A Manifold Learning based Video Prediction approach for Deep Motion Transfer

Yuliang Cai
Dept. of ECE, UCSD
ycai@ucsd.edu

Sumit Mohan
Intel Corporation
sumit.mohan@intel.com

Adithya Niranjan
Intel Labs
BITS Pilani, Goa
f20160444g@alumni.bits-pilani.ac.in

Nilesh Jain
Intel Labs
nilesh.jain@intel.com

Alex Cloninger
Dept. of Mathematics, UCSD
Halıcıoğlu Data Science Institute, UCSD
acloninger@ucsd.edu

Srinjoy Das
School of Mathematical and Data Sciences, WVU
srinjoy.das@mail.wvu.edu

Abstract

We propose a novel manifold learning based end-to-end prediction and video synthesis framework for bandwidth reduction in motion transfer enabled applications such as video conferencing. In our workflow we use keypoint based representations of video frames where image and motion specific information are encoded in a completely unsupervised manner. Prediction of future keypoints is then performed using the manifold of a variational recurrent neural network (VRNN) following which output video frames are synthesized using an optical flow estimator and a conditional image generator in the motion transfer pipeline. The proposed architecture which combines keypoint based representation of video frames with manifold learning based prediction enables significant additional bandwidth savings over motion transfer based video conferencing systems which are implemented solely using keypoint detection. We demonstrate the superiority of our technique using two representative datasets for both video reconstruction and transfer and show that prediction using VRNN has superior performance as compared to a non manifold based technique such as RNN.

1. Introduction

Global trends such as an increasing number of people working from home or other remote locations, and with project teams being dispersed across different geographic locations, this has led to a proliferation of applications such as live streaming and video conferencing. Motion Transfer based video synthesis is an important technique that has been proposed to efficiently implement such applications with the goal of maintaining high video quality as well as achieve a significant degree of compression for bandwidth efficiency [24]. Previously proposed motion transfer frameworks use keypoint based representations for the source image and the driving video which are learnt in an unsupervised manner [12] and then processed through an optical flow estimation network and a conditional image generator [20]. In this paper we construct an end-to-end motion transfer pipeline using prediction from the manifold of a Variational Recurrent Neural Network [4] in conjunction with Deep Learning (DL) models for video synthesis. This framework enables improving the performance, latency and quality of service across mobile client devices through the content distribution network as well as in data centers as listed below:

- Enhancing a motion transfer pipeline using prediction for the keypoints enables the mobile or PC client devices to reduce the amount of data that needs to be sent over to the end device.
- It further allows the client device to reduce its compute requirements, as the client does not need to capture or process every single frame of video. This improves power and performance of the application on the client device.
- Over the network the prediction helps to further reduce the data bandwidth needed for the application, and hence enable higher latency tolerance for applications like video conferencing.

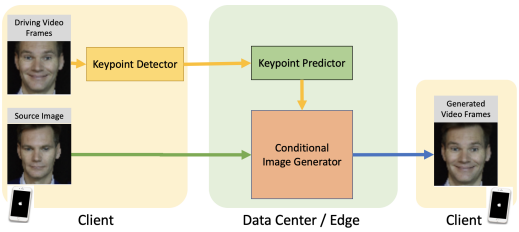

Figure 1. End to End Prediction and Video Synthesis Architecture for Motion Transfer

During inference the higher compute nodes of CPU and GPU in the data center or edge can be used to enable DL based models for prediction and image synthesis, along with other video and image analytics techniques to improve the video and provide a higher quality stream to the end device. A drawing of this scheme is shown in Figure 1.

2. Manifold based prediction

Using keypoints extracted from videos for prediction as compared to directly using pixels from the input video frames have advantages in terms of error accumulation [16]. In addition keypoint based representations are more suitable for downstream tasks in a motion transfer workflow such as optical flow estimation and therefore can be used for prediction instead of raw pixels. Keypoint based representations obtained from video frames constitute high dimensional time series. There are several options available for forecasting of such data including models such as Vector Autoregressive Models (VAR) [3] or Recurrent Neural Networks (RNNs) [15]. However in the context of such problems the manifold hypothesis states that high dimensional data $x \subseteq M_X$ lie on an underlying low dimensional manifold $z \subseteq M_z$ [9, 17, 18]. Therefore for time series in high dimensional space such as keypoints obtained from video frames a prediction framework can be constructed in latent space z by performing the steps as below:

