View-LSTM: Novel-View Video Synthesis Through View Decomposition

Mohamed ILyes Lakhal[1], Oswald Lanz[2], Andrea Cavallaro[1]
[1] CIS, Queen Mary University of London, [2] TeV, Fondazione Bruno Kessler
{m.i.lakhal,a.cavallaro}@qmul.ac.uk, lanz@fbk.eu

Abstract

We tackle the problem of synthesizing a video of multiple moving people as seen from a novel view, given only an input video and depth information or human poses of the novel view as prior. This problem requires a model that learns to transform input features into target features while maintaining temporal consistency. To this end, we learn an invariant feature from the input video that is shared across all viewpoints of the same scene and a view-dependent feature obtained using the target priors. The proposed approach, View-LSTM, is a recurrent neural network structure that accounts for the temporal consistency and target feature approximation constraints. We validate View-LSTM by designing an end-to-end generator for novel-view video synthesis. Experiments on a large multi-view action recognition dataset validate the proposed model.

1. Introduction

Generating a video from cues such as a textual description, information on a specific object or scene type, or a single frame is an interesting challenge with applications in data augmentation and action imitation. Generating a realistic video without specific priors is a challenging task. Deep-generative models can synthesize (predict) a video using an approximation of the density distribution of the data (probabilistic methods [16, 30, 19]) or an input representation (context-based methods [42, 41, 51]). Temporal Generative Adversarial Nets (TGAN) [30] produces a fix length vector that corresponds to a latent representation of frames that is used in the decoder of the GAN [7] structure to synthesize a video.

Recent works [55, 23] include priors to help the generation but produce more intractable results. For example, style-content based models [11, 39, 38] separate the generation as content generation (*e.g.* background scene) from the generation of the motion or dynamics. Pose Guided [54, 2] models rely on the 2D body pose information as guidance in the generation process. Other methods predict Optical flow and use it along with a conditioned image to synthesize the next frame [5] or a sequence of frames [24].

Multi-view data can be decomposed into a generic, view-invariant component, and a view-dependent component. This concept was used for 3D object generation via the so-called intact space and noise as prior [52]. Feature aggregation between view-invariant and view-dependent information is performed using summation. However, joining two different feature points as summation for real world data such as image or video stream might not be appropriate with deep learning models. Features are coming from non-linear mapping and using a linear operator (summation) as a way to combine them can damage important feature information. In fact concatenation is preferred in recent architectures [21, 45, 41]. Similarly, features can be projected into appearance space that is common for the same 3D object and pose space that contains the object 3D orientation [25].

We propose View-LSTM, a novel convolutional Long Short-Term Memory (LSTM) structure that takes advantage of the temporal learning capability of Recurrent Neural Network (RNN) to approximate the target view sequence in the feature space. It does so by learning to aggregate a view-invariant representation of the input view with view-dependent information from the target prior. Moreover, we extend the perceptual loss [14] to account for relevant temporal information as needed in our video prediction task. For this we use the features obtained from the intermediate layers of a 3D CNN as spatiotemporal representations and verify empirically that Conv-LSTM implicitly learns invariant feature representation.

2. Related Work

Recurrent structures can predict frames from a sequence of previous frames. Spatiotemporal LSTM (ST-LSTM) [48] keeps track of a memory cell between subsequent time steps to prevent the vanishing gradient problem. PredRNN++ [46] extends ST-LSTM to allow deeper stacking of layers through an additional gating mechanism. Eidetic 3D LSTM [47] extends ST-LSTM to 3D convolution inside the RNN gating computation thus enabling local temporal memory. The problem of spatiotemporal prediction using stationary and non-stationary components can be addressed with two modules inside the recurrent structure that achieve longer predictions [49]. All the aforementioned single-view predictive methods were tested on small datasets only, such as KTH [31] and Moving-MNIST [35].

Long Short-Term Attention (LSTA) [36] extends the Conv-LSTM [33] to focus on relevant spatial parts through attention pooling for smooth temporal tracking. The model was used for Egocentric Action Recognition with cross-modal fusion in a two-stream architecture. Coupled Recurrent Network (CRN) [37] is a two-stream architecture using independent Conv-LSTMs for each stream. The results of the two streams are fused to distill reciprocal representations. However, these methods are not directly applicable to novel-view video synthesis. Because they keep track of a separate hidden state memory for each stream (modality) whereas for the synthesis problem we only maintain one hidden state that represents the appearance feature.

Novel-view rendering has been applied to faces and human bodies. Deep Appearance Model [26] matches object shape (mesh) and appearance (texture) to a new (unseen) viewpoint. The model uses an autoencoder whose encoder uses a mesh and an average texture from all the views. The novel-view face rendering is then obtained using the compact feature from the encoder and a target viewpoint as an input to the decoder. Estimating the 3D full body mesh with the pose [43] is an active area of research. Synthesizing a full body is based on the availability of prerecorded body scans [1]. Self-supervised models also exist for frame-based novel view synthesis of multiple subjects using a static background as guidance and by decomposing the image into a latent representation that corresponds to rotation and translation matrices [28]. However, our problem is different in two aspects. First, we target the temporal domain and thus need to ensure consistency across the synthesized frames. Second, we need to cope with natural variations in the background of different views.

View-invariant action representations from an input frame and a ground-truth target prior can synthesize a target-view optical flow [23] using a global temporal feature learning with Bidirectional-LSTM [8]. Unlike our problem, this model does not have to hallucinate the background. Actions can also be synthesized from a given view to a target view from non-overlapping input and output time frames [44]. This model restricts the learning of the motion representation to a set of predefined patterns to synthesize the action in the target view. We instead focus on learning to approximate the target view in the feature space with more freedom in the types of motions that can be synthesized.

3. Temporal target feature approximation

We define the *view decomposition* assumption for a feature point ϵ^i of view i as a decomposition into an invariant feature z, which is common to all the views, and a view-dependent feature π^i, which is specific to the view i. z and π^i are combined through the operator ψ:

$$\epsilon^i = \psi(z, \pi^i). \tag{1}$$

In multi-view video synthesis, we are only given an input video sequence I^i from a view i and a set of M priors $\mathcal{P}^j = \{P^j_m\}^M_{m=1}$ of a target view j, where I^i and P^j_m have T frames.

Suppose that we are given a decoder f_D that takes a spatiotemporal appearance feature point of view j and decodes it back to the pixel space I^j. Hence the problem of multi-view video synthesis is reduced to approximating the target spatiotemporal appearance feature point ϵ^j of view j.

In what follows, we provide details on how to obtain each component of the *view decomposition* assumption. We show an architectural constraint on enforcing the invariance to obtain z. Then π^j is presented as a linear combination of the encoded feature representation of the prior P^j_m for $m \in [1..M]$. Finally, we present View-LSTM that extends the Conv-LSTM [33] to implement the aggregator ψ.

3.1. Invariant feature

We extract the feature input (resp. target) view sequence I^i (resp. I^j) using the encoder f^i_E (resp. f^j_E). A straightforward way to enforce invariance is to share the weights of f^i_E and f^j_E using Siamese architecture [4] as in FD-GAN [6] for example.

