[27] C. Vondrick and A. Torralba. Generating the future with adversarial transformers. In *CVPR*, 2017.

[28] Q. Wang and S. Zhu. Modeling textured motion: particle, wave and sketch. In *ICCV*, 2003.

[29] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 2004.

[30] D. Weinshall, G. Cohen, and D. Amir. Curriculum learning by transfer learning: Theory and experiments with deep networks. In *ICML*, 2018.

[31] Y. Weiss, E. Simoncelli, and E. Adelson. Motion illusions as optimal percepts. *Nature Neuroscience*, 5(6):598–604, June 2002.

[32] J. Xie, S. Zhu, and Y. Wu. Synthetising dynamic patterns by spatial temporal generative conv net. In *CVPR*, 2017.

[33] T. Xue, J. Wu, K. L. Bouman, and W. T. Freeman. Visual dynamics: Probabilistic future frame synthesis via cross convolutional network. In *NIPS*, 2016.

[34] X. Yan, H. Chang, A. Shan, and X. Chen. Modeling video dynamics with deep dynencoder. In *ECCV*, 2014.