- Map X_t where $t = 1, 2, \ldots, n$ in the original high dimensional manifold M_X to z_t lying on the low dimensional manifold M_z by a mapping parameterized by ω.

$$\psi_\omega : \; X_t \rightarrow z_t$$

- Perform prediction in the low dimensional space M_z to generate z_{n+k} where $k \geq 1$
- Generate the predicted value of X by the inverse mapping:

$$\psi_\omega^{-1} : \; z_{n+k} \rightarrow X_{n+k}$$

Two principal approaches for prediction of time series on manifolds are outlined as below:

- The mapping ψ_ω can be obtained using nonlinear dimensionality reduction techniques such as Diffusion Maps [6] or Laplacian eigenmaps [2]. In such cases the data X_t is transformed to the eigenvectors z_t of its associated graph Laplacian matrix. However such methods are unsuitable for handling large datasets and also fail to generalize to out-of-sample data. Deep Learning based approaches in conjunction with nonlinear dimensionality reduction techniques such as Diffusion Maps for obtaining the mappings $\psi_\omega, \psi_\omega^{-1}$ have been proposed to overcome these problems [14, 19]. However handling time series prediction using such approaches is still problematic as the obtained low dimensional embedding z does not specifically take into account the time order of the measurements. [21, 22]. In addition generating the map ψ_ω^{-1} is often difficult and not designed for streaming data or high dimensional time series [1, 5].
- The mapping ψ_ω is obtained using Deep Learning based architectures such as Autoencoders [25] or Variational Autoencoders [13]. In such cases the transformations $\psi_\omega, \psi_\omega^{-1}$ are constructed using the training data and the latent space z_t can be used for prediction. Details are described in the next section.

3. Prediction using Variational Autoencoder

A Variational Autoencoder (VAE) is a Deep Generative Model which learns a smooth mapping from the modeled distribution of the data X lying on a high dimensional manifold M_X to a user-defined prior distribution z lying on a manifold M_z. For our problem, keypoints X_t, X_{t+k} are derived from successive video frames at times $t, t+k$ where $k \geq 1$. Two key properties of the mapping of the input space X_t to the latent space z_t which enable prediction are outlined as below:

- P1: Every point X_t in the input space where $t = 1, 2, \ldots, n$ can be mapped to a point z_t in the latent space
- P2: Points which are close in X are also close in z. This is owing to the Lipschitz continuity property of neural networks with respect to their inputs [10] which also hold for the VAE encoder and is mathematically described as below:
  Given condition on keypoints: $||X_t - X_{t+k}|| < \epsilon$ it follows that:

$$||f_\theta(X_t) - f_\theta(X_{t+k})|| < L||X_t - X_{t+k}||$$
$$\text{i.e. } ||f_\theta(X_t) - f_\theta(X_{t+k})|| < L\epsilon$$

  Here L is the Lipschitz constant of the VAE encoder f which is parameterized by θ.

While it is possible to directly use z_t to predict X_{n+k} where $k \geq 1$ [11] it is often advantageous to build a prediction framework in the latent space owing to possible complex nonlinear relationships between the underlying factors of variation and the observed data X_t. We therefore use a Variational Recurrent Neural Network (VRNN) for prediction of keypoints [4].

For inference using an RNN with inputs X_t where $t = 1, 2 \ldots, n$ and the RNN hidden state is denoted by h_t we have the following:

$h_t = f(x_t, h_{t-1}; \theta)$
where f is a nonlinear activation function, θ denotes a set of parameters and

$p(X_t|X_i, i < t) = g(h_{t-1}; \tau)$
where g is a function mapping the hidden state h_{t-1} to the output conditional probability distribution p whose parameters are given by τ.

In contrast a VRNN contains an RNN at each timestep whose hidden state h_t is given by:

$h_t = f(\psi_\tau^X(X_t), \psi_\tau^z(z_t), h_{t-1}; \theta)$
where f is a nonlinear activation function and ψ_τ^X, ψ_τ^z denote feedforward neural networks on the input X and VAE latent variable z_t respectively.

In case of the VRNN the probability distribution of the output X_t in the VAE decoder is a nonlinear function of both the VAE latent variable z_t as well as the RNN hidden state h_{t-1}.