Let W_E be the shared weight between the encoder f^i_E and f^j_E. The backpropagation using the reconstruction loss $\mathcal{L}_r$ with respect to W_E is given as:

$$\frac{\partial \mathcal{L}_r}{\partial W_E} = \frac{\partial \mathcal{L}_r}{\partial z^i}\frac{\partial z^i}{\partial W_E} + \frac{\partial \mathcal{L}_r}{\partial z^j}\frac{\partial z^j}{\partial W_E}, \tag{2}$$

where the term z^i (resp. z^j) is given as $f^i_E(I^i)$ (resp. $f^j_E(I^j)$). The weights of f^j_E will also be affected by the error coming from the reconstruction loss. We found empirically that this affects the synthesis process where the model failed to synthesize the target sequence.

Recall that a mapping f_E is said to be invariant if $f_E(x) = f_E(y)$ where $x \neq y$. In order to enable the model to synthesize a video sequence, a simple solution is to separate the parameters of f^i_E and f^j_E. Since in neural network the mapping has learnable weights and the encoder f^j_E only serves as a guide during the training for enforcing the invariance, we use the same architecture for f^i_E and f^j_E and after training with an invariance loss the weights of both encoders will have equivalent values. We therefore have: $z = f^i_E(I^i)$ and $z \approx f^j_E(I^j)$ such that z has $T^{'} \leq T$.

3.2. View-dependent feature

Let us define the encoder $g^{j,m}_E$ that maps each prior P^j_m of the view j to a lower dimensional feature π^j_m. We define

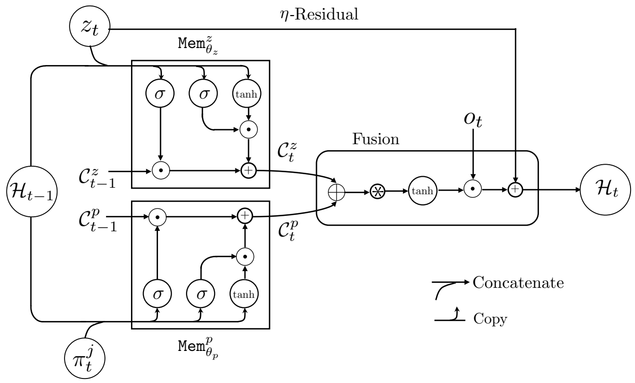


Figure 1: The proposed View-LSTM recurrent structure with one prior. The invariant feature z_t (resp. view-dependent feature π^j_t) is fed to the block $\mathtt{Mem}^z_{\theta_z}$ (resp. $\mathtt{Mem}^p_{\theta_p}$) that outputs the memory cell of Conv-LSTM [33]. The memory cells $\mathcal{C}^z_t$ and $\mathcal{C}^p_t$ of the invariant and the prior feature are then combined with a fusion scheme to obtain the hidden state $\mathcal{H}_t$ that approximates $\hat{\epsilon}^j_t$.

the view-dependent feature vector π^j as a linear combination weighted by the coefficient w_m for each of the prior feature. The component of π^j at time step t is given as:

$$\pi^j_t = \sum_{m=1}^{M} w_m \pi^j_{t,m}. \tag{3}$$

The feature π^j is a concatenation of each of π^j_t such that: $\pi^j = (\pi^j_1, \dots, \pi^j_{T'})$.

3.3. Aggregator

The original LSTM [12] model uses fully connected layer in the state transitions which cause the loss of spatial information. Conv-LSTM [33] solves this problem by keeping track of the spatiotemporal feature in its state transitions using convolutions. The equations to compute the memory cell are given as:

$$g_t = \tanh(W_{xg} * x_t + W_{hg} * \mathcal{H}_{t-1}) \tag{4a}$$
$$i_t = \sigma(W_{xi} * x_t + W_{hi} * \mathcal{H}_{t-1}) \tag{4b}$$
$$f_t = \sigma(W_{xf} * x_t + W_{hf} * \mathcal{H}_{t-1}) \tag{4c}$$
$$\mathcal{C}_t = f_t \odot \mathcal{C}_{t-1} + i_t \odot g_t, \tag{4d}$$

where the $W_{(.)}$ with subscript are learnable weights, $*$ is the convolution operation, σ is the sigmoid non-linearity and $\odot$ is the Hadamard product. The gating i_t and f_t control how much information should be kept, or updated with g_t, to compute the memory cell $\mathcal{C}_t$ for each time step t. We omit the bias term in each equation for simplicity.

The View-LSTM accepts $M+1$ inputs, the invariant feature z and the M prior features $\{\pi^j_m\}^M_{m=1}$. In order to have a spatiotemporal consistency for each of the $M+1$ inputs, we keep track of a separate memory cell $\mathcal{C}^m_t$ for each input, and it is computed through the gating mechanism for each time step t as:

$$\mathcal{C}^m_t = \begin{cases} \mathtt{Mem}^0_{\theta_0}(z_t, \mathcal{H}_{t-1}, \mathcal{C}^0_{t-1}) & \text{if } m = 0 \\ \mathtt{Mem}^m_{\theta_m}(\pi^j_{t,m}, \mathcal{H}_{t-1}, \mathcal{C}^m_{t-1}) & \text{if } m \in [1..M], \end{cases} \tag{5}$$

where $\mathtt{Mem}^m_{\theta_m}$ refers to Equation 4a to Equation 4d with the weights θ_m for the input $m \in [0..M]$. The appearance input z_t is given with index $m = 0$ and $\mathcal{C}^0_t = \mathcal{C}^z_t$. The hidden state $\mathcal{H}_{t-1}$ is used in all the $M+1$ blocks and it approximates the target feature view at the time step $t-1$. Therefore, only one hidden state should be tracked over time as it represents the actual appearance feature.

We finally fuse the information of the invariant feature memory $\mathcal{C}^z_t$ and the priors memory $\mathcal{C}^m_t$ along with the current features information as a linear combination. The output gate o_t regulates how much information each of the $M+1$ gates will be passed to the hidden state and the equation is given as:

$$o_t = \sigma\Big(W_{zo} * z_t + W^z_{co} * \mathcal{C}^z_t + \sum_{m=1}^{M} p^m\Big), \tag{6}$$

where $p^m = W_{mo} * \pi^j_{t,m} + W^m_{co} * \mathcal{C}^m_t$ which relates the current temporal prior feature $\pi^j_{t,m}$ with the memory $\mathcal{C}^m_t$ of the m-th prior. The sum of p^m over $m \in [1..M]$ approximates the spatiotemporal feature π^j_t as a linear combination of the pre-defined set of M priors in an early fusion scheme.

Finally, the hidden state $\mathcal{H}_t$ that approximate $\hat{\epsilon}^j_t$ is obtained by combining the output gate o_t with the concatenation of all the memory cells $\mathcal{C}^m_t$ and is computed as:

$$\mathcal{H}_t = o_t \odot \tanh\Big(W_{1\times 1} * \bigoplus_{m=0}^{M} \mathcal{C}^m_t\Big) + \eta z_t, \tag{7}$$

where $W_{1\times1}$ is a 2D convolution with a $(1,1)$-kernel that is used to match the dimension of o_t. We add a small residual of the input view z_t (η-Residual) controlled by η in order to alleviate the color information loss during the approximation process (see Figure 1).