Kernel PCA (kPCA) embeddings from the VRNN manifolds of the keypoints corresponding to the source video frames are shown in Figure 2. From this figure we see that the aforementioned properties P1 and P2 of the mapping between X_t and z_t are obeyed i.e. every input video frame X_t is mapped to a unique z_t and points which are close in X are also close in z. In addition the simplicity of the low dimensional embedding in this visualization supports our intuition that temporal prediction is easier on the manifold M_z as compared to directly performing it on M_X.

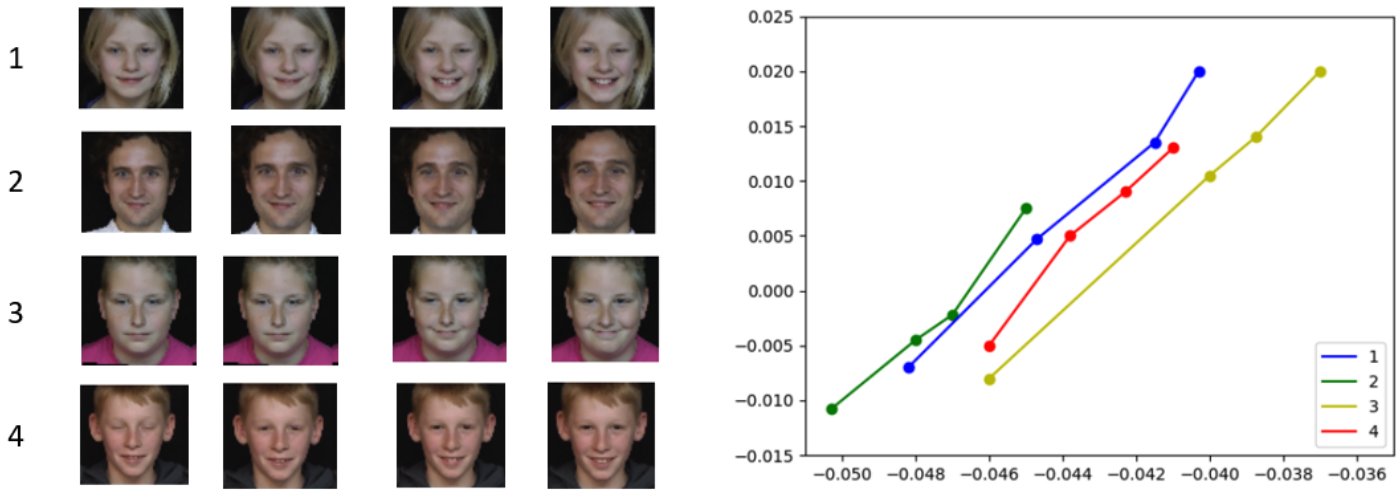

Figure 2. kPCA embedding from latent space of keypoints for 4 different video sequences of faces

4. Synthesis and Prediction pipeline for Motion Transfer

4.1. Architecture

The architecture of the Motion Transfer pipeline enabling keypoint prediction and video synthesis is shown in Figure 3. The keypoint detector $K1$ is constructed based on [12] however in this case it is applied to successive driving video frames $1, 2, \ldots, n$ of spatial dimensions H x W. The keypoint detector $K2$ is applied to a source image which can be the first frame of the same video sequence or from a different sequence. These two kinds of operations are denoted as reconstruction and transfer respectively and prediction can be performed in each mode for obtaining the final video sequence as described in Section 5. The number of keypoints K is a hyperparameter and the detector produces K heatmaps. The actual keypoints are estimated as the expectation of the corresponding heatmaps where the averag-

ing is performed over all $x \in H$ and $y \in W$ [12]. These expected values along with the mean intensity which is a continuous valued indicator of the modeled object [16] is then used to train the VRNN. Given keypoints corresponding to M video frames the VRNN is used for prediction of the keypoints in the next N video frames. Hence a full video sequence is composed of multiple groups of frames where each group is of length $M + N$. Following the predicted values from the VRNN the final videos are synthesized using the optical flow estimator E and a conditional image generator G [20].

4.2. Training and Inference

The pipeline is first trained end-to-end for video reconstruction (i.e. without prediction) using the loss functions given in [20]. Following this the keypoint detector from the trained model is used to generate keypoints to train the VRNN as per the loss function below which includes a reconstruction loss and a Kullback-Leibler divergence term between the prior N_t^{prior} and posterior N_t^{enc} [4]:

$$L_{VRNN} = \Sigma_{t=1}^{n} E\left[log\ p(X_t|z_{\leq t}, x_{\leq t}) - \beta KL(N_t^{enc}||N_t^{prior}) \right]$$

During inference a source image (A) and a sequence of video frames (B) are input to the keypoint detectors during the first M frames of a video sequence. After this no inputs are supplied for the next N frames during which phase the VRNN is used to predict the keypoints followed by which the optical flow and conditional image generation networks are used to synthesize the N video frames.

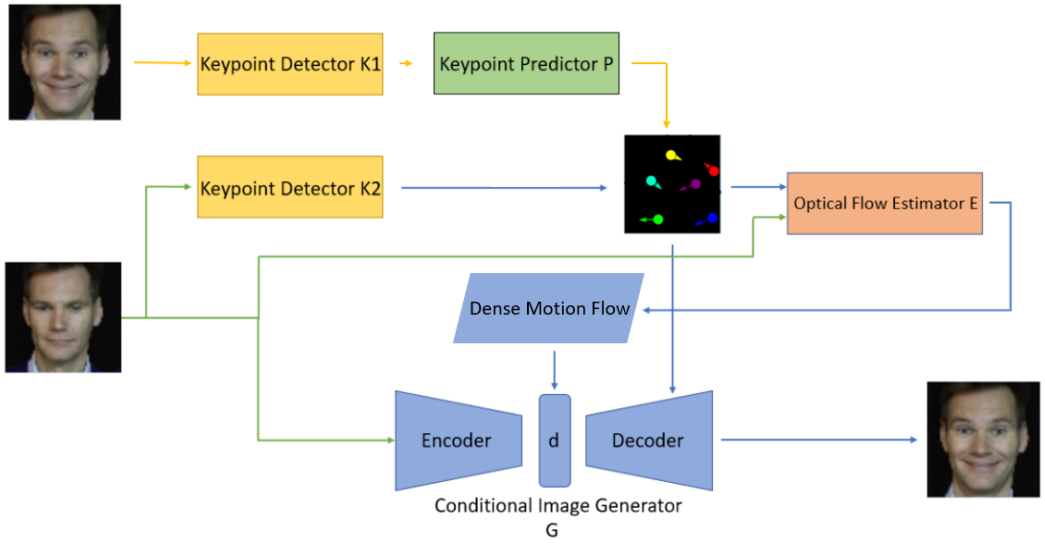

Figure 3. Components of pipeline for Motion Transfer synthesis and prediction

5. Experiments and Results

We perform our experiments with two networks, namely VRNN and RNN, on UvA-Nemo [7] and BAIR [8] datasets in both reconstruction and transfer modes. In the following subsections we first briefly describe our datasets and how these two modes operate during prediction. Following this we describe our experimental results and demonstrate improved performance of VRNN over RNN for both the Nemo dataset and the BAIR dataset in both modes.

5.1. Types of analysis

We evaluate the video generation quality on two video datasets. The UvA-Nemo dataset (hereby denoted as the Nemo dataset) contains 1240 videos, we split it into 1110 training data and 124 testing data of facial expressions with 32 frames for each. Each video consists of a person starting from no expression to a smiling face, which can help test the ability of capturing subtle changes from each frame. We also use 5001 training and 256 test videos from the BAIR dataset, with 30 frames for each video, consisting of moving robotic arms grabbing different objects. This dataset is useful to evaluate the performance of our proposed architecture in a complicated environment with varying backgrounds and irregular movements.

We compare the predictive performance of VRNN and RNN on Nemo and BAIR datasets for both reconstruction and transfer modes. In reconstruction mode the network takes the source image S and driving video D from the same video A. In this case the first frame of A is the source

image S and the following frames are the driving video frames, D. We use the first M key points K_d generated from D as input to the prediction network P (Figure 3) to generate the rest of N video keypoints K_p. Following this we use S and the combination of keypoints K_d and K_p to reconstruct the video through the whole pipeline and calculate the Mean Square Error (MSE) and Frechet Video Distance (FVD) [23] of the generated video versus the corresponding original ground truth video to evaluate the performance.

In transfer mode, the network takes the source image S and driving video D from different videos A and B. In this case the first frame of video A is used as the source image S and the frames from video B are used as the driving video D. Similar to what we do in reconstruction mode, we use the first M keypoints, K_d as input to prediction network P and then use the combination of S, K_d and K_p to generate the motion transfer video. Note that in this case there is no source video to use as ground truth for the transferred video. We therefore generate the $M + N$ video frames without prediction as ground truth to evaluate the performance of both VRNN and RNN.