4. Novel-view video synthesis

In this section we present View Decomposition Network (VDNet), our end-to-end learning framework for novel-view video synthesis based on view decomposition. We first overview the proposed network architecture and then describe how invariance is obtained. Moreover, we present our temporal extension of the perceptual loss and detail the training procedure.

4.1. Architecture

The network architecture of the proposed VDNet using one prior (*e.g.* depth or skeleton) is presented in Figure 2.

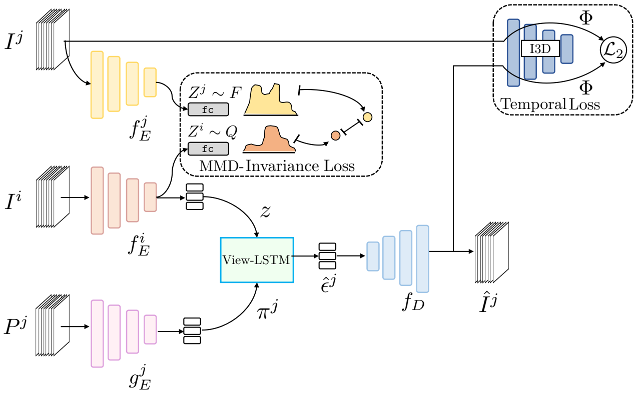


Figure 2: Proposed View Decomposition Network (VDNet) for multi-view video synthesis. Given a video sequence I^i of a view i and a target prior (or set of priors) *e.g.* depth. We synthesize the corresponding target visual sequence of the view j using the shown framework. During training, the invariance in the view decomposition assumption is enforced using Maximum Mean Discrepancy (MMD). The target appearance feature is approximated using the proposed View-LSTM recurrent structure. We maintain the temporal consistency using the proposed temporal loss which maps the synthesized frames and the ground-truth ones into a spatiotemporal feature space and computes the loss in that space.

When the number of priors $m > 1$, we only have to add an encoder $g_E^{j,m}$ for each prior P_m^j and the View-LSTM combines them as described earlier.

We are given a video $I^i \in \mathbb{R}^{T\times W\times H\times 3}$ of T frames of size $W \times H$ each from view i and the (synchronized) prior $P^j \in \mathbb{R}^{T\times W\times H\times c}$ of target view j. T, W, H, c are the number of frames, width, height, and number of channels respectively. The prior P^j can have different representations. For example if we use depth-maps as prior the number of channels is $c = 3$ (RGB like image). If the prior is a 2D-skeleton with c-keypoints for one human body, the sequence of T frames is represented as c-images of shape $\mathbb{R}^{W\times H}$. Each image is a Gaussian heatmap [18] with the center being the location of one body joint. If there are N human bodies in the scene, we sum the N heatmap sequences of each individual obtained as described above.

We pass the input I^i (resp. prior P^j) through the encoder f_E^i (resp. g_E^j) that maps it to a lower dimensional feature space obtaining z^i (resp. π^j). To enforce the invariance, we add an additional encoder f_E^j with separate weights (as described in Section 3.1) of the target view j during the training such that $z^j = f_E^j(I^j)$. After the training we set $z = f_E^i(I^i)$ since f_E^i is an invariant encoder.

Both z and π_j are passed through the proposed View-LSTM structure to approximate the target appearance feature $\hat{\epsilon}_j$. The resulting feature $\hat{\epsilon}_j$ is transformed to the image space using a decoder f_D resulting in a sequence $\hat{I}^j$.

In addition to the pixel-wise reconstruction loss $\mathcal{L}_r$, we use an invariance loss based on the Maximum Mean Discrepancy (MMD) [9] and a proposed temporal loss $\mathcal{L}_t$ that penalizes the prediction in a temporal feature space.

4.2. Invariance loss

The Maximum Mean Discrepancy (MMD) [9] is used in domain adaptation [22, 29, 53] to learn an invariant representation of objects of the same concept from different domain (sources), while preserving a meaningful representation of the data. We propose to use MMD to learn the invariance mapping between the input view i and the target view j.

Let $\hat{Z}^i = \{z_b^i\}_{b=1}^B$ (resp. $\hat{Z}^j = \{z_b^j\}_{b=1}^B$) be the empirical batch of size B of the feature of the input (resp. target) view obtained using the encoder f_E^i (resp. f_E^j). MMD compares two distributions F and Q and maps the data to a reproducing kernel Hilbert space (RKHS) using a feature mapping ϕ. The invariance loss $\mathcal{L}_n$ to train our network is given as MMD defined as:

$$\mathcal{L}_n = \mathrm{MMD}(\hat{Z}^i, \hat{Z}^j) = \left\| \frac{1}{B} \sum_{i=1}^{B} \phi(z_b^i) - \phi(z_b^j) \right\|. \quad (8)$$

From the statistical test provided in [9] we have that $F = Q$ if and only if $\mathrm{MMD}(\hat{Z}^i, \hat{Z}^j) = 0$. The characteristic kernel k associated with the feature map ϕ is given as: $k(.,.) = \langle \phi(.), \phi(.) \rangle$. We choose the commonly used Radial basis function (RBF) kernel defined as $k(\mathbf{x}, \mathbf{x}') = \exp(-\frac{1}{2\sigma^2}\|\mathbf{x} - \mathbf{x}'\|^2)$ with $\sigma = 0.2$ as default value in our experiments.

4.3. Temporal perceptual loss

The perceptual loss [14] for image-based generative models operates in the feature space of a pre-trained 2D-CNN (perceptual network Φ). Current video generative models such as [17, 20] use an averaging over the predicted frames, but by doing so they summarize too coarsely the temporal aspect of the problem. Another problem is the loss of the temporal ordering. Therefore, we penalize the prediction using spatiotemporal perceptual network Φ such as 3D-CNN. We use a pre-trained perceptual network and freeze the weights so that it only serves as a feature mapping. The other reason is that we do not want to have the perceptual network to be specific to the novel-view video synthesis task. The Temporal perceptual loss function $\mathcal{L}_t$ is therefore:

$$\mathcal{L}_t = \sum_{l=1}^{L} \frac{\lambda_l}{T_l W_l H_l C_l} \left\| \Phi_l(\hat{I}^j) - \Phi_l(I^j) \right\|_2, \quad (9)$$

where λ_l is a coefficient, Φ_l is the l-th feature map from the perceptual network of shape $T_l \times W_l \times H_l \times C_l$ with T_l,

W_l, H_l and C_l being the time frame length, width, height and number of channels of the feature map. L is the number of chosen layers in the network.