Figure 4. Nemo dataset examples

Figure 5. Bair dataset examples

5.2. Results on Nemo dataset

For the Nemo dataset, we use K_d from the driving video to get K_p in the following manner:

- M1: Using the first 5 frames' keypoints to predict the following 27 frames' keypoints. In this case $M = 1, 2, 3, 4, 5$ and $N = 6, 7, \ldots, 32$. For our experiments we apply 10 model initializations.
- M2: Using 1 video frame's keypoint to predict the next 1 frame's keypoint for all video frames. In this case $M = 1, 3, 5, \ldots, 31$ and $N = 2, 4, \ldots, 32$. For our experiments we apply 5 model initializations.

Prediction results for Nemo dataset for the above cases are shown in Figures 6,7,8,9. It can be seen that in all cases VRNN conclusively outperforms RNN for both tasks of reconstructing ground truth video as well as generating motion transfer video as shown by Mean Square Error (MSE) and Frechet Video Distance (FVD). Moreover, we see that FVD shows a larger relative improvement for VRNN versus RNN than MSE. As MSE only focuses on reconstruction quality i.e. pixel-wide difference, between ground truth and generated videos, and FVD considers not only the video quality but also temporal coherence and diversity. Based on these results we can conclude that the compared to non-manifold based prediction based approach using RNN, outputs generated from VRNN have superior video quality, temporal coherence and diversity.

5.3. Results on BAIR dataset

For the BAIR dataset, we use the keypoints corresponding to the first 5 frames to predict the keypoints corresponding to the next 25 frames i.e. in this case $M = 1, 2, 3, 4, 5$ and $N = 6, 7, 8, ..., 30$. For our experiments we apply 5 model initializations. Prediction results for the BAIR dataset are shown in Figures 10,11. It can be seen that also in this case with a more complex dataset the VRNN outperforms RNN. Moreover, the results from the BAIR dataset also share the same trend as those from Nemo dataset where FVD shows a larger relative improvement for VRNN versus RNN than MSE. This signifies that using prediction with VRNN versus RNN leads to the generation of outputs with better video quality, temporal coherence and sample diversity.

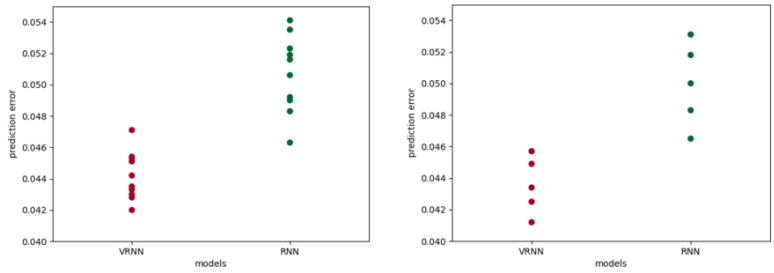

Figure 6. Nemo reconstruction mode MSE error. Left: M1 Right: M2

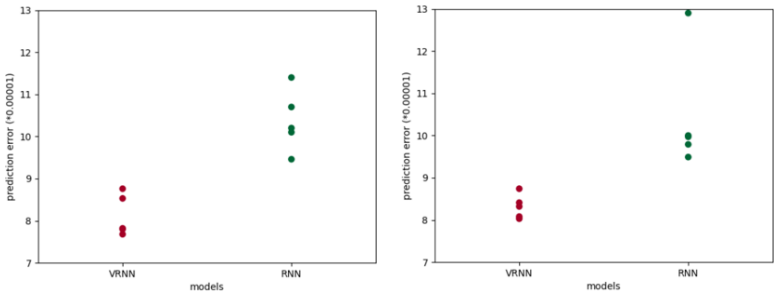

Figure 7. Nemo reconstruction mode FVD error. Left: M1 Right: M2

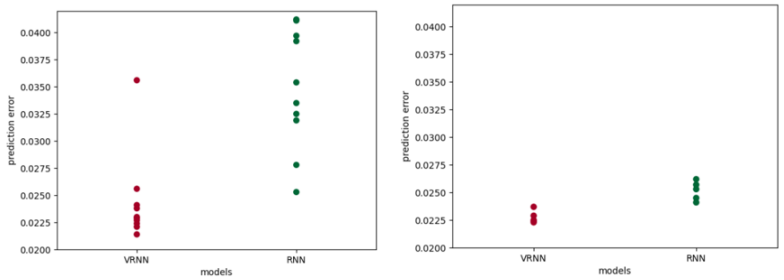

Figure 8. Nemo transfer mode MSE error. Left: M1 Right: M2

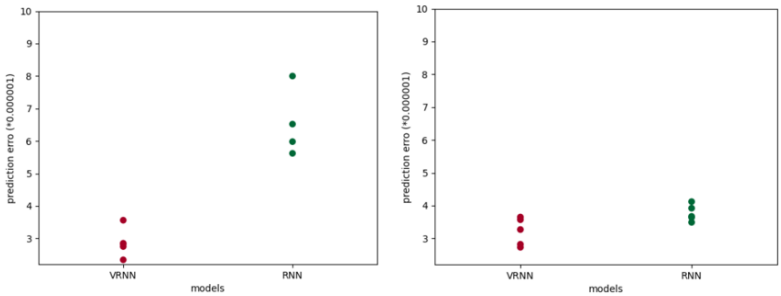

Figure 9. Nemo transfer mode FVD error. Left: M1 Right: M2

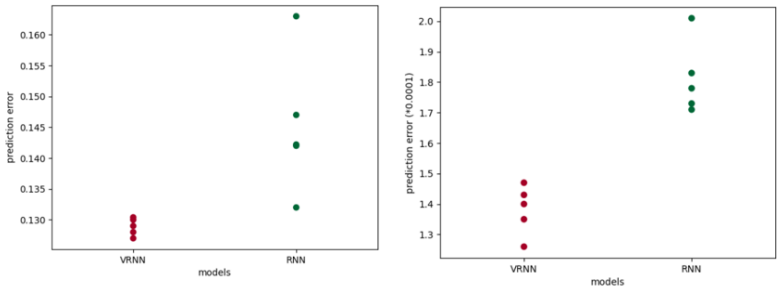

Figure 10. Bair reconstruction mode error. Left: MSE Right: FVD

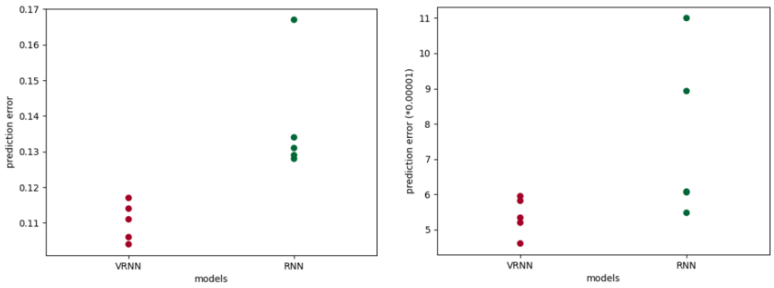

Figure 11. Bair transfer mode error. Left: MSE Right: FVD

6. Conclusions and Future Work

In this paper we demonstrate a VRNN enabled manifold learning based prediction and synthesis flow for implementing motion transfer in streaming applications such as video conferencing. Using prediction in the motion transfer architecture enables bandwidth savings of 2x or more for the driving video data as compared to case where prediction is not used. With representative datasets and different prediction horizons we compare the performance of manifold-based prediction method using VRNN over a non-manifold based prediction method using RNN. For all datasets we demonstrate the superior performance of VRNN for video prediction in both cases of reconstruction and transfer. Our future plans include developing manifold based prediction techniques for high dimensional time series which combine nonlinear dimensionality reduction and Deep Generative Models such as VAEs for enabling superior performance in motion transfer based applications.

References

[1] Pablo Arias, Gregory Randall, and Guillermo Sapiro. Connecting the out-of-sample and pre-image problems in kernel methods. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2007. 2

[2] Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural computation*, 15(6):1373–1396, 2003. 2

[3] Peter J Brockwell and Richard A Davis. *Time series: theory and methods*. Springer Science & Business Media, 2009. 2

[4] Junyoung Chung, Kyle Kastner, Laurent Dinh, Kratarth Goel, Aaron C Courville, and Yoshua Bengio. A recurrent latent variable model for sequential data. *Advances in neural information processing systems*, 28:2980–2988, 2015. 1, 3, 4