4.4. Training

We use a 6-layered 3D ResNet [13] as a backbone architecture for VDNet. Note that other feature mapping encoders (e.g. VGG [34]) are outside the scope of this study. The final loss, $\mathcal{L}$, we employ to train the model is:

$$\mathcal{L} = \mathcal{L}_r + \lambda_t \mathcal{L}_t + \lambda_n \mathcal{L}_n, \tag{10}$$

where $\mathcal{L}_r$ is the pixel-wise reconstruction loss between the model synthesized sequence $\hat{I}^j$ and the ground-truth target I^j; $\mathcal{L}_t$ is the temporal perceptual loss (see Eq. 9); and $\mathcal{L}_n$ is the invariance loss (see Eq. 8). We empirically set the regularizers as $\lambda_t = 10^{-2}$ and $\lambda_n = 10^{-3}$. As 3D perceptual network we choose I3D [3], an action recognition model trained on the Kinetics Dataset [3]. In $\mathcal{L}_t$ we use the following four feature maps to obtain diverse spatiotemporal feature representations generated by the perceptual network: Conv3d_1a_7×7, Conv3d_2b_1×1, Mixed_3c, and Mixed_4b. We set the coefficients λ_l to 1 in all the four layers used.

5. Experiments

In this section we validate the proposed View-LSTM recurrent structure, and the View Decomposition Network (VDNet). We first provide the experimental setup with the baselines and evaluation metrics. We then motivate the choice of using Conv-LSTM as a baseline for our proposed View-LSTM, and we compare it with other aggregation schemes. Finally, we provide an extensive set of experiments to validate the proposed VDNet.

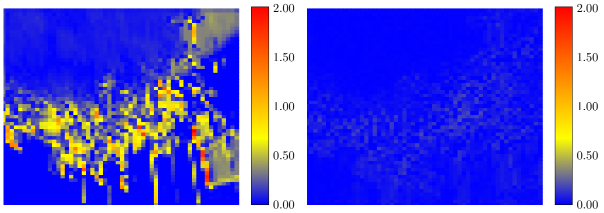
5.1. Setup

There are no direct state-of-the-art methods that tackle the problem of novel-view video synthesis. We therefore compare our proposed VDNet with one video-based baseline (**ResNet** [13]) and two frame-based methods (**PG**^{2} [27], and **VDG** [18]) that solve the pose guided human synthesis problem. The **ResNet** model is adapted from the paper [13] we replace the 2D convolution kernels with 3D kernels. The model serves as a baseline for video-based model. We train the model with $\lambda_t = 10^{-1}$ and $\lambda_n = 10^{-3}$ for all the experiments. For both **PG**^{2} and **VDG** we replace the pose with depth and train the model with the same hyperparameters as defined in the paper. Additionally, for **VDG** we replace the fully connected target branch with the input encoder structure.

To evaluate the performance of the generators, we use Structural Similarity (SSIM) [50] score as a per-frame quantitative measure and Fréchet Video Distance (FVD) [40] to measure the video quality and we use I3D [3] to extract the embeddings.

Layer	**Conv-LSTM**	$\mathcal{L}_2(v^1, v^2)$	$\mathcal{L}_2(v^1, v^3)$	$\mathcal{L}_2(v^2, v^3)$
Conv$_1$	✗	$.152 \pm .072$	$.149 \pm .072$	$.173 \pm .088$
	✓	$\mathbf{.027 \pm .020}$	$\mathbf{.026 \pm .020}$	$\mathbf{.029 \pm .023}$
Conv$_2$	✗	$.273 \pm .081$	$.269 \pm .081$	$.297 \pm .088$
	✓	$\mathbf{.056 \pm .019}$	$\mathbf{.055 \pm .019}$	$\mathbf{.060 \pm .020}$

Table 1: Effect of Conv-LSTM in l-Net model on the feature map invariance between different views. KEY – v^i: view i for $i \in \{1, 2, 3\}$.



(a) l-Net w/o Conv-LSTM (b) l-Net

Figure 3: Visual result of the $\mathcal{L}_1$ difference between the third channel of the first convolution layer of l-Net w/o (and with) Conv-LSTM between view 1 and view 2.

For all our experiments we use the NTU RGB+D [32] dataset. We chose it because currently it is the only large scale multi-view multi-modal synchronized dataset. The dataset has 60 action classes performed by 40 participants. Three cameras are used at the same height and different horizontal angles during each recording. We use the cross-subject split which is divided into train and test split with $40,320$ and $16,560$ samples.

We train all the models with a batch size of 6 and Adam optimizer [15] with $(\beta_1, \beta_2) = (0.5, 0.999)$ and a learning rate of 2.10^{-5}. The size of the frames is fixed to 112×112 for all the experiments.

The proposed VDNet model is implemented using the PyTorch framework. All the experiments were carried out using a server equipped with Tesla V100 GPU.

5.2. Conv-LSTM as invariance baseline

We start with a 3D ResNet [10] model for action recognition. We replace all the (k, k, k) convolution kernels in the network with the kernel $(1, k, k)$ which means that we are only convolving over the spatial dimension. We refer to this model as ”Net”. We present four variants of the Net model. *f-Net w/o C-LSTM* where we add a fully connected layer before the classification layer. *l-Net w/o C-LSTM* replaces the fully connected with an LSTM layer. We add a Conv-LSTM after each convolution block for each of the two models and name them *f-Net* and *l-Net* respectively.

After training the four models on action recognition, we evaluate the invariance property of the Conv-LSTM as a retrieval task. We select a query frame index from an input view and we perform a Rank$_k$ over a target view. The fea-

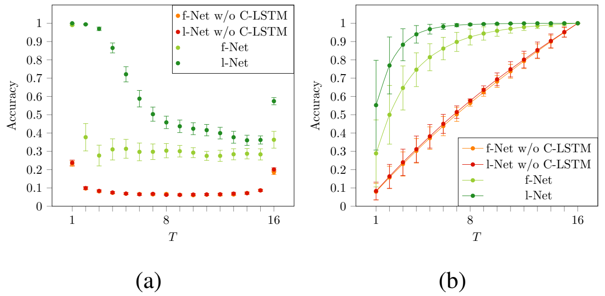


Figure 4: Multi-view retrieval scores: (a) average over all the views; (b) retrieval from view 3 to view 2.

Model	Operator ψ	↑ SSIM	↓ FVD
ResNet	Concatenation	.614	13.19
	Early-fusion	.496	22.08
	Late-fusion	.612	22.59
VDNet	View-LSTM ($\eta = 0$)	.308	23.64
	View-LSTM ($\eta = 10^{-1}$)	.623	11.30
	View-LSTM ($\eta = 10^{-2}$)	**.710**	**9.35**
	View-LSTM ($\eta = 10^{-3}$)	.295	47.97

Table 2: Performance comparison of our proposed View-LSTM used in VDNet against early and late fusion.

#layers	6 (3D)	6	18	20	34
#params	85.06M	34.70M	77.20M	84.29M	133.87M
SSIM ($t = 8$)	**.783**	.698	.711	.609	.663

Table 3: 2D backbone ResNet for the proposed VDNet.

tures are obtained from the average pooling layer of each model.