[5] Alexander Cloninger, Wojciech Czaja, and Timothy Doster. The pre-image problem for laplacian eigenmaps utilizing l1 regularization with applications to data fusion. *Inverse Problems*, 33(7):074006, 2017. 2

[6] Ronald R Coifman and Stéphane Lafon. Diffusion maps. *Applied and computational harmonic analysis*, 21(1):5–30, 2006. 2

[7] Hamdi Dibeklioğlu, Albert Ali Salah, and Theo Gevers. Are you really smiling at me? spontaneous versus posed enjoyment smiles. In *European Conference on Computer Vision*, pages 525–538. Springer, 2012. 4

[8] Frederik Ebert, Chelsea Finn, Alex X Lee, and Sergey Levine. Self-supervised visual planning with temporal skip connections. In *CoRL*, pages 344–356, 2017. 4

[9] Charles Fefferman, Sanjoy Mitter, and Hariharan Narayanan. Testing the manifold hypothesis. *Journal of the American Mathematical Society*, 29(4):983–1049, 2016. 2

[10] Henry Gouk, Eibe Frank, Bernhard Pfahringer, and Michael J Cree. Regularisation of neural networks by enforcing lipschitz continuity. *Machine Learning*, 110(2):393–416, 2021. 2

[11] Carlos X Hernández, Hannah K Wayment-Steele, Mohammad M Sultan, Brooke E Husic, and Vijay S Pande. Variational encoding of complex dynamics. *Physical Review E*, 97(6):062412, 2018. 3

[12] Tomas Jakab, Ankush Gupta, Hakan Bilen, and Andrea Vedaldi. Unsupervised learning of object landmarks through conditional image generation. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 4020–4031, 2018. 1, 3, 4

[13] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 2

[14] Henry Li, Ofir Lindenbaum, Xiuyuan Cheng, and Alexander Cloninger. Variational diffusion autoencoders with random walk sampling. In *European Conference on Computer Vision*, pages 362–378. Springer, 2020. 2

[15] Zachary C Lipton, John Berkowitz, and Charles Elkan. A critical review of recurrent neural networks for sequence learning. *arXiv preprint arXiv:1506.00019*, 2015. 2

[16] Matthias Minderer, Chen Sun, Ruben Villegas, Forrester Cole, Kevin Murphy, and Honglak Lee. Unsupervised learning of object structure and dynamics from videos. *arXiv preprint arXiv:1906.07889*, 2019. 2, 4

[17] Hariharan Narayanan and Sanjoy Mitter. Sample complexity of testing the manifold hypothesis. In *Proceedings of the 23rd International Conference on Neural Information Processing Systems-Volume 2*, pages 1786–1794, 2010. 2

[18] Ali Rahimi, T Darrell, and B Recht. Learning appearance manifolds from video. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 868–875. IEEE, 2005. 2

[19] Uri Shaham, Kelly Stanton, Henry Li, Boaz Nadler, Ronen Basri, and Yuval Kluger. Spectralnet: Spectral clustering using deep neural networks. *arXiv preprint arXiv:1801.01587*, 2018. 2

[20] Aliaksandr Siarohin, Stéphane Lathuilière, Sergey Tulyakov, Elisa Ricci, and Nicu Sebe. Animating arbitrary objects via deep motion transfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2377–2386, 2019. 1, 4

[21] Ronen Talmon and Ronald R Coifman. Empirical intrinsic geometry for nonlinear modeling and time series filtering. *Proceedings of the National Academy of Sciences*, 110(31):12535–12540, 2013. 2

[22] Ronen Talmon and Ronald R Coifman. Intrinsic modeling of stochastic dynamical systems using empirical geometry. *Applied and Computational Harmonic Analysis*, 39(1):138–160, 2015. 2

[23] Thomas Unterthiner, Sjoerd van Steenkiste, Karol Kurach, Raphael Marinier, Marcin Michalski, and Sylvain Gelly. Towards accurate generative models of video: A new metric & challenges. *arXiv preprint arXiv:1812.01717*, 2018. 5

[24] Ting-Chun Wang, Arun Mallya, and Ming-Yu Liu. One-shot free-view neural talking-head synthesis for video conferencing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10039–10049, 2021. 1

[25] Junhai Zhai, Sufang Zhang, Junfen Chen, and Qiang He. Autoencoder and its various variants. In *2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 415–419. IEEE, 2018. 2