Figure 4a shows the average retrieval score between all the views. We clearly see the advantage of using Conv-LSTM. Using LSTM for global temporal learning also helps. Because most of the motion happens after the first few frames, the retrieval after the 5-th frame is below 0.7 accuracy. Figure 4b reports the top-k retrieval between view 2 and view 3. For models that are not using Conv-LSTM the graph is almost linear with respect to the frame index.

Table 1 compares the invariance score in the feature space as an average $\mathcal{L}_2$ error over all the channels for the first two convolutional blocks. We report good invariance scores on l-Net which suggest that the Conv-LSTM learns better invariant representation. From the difference map presented in Figure 3 we can see that Conv-LSTM is invariant to the viewpoint.

5.3. View-LSTM

We fix η as defined in Equation 7 to 10^{-2} in all the experiments. Table 2 shows the sensitivity results of the η-residual. We note that when the residual is too small ($\eta = 10^{-3}$) or not used as in standard methods ($\eta = 0$) the model performs poorly. The best score was obtained with $\eta = 10^{-2}$. With $\eta = 10^{-1}$ still preforming better than $\eta = 10^{-3}$. Figure 6 shows one example of the synthesized first frame by varying η. For $\eta \in \{0, 10^{-3}\}$ View-LSTM was able to get the structure of the bodies but the color information could not be recovered. For $\eta \in \{10^{-1}, 10^{-2}\}$ the

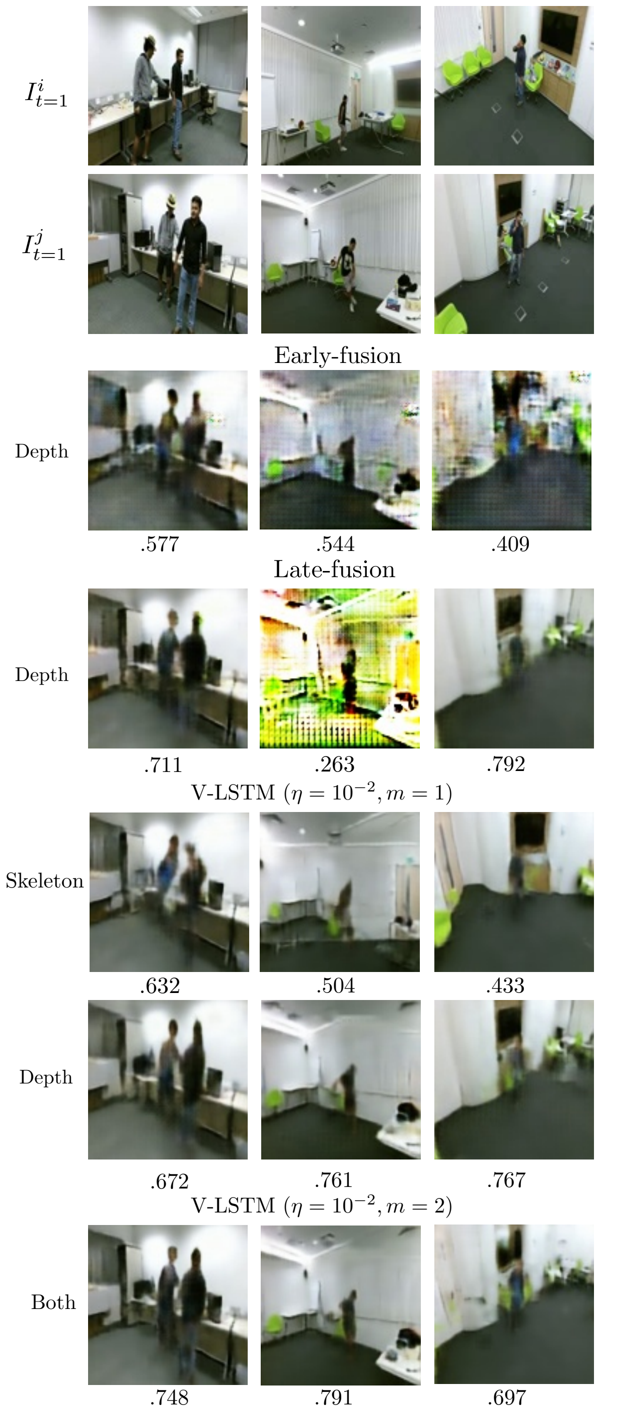


Figure 5: Synthesized first frame (SSIM score on the bottom) with early, late fusion and different configurations of View-LSTM.

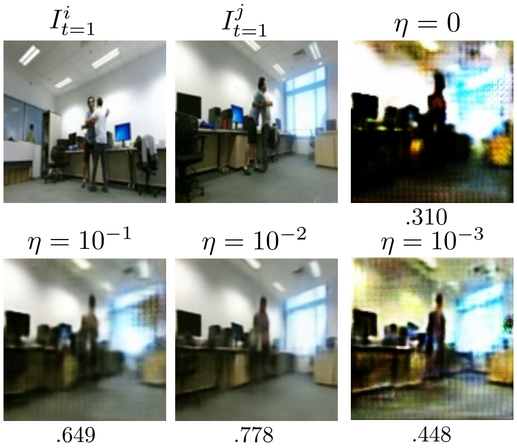


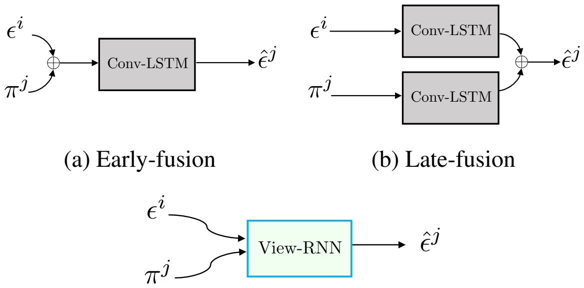
Figure 6: Synthesized first frame (SSIM score on the bottom) with different η values in View-LSTM using depth as prior.

colors were recovered with the residual. The 0.649 SSIM for $\eta = 10^{-1}$ is justified by the fact that View-LSTM may be influenced more by the input than in $\eta = 10^{-2}$.

Table 2 compares View-LSTM with the early and late fusion. In early fusion both the input feature and the prior feature are first concatenated and fed to a single Conv-LSTM. In late fusion both the input and target prior feature have separate Conv-LSTM and the last hidden state of each recurrent model are concatenated together (see Figure 7). The improvement with the three methods is almost linear with SSIM. View-LSTM shows advantage compared to standard fusion schemes. This is because the recurrent structure is built to approximate the target feature directly in its output gate. The late fusion outperforms the early fusion. Note that using only concatenation as fusion scheme works better than early and late fusion. Figure 5 shows three examples of depth with early, late fusion, and our View-LSTM with $\eta = 10^{-2}$. We notice some irregularities in the generation for early and later fusion (i.e., frequent flickering images) whereas our View-LSTM produces regular images.

5.4. VDNet

Table 4 shows the SSIM scores with the baselines using $T = 8$ time steps. Video based models surpass frame based models by a noticeable margin. Using only the pixel-wise reconstruction loss VDNet got $.710 \pm .076$ against $.614 \pm .144$ for ResNet. This is because View-LSTM inherits the implicit invariance property from the Conv-LSTM as shown in Section 5.2. On both ResNet and VDNet we clearly see the advantage of the invariance loss term $(\mathcal{L}_n, \mathcal{L}_{n-1})$ compared to only using the reconstruction term $\mathcal{L}_r$. MMD invariance loss $\mathcal{L}_n$ shows good performance in our VDNet model. The proposed temporal loss $\mathcal{L}_t$ improves over the perceptual loss $\mathcal{L}_p$. Finally, combining all the losses our VDNet gets the best performance. Figure 8a shows one example of all the models. On both PG^2 and VDG the models were not able to synthesize the target view which shows the effectiveness of the 3D models compared to 2D ones. We further run additional experiments with the 2D CNN version of our proposed VDNet (see Table 3) with different layer depths. Results confirm our claim that the spatiotemporal features considerably help video synthesis. ResNet model shows good reconstruction of the body and background but has visible artifacts which cause the SSIM score of about .50 on each of the frames. Our model was able to synthesize well the body and the background with a good use of the input color information using the η-residual in v-LSTM.



(a) Early-fusion

(b) Late-fusion

(c) View-LSTM

Figure 7: Aggregator ψ scheme to approximate $\hat{\epsilon}^j$: (a) both features are concatenated and fed to a Conv-LSTM; (b) each feature has a separate Conv-LSTM and then concatenated together; (c) the two features are combine using the proposed View-LSTM structure.

Figure 8b shows the FVD distance as a boxplot. Except for the poor invariance loss $\mathcal{L}_n$ performance of VDNet, we can see the advantage of VDNet compared to ResNet.

We ran additional experiments to test models behaviour on longer video sequences. Figure 8c shows the per-frame average SSIM scores of the four models presented above. The decrease of the ResNet and VDNet from $T = 8$ to $T = 16$ is almost linear. For $T = 24$ the ResNet model has similar score with $T = 16$ whereas VDNet was heavily affected by the temporal length of $T = 24$.

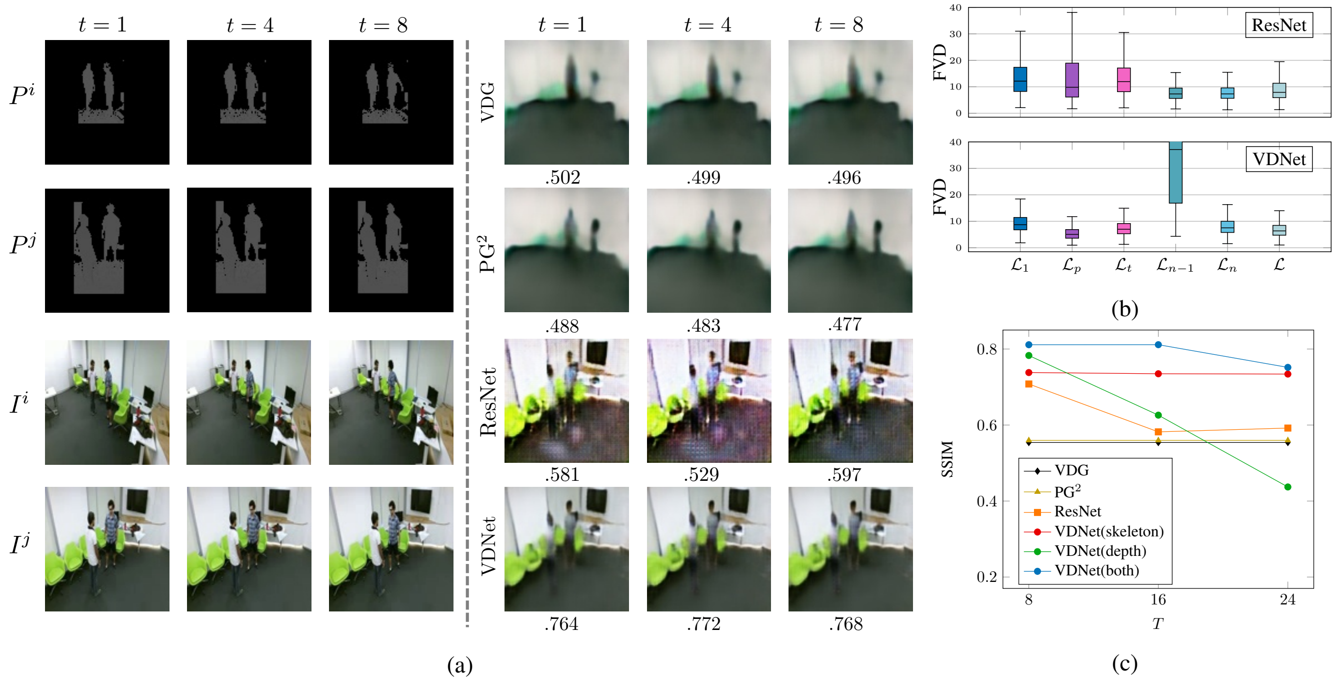
Frame based models are not affected by the time length. This is expected since the frame ignores the time axis and treats each frame separately regardless of the time step. However, when looking at the generated image sequence in Figure 8a we see that most of the score comes from the background rather than the person.

We ran two more experiments using the skeleton prior and by combining both the depth and skeleton.

Using the skeleton gives a constant SSIM score on longer sequence (Figure 8c) compared to the depth prior. However, as can be seen in Figure 5 the VDNet model could not synthesis well the full body and the limbs. Combining both

Model	Losses				Pair-view SSIM score						Average
	$\mathcal{L}_r$	$\mathcal{L}_t$	$\mathcal{L}_n$	$\mathcal{L}_g$	$v^1 \to v^2$	$v^1 \to v^3$	$v^2 \to v^1$	$v^2 \to v^3$	$v^3 \to v^1$	$v^3 \to v^2$	
VDG [18]	✓			✓	.502 ± .058	.543 ± .068	.584 ± .060	.563 ± .062	.611 ± .077	.522 ± .063	.554 ± .075
PG^2 [27]	✓			✓	.499 ± .071	.561 ± .060	.600 ± .064	.557 ± .071	.598 ± .075	.543 ± .066	.560 ± .076
ResNet	✓				.594 ± .154	.603 ± .153	.641 ± .143	.620 ± .135	.636 ± .130	.593 ± .138	.614 ± .144
	✓	$\mathcal{L}_p$			.591 ± .131	.580 ± .139	.562 ± .143	.538 ± .144	.646 ± .159	.612 ± .142	.588 ± .148
	✓	✓			.601 ± .167	.584 ± .160	.644 ± .120	.619 ± .120	.645 ± .127	.623 ± .136	.619 ± .141
	✓		$\mathcal{L}_{n-1}$		.773 ± .078	.767 ± .083	.789 ± .061	.721 ± .097	.782 ± .084	.746 ± .088	.763 ± .086
	✓		✓		.776 ± .072	.757 ± .089	.785 ± .071	.735 ± .089	.788 ± .067	.739 ± .087	.764 ± .082
	✓	✓	✓		.705 ± .115	.735 ± .095	.717 ± .130	.690 ± .122	.734 ± .127	.669 ± .150	.708 ± .127
VDNet	✓				.721 ± .069	.717 ± .068	.735 ± .067	.676 ± .083	.728 ± .069	.685 ± .079	.710 ± .076
	✓	$\mathcal{L}_p$			.753 ± .087	.769 ± .062	.775 ± .072	.734 ± .083	.789 ± .055	.700 ± .118	.753 ± .087
	✓	✓			.768 ± .076	.772 ± .069	.773 ± .082	.752 ± .068	.772 ± .071	.737 ± .082	.762 ± .076
	✓		$\mathcal{L}_{n-1}$		.346 ± .221	.354 ± .217	.511 ± .233	.478 ± .215	.570 ± .263	.541 ± .249	.467 ± .249
	✓		✓		.762 ± .071	.763 ± .070	.767 ± .081	.737 ± .074	.769 ± .079	.737 ± .077	.756 ± .077
	✓	✓	✓		**.789 ± .076**	**.791 ± .069**	**.800 ± .076**	**.765 ± .079**	**.797 ± .067**	**.756 ± .089**	**.783 ± .078**

Table 4: SSIM scores with $T = 8$. We report the scores of all the combinations of the three views and the average score on each model. As ablation study, we replace $\mathcal{L}_t$ (resp. $\mathcal{L}_n$) with $\mathcal{L}_p$ (resp. $\mathcal{L}_{n-1}$) KEY – $\mathcal{L}_r$: pixel-wise reconstruction loss, $\mathcal{L}_g$: adversarial loss, $\mathcal{L}_{n-1}$: replacing MMD with $\mathcal{L}_1$ term, $\mathcal{L}_n$: MMD invariance loss, $\mathcal{L}_p$: perceptual loss, $\mathcal{L}_t$: proposed temporal loss, v^i: view i for $i \in \{1, 2, 3\}$.



(a)

(b)

(c)

Figure 8: Comparison of our proposed VDNet with state-of-the-art methods using $T = 8$: (a) synthesized frames (SSIM score on bottom) where the input pose P^i is only used in VDG and PG^2; (b) FVD score with different losses; (c) SSIM score when varying the time step in the synthesis.

priors (depth and skeleton) shows improvement especially for $T = 24$ compared to using only depth (Figure 8c).

6. Conclusion

We proposed to solve the novel-view video synthesis problem by decomposing a view into an invariant representation, which is shared across all views of the same scene, and a view-dependent representation, which is specific to the selected viewpoint. We implemented this decomposition by extending the Conv-LSTM recurrent structure to approximate the target feature vector. We used the proposed View-LSTM in an end-to-end generator, VDNet, and tested it against state-of-the-art models. The experimental results showed the effectiveness of the proposed architecture and validate View-LSTM in handling multiple types of priors.

Acknowledgements This project acknowledges the use of the ESPRC funded Tier 2 facility, JADE.

References

[1] Thiemo Alldieck, Marcus Magnor, Bharat Lal Bhatnagar, Christian Theobalt, and Gerard Pons-Moll. Learning to reconstruct people in clothing from a single RGB camera. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[2] Haoye Cai, Chunyan Bai, Yu-Wing Tai, and Chi-Keung Tang. Deep video generation, prediction and completion of human action sequences. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *Computer Vision – ECCV 2018*, pages 374–390, Cham, 2018. Springer International Publishing.

[3] J. Carreira and A. Zisserman. Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4724–4733, July 2017.

[4] S. Chopra, R. Hadsell, and Y. LeCun. Learning a similarity metric discriminatively, with application to face verification. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 539–546 vol. 1, June 2005.

[5] Hang Gao, Huazhe Xu, Qi-Zhi Cai, Ruth Wang, Fisher Yu, and Trevor Darrell. Disentangling propagation and generation for video prediction. In *CoRR*, volume abs/1812.00452, 2018.

[6] Yixiao Ge, Zhuowan Li, Haiyu Zhao, Guojun Yin, Shuai Yi, Xiaogang Wang, and hongsheng Li. FD-GAN: Pose-guided feature distilling GAN for robust person re-identification. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 1222–1233. Curran Associates, Inc., 2018.

[7] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative Adversarial Nets. In *Advances in Neural Information Processing Systems*, December 2014.

[8] Alex Graves and Jürgen Schmidhuber. Framewise phoneme classification with bidirectional LSTM networks. In *IEEE International Joint Conference on Neural Networks*, July 2005.

[9] Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *Journal of Machine Learning Research (JMLR)*, 13:723–773, Mar. 2012.

[10] K. Hara, H. Kataoka, and Y. Satoh. Can spatiotemporal 3D CNNs retrace the history of 2D CNNs and Imagenet? In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6546–6555, June 2018.

[11] Jiawei He, Andreas Lehrmann, Joseph Marino, Greg Mori, and Leonid Sigal. Probabilistic video generation using holistic attribute control. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *Computer Vision – ECCV 2018*, pages 466–483, Cham, 2018. Springer International Publishing.

[12] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, Nov. 1997.

[13] P. Isola, J. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5967–5976, July 2017.

[14] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision – ECCV 2016*, pages 694–711, Cham, 2016. Springer International Publishing.

[15] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, May 2015.

[16] Manoj Kumar, Mohammad Babaeizadeh, Dumitru Erhan, Chelsea Finn, Sergey Levine, Laurent Dinh, and Durk Kingma. VideoFlow: A Flow-Based Generative Model for Video. In *CoRR*, volume abs/1903.01434, 2019.

[17] Wei-Sheng Lai, Jia-Bin Huang, Oliver Wang, Eli Shechtman, Ersin Yumer, and Ming-Hsuan Yang. Learning blind video temporal consistency. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *Computer Vision – ECCV 2018*, pages 179–195, Cham, 2018. Springer International Publishing.

[18] Mohamed Ilyes Lakhal, Oswald Lanz, and Andrea Cavallaro. Pose guided human image synthesis by view disentanglement and enhanced weighting loss. In Laura Leal-Taixé and Stefan Roth, editors, *Computer Vision – ECCV 2018 Workshops*, pages 380–394, Cham, 2019. Springer International Publishing.

[19] Alex X. Lee, Richard Zhang, Frederik Ebert, Pieter Abbeel, Chelsea Finn, and Sergey Levine. Stochastic adversarial video prediction. volume abs/1804.01523, 2018.

[20] Donghoon Lee, Tomas Pfister, and Ming-Hsuan Yang. Inserting videos into videos. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[21] Hsin-Ying Lee, Hung-Yu Tseng, Jia-Bin Huang, Maneesh Singh, and Ming-Hsuan Yang. Diverse image-to-image translation via disentangled representations. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *Computer Vision – ECCV 2018*, pages 36–52, Cham, 2018. Springer International Publishing.

[22] H. Li, S. J. Pan, S. Wang, and A. C. Kot. Domain generalization with adversarial feature learning. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5400–5409, June 2018.

[23] Junnan Li, Yongkang Wong, Qi Zhao, and Mohan Kankanhalli. Unsupervised learning of view-invariant action representations. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 1254–1264. Curran Associates, Inc., 2018.

[24] Yijun Li, Chen Fang, Jimei Yang, Zhaowen Wang, Xin Lu, and Ming-Hsuan Yang. Flow-grounded spatial-temporal video prediction from still images. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *Computer Vision – ECCV 2018*, pages 609–625, Cham, 2018. Springer International Publishing.

[25] B. Liu, X. Wang, M. Dixit, R. Kwitt, and N. Vasconcelos. Feature space transfer for data augmentation. In

2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 9090–9098, June 2018.

[26] Stephen Lombardi, Jason Saragih, Tomas Simon, and Yaser Sheikh. Deep appearance models for face rendering. *ACM Trans. Graph.*, 37(4):68:1–68:13, July 2018.

[27] Liqian Ma, Xu Jia, Qianru Sun, Bernt Schiele, Tinne Tuytelaars, and Luc Van Gool. Pose guided person image generation. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 406–416. Curran Associates, Inc., 2017.

[28] Helge Rhodin, Victor Constantin, Isinsu Katircioglu, Mathieu Salzmann, and Pascal Fua. Neural scene decomposition for multi-person motion capture. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[29] A. Rozantsev, M. Salzmann, and P. Fua. Beyond sharing weights for deep domain adaptation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(4):801–814, April 2019.

[30] Masaki Saito, Eiichi Matsumoto, and Shunta Saito. Temporal generative adversarial nets with singular value clipping. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2017.

[31] Christian Schuldt, Ivan Laptev, and Barbara Caputo. Recognizing Human Actions: A Local SVM Approach. In *Proceedings of the Pattern Recognition, 17th International Conference on (ICPR) Volume 3 - Volume 03*, ICPR '04, pages 32–36, Washington, DC, USA, 2004. IEEE Computer Society.

[32] Amir Shahroudy, Jun Liu, Tian-Tsong Ng, and Gang Wang. NTU RGB+D: A large scale dataset for 3D human activity analysis. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[33] Xingjian SHI, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-kin Wong, and Wang-chun WOO. Convolutional LSTM Network: A machine learning approach for precipitation nowcasting. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 802–810. Curran Associates, Inc., 2015.

[34] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations (ICLR)*, May 2015.

[35] Nitish Srivastava, Elman Mansimov, and Ruslan Salakhutdinov. Unsupervised learning of video representations using LSTMs. In *Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, pages 843–852. JMLR.org, 2015.

[36] Swathikiran Sudhakaran, Sergio Escalera, and Oswald Lanz. LSTA: long short-term attention for egocentric action recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[37] Lin Sun, Kui Jia, Yuejia Shen, Silvio Savarese, Dit-Yan Yeung, and Bertram E. Shi. Coupled Recurrent Network (CRN). In *CoRR*, volume abs/1812.10071, 2018.

[38] Ximeng Sun, Huijuan Xu, and Kate Saenko. A two-stream variational adversarial network for video generation. In *CoRR*, volume abs/1812.01037, 2018.

[39] Sergey Tulyakov, Ming-Yu Liu, Xiaodong Yang, and Jan Kautz. MoCoGAN: Decomposing motion and content for video generation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[40] Thomas Unterthiner, Sjoerd van Steenkiste, Karol Kurach, Raphael Marinier, Marcin Michalski, and Sylvain Gelly. Towards accurate generative models of video: A new metric & challenges. In *CoRR*, volume abs/1812.01717, 2018.

[41] Ruben Villegas, Jimei Yang, Seunghoon Hong, Xunyu Lin, and Honglak Lee. Decomposing motion and content for natural video sequence prediction. In *International Conference on Learning Representations (ICLR)*, 2017.

[42] Ruben Villegas, Jimei Yang, Yuliang Zou, Sungryull Sohn, Xunyu Lin, and Honglak Lee. Learning to generate long-term future via hierarchical prediction. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, pages 3560–3569. JMLR.org, 2017.

[43] Timo von Marcard, Roberto Henschel, Michael J. Black, Bodo Rosenhahn, and Gerard Pons-Moll. Recovering accurate 3D human pose in the wild using IMUs and a moving camera. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *Computer Vision – ECCV 2018*, pages 614–631, Cham, 2018. Springer International Publishing.

[44] Shruti Vyas, Yogesh Singh Rawat, and Mubarak Shah. Time-aware and view-aware video rendering for unsupervised representation learning. In *CoRR*, volume abs/1811.10699, 2018.

[45] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Nikolai Yakovenko, Andrew Tao, Jan Kautz, and Bryan Catanzaro. Video-to-video synthesis. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 1144–1156. Curran Associates, Inc., 2018.

[46] Yunbo Wang, Zhifeng Gao, Mingsheng Long, Jianmin Wang, and Philip S Yu. PredRNN++: Towards a resolution of the deep-in-time dilemma in spatiotemporal predictive learning. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 5123–5132, Stockholmsmssan, Stockholm Sweden, 10–15 Jul 2018. PMLR.

[47] Yunbo Wang, Lu Jiang, Ming-Hsuan Yang, Li-Jia Li, Mingsheng Long, and Li Fei-Fei. Eidetic 3D LSTM: A model for video prediction and beyond. In *International Conference on Learning Representations (ICLR)*, May 2019.

[48] Yunbo Wang, Mingsheng Long, Jianmin Wang, Zhifeng Gao, and Philip S Yu. PredRNN: Recurrent neural networks for predictive learning using spatiotemporal LSTMs. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 879–888. Curran Associates, Inc., 2017.

[49] Yunbo Wang, Jianjin Zhang, Hongyu Zhu, Mingsheng Long, Jianmin Wang, and Philip S. Yu. Memory in memory:

A predictive neural network for learning higher-order non-stationarity from spatiotemporal dynamics. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[50] Zhou Wang, Alan Conrad Bovik, Hamid Rahim Sheikh, and Eero P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing (TIP)*, April 2004.

[51] Nevan Wichers, Ruben Villegas, Dumitru Erhan, and Honglak Lee. Hierarchical long-term video prediction without supervision. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, pages 6033–6041, 2018.

[52] C. Xu, D. Tao, and C. Xu. Multi-view intact space learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(12):2531–2544, Dec 2015.

[53] H. Yan, Y. Ding, P. Li, Q. Wang, Y. Xu, and W. Zuo. Mind the class weight bias: Weighted maximum mean discrepancy for unsupervised domain adaptation. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 945–954, July 2017.

[54] Ceyuan Yang, Zhe Wang, Xinge Zhu, Chen Huang, Jianping Shi, and Dahua Lin. Pose guided human video generation. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *Computer Vision – ECCV 2018*, pages 204–219, Cham, 2018. Springer International Publishing.

[55] Long Zhao, Xi Peng, Yu Tian, Mubbasir Kapadia, and Dimitris Metaxas. Learning to forecast and refine residual motion for image-to-video generation. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *Computer Vision – ECCV 2018*, pages 403–419, Cham, 2018. Springer International Publishing.