Comparing linear structure-based and data-driven latent spatial representations for sequence prediction

Myriam Bontonou, Carlos Lassance, Vincent Gripon, Nicolas Farrugia

HAL Id: hal-02189734

https://hal-imt-atlantique.archives-ouvertes.fr/hal-02189734

Submitted on 19 Jul 2019

Comparing linear structure-based and data-driven latent spatial representations for sequence prediction

Myriam Bontonou[a,b], Carlos Lassance[a,b], Vincent Gripon[a,b], and Nicolas Farrugia[a]

[a]IMT Atlantique and Lab-STICC, France
[b]Université de Montréal and MILA, Canada

ABSTRACT

Predicting the future of Graph-supported Time Series (GTS) is a key challenge in many domains, such as climate monitoring, finance or neuroimaging. Yet it is a highly difficult problem as it requires to account jointly for time and graph (spatial) dependencies. To simplify this process, it is common to use a two-step procedure in which spatial and time dependencies are dealt with separately. In this paper, we are interested in comparing various linear spatial representations, namely structure-based ones and data-driven ones, in terms of how they help predict the future of GTS. To that end, we perform experiments with various datasets including spontaneous brain activity and raw videos.

Keywords: Graph-supported Time Series, Latent Representations, Auto-Encoders, Graph Signal Processing, Sequence Prediction, Spatio-temporal Sequences

1. INTRODUCTION

Spatially structured temporal sequences are everywhere: in video prediction, in climate monitoring, in functional Magnetic Resonance imaging (fMRI) and so on. Although much progress has been made (c.f. Section 2), predicting their evolution over time remains a highly difficult challenge.

Such sequences are naturally described as signals over a graph (or a series of graphs), evolving over time. For instance, in the case of fMRI signals, the nodes of the graph are groups of neurons called Regions Of Interest (ROI), the spatial structure is described by the edges of the graph, which represent either the physical connections or statistical similarities between signals across ROIs. The fMRI signal is an indirect measure of brain activity through the blood-oxygen level dependent (BOLD) signal.[1] Averaging such signals in ROIs results in a scalar for each node of the graph at each time point.

The prediction of the evolution of such sequences in the raw spatial domain can prove to be very challenging, especially if the spatial domain is subject to important noise. Instead, we would prefer to predict the future of these sequences in a simplified latent space representation, in which the content is described using sparse, robust, low-dimensional representations.

Finding such representations is a common problem in signal processing and data science. The most popular methods are principal components analysis and independent component analysis. In signal processing, recent research efforts attempt at exploiting graphs to derive such low-dimensional representations, by defining the Graph Fourier Transform (GFT), or the spectral graph wavelet transform.[2] In the context of optimization, popular methods include dictionary learning[3] and non-negative matrix factorization.[4] In machine learning, auto-encoders and their variants are popular frameworks based on neural networks to learn latent representations.[5] Because it simply consists in changing the basis in which data is represented, GFT comes with a rich set of guarantees, theorems and intuitions. And as such, it is a strong candidate in a rising field of research.[6] However, it is known that GFT can fail at fully leveraging the underlying structure, even in the case of very regular domains.[7]

The question we ask in this paper is: what are the linear transforms that are the best fitted to predict the evolution of GTS?

We consider three approaches. First, we introduce a fully structured-based approach, where the linear transform is defined using the lower part of the spectrum of the Graph Fourier basis. Second, we consider a mixed strategy, where both the structure and data are considered while looking for the best basis. To this end, we use a semi-geometric graph which support is the same as for the first case, but where the weights of the edges account for the covariance of the signals measured at different nodes. Finally, we consider a learned scheme, where the structure is disregarded and only data helps in finding the good representation.

The question we ask is twofold:

1. Which linear transforms give the best latent representation for spatial compression of the signals?

2. Which linear transforms give the most fitted representations for sequence (time) prediction?

As we consider data-driven approaches in our study, we mainly support our claims through experimental validation. We thus consider three datasets made of images and fMRI signals.

The outline of the article is as follows. In Section 2, we introduce related works. In Section 3, we explain our methodology to obtain latent spatial representations and perform sequence (time) prediction. In Section 4, we empirically compare different approaches. Finally, Section 5 is a discussion and Section 6 a conclusion.

2. RELATED WORK

Many studies so far have dealt with temporal prediction of spatially structured data. For instance, in deep learning, variants of Recurrent Neural Networks (RNNs), such as Long-Short Time Memory (LSTM),[8] are the basis of temporal prediction. These networks build an internal memory about previous data. Thus, they are able to remember previous information to inform later steps. Data is input frame after frame in the networks. At each step, they output an estimate of the future frame. However, the networks do not handle directly data structure. In past years, some works addressed the problem. For instance, in video prediction, data is images. Replacing the RNNs internal linear operations by convolutions allows them to handle the image structure. It has been shown that it improves frame prediction.[9–11] More recently, prediction on data residing on the nodes of graphs have received much attention. A basic idea is to replace the RNNs internal linear operations by graph convolutions, but the best results are achieved using more complex architectures.[12, 13]

In Deep Learning, most methods act as black boxes in which parameters are tuned using gradient descent, without the ability to understand the purpose of each of them. Finding the good hyperparameters can be very demanding, as the combinatorial search space rapidly explodes. Therefore, it is hard to obtain guarantees about robustness of the method, or even reproducibility on other datasets.

When it comes to graphs, a natural approach to define a latent representation is a structure-based approach. It consists in defining the notion of frequency on a graph by analogy with the Fourier Transform (FT).[6] The so-called Graph Fourier Transform (GFT) represents the signal in another basis. This basis is built according to the graph structure without looking at the data at all. The eigenvectors of the Laplacian matrix of the graph are used as a new basis. These eigenvectors are ordered by increasing eigenvalues. Small eigenvalues correspond intuitively to low frequencies, and large eigenvalues to high frequencies. Consequently, to reduce the data dimension, only some of the first eigenvectors are kept. Data is back projected using only those first eigenvectors, and a low-dimensional representation of the signal is obtained. However, the analogy with the FT is not complete, as for instance, the GFT using a 2-dimensional grid does not match the 2D classical FT.[7]

Finding the best graph structure for a given problem is a question that recently sparked a lot of interest in the literature.[14–16] In many cases, a good solution consists in using the covariance matrix (or its inverse) as the adjacency matrix of the graph. Following this lead, in this paper, we make use of a semi-geometric graph. It combines the geometry of images captured through a regular grid graph with the covariance of the pixels measured on different frames. As such, the support of the graph remains the same as that of the grid graph,

but the weights are replaced by the covariance between corresponding pixels. Therefore, this graph takes into account both the data structure and the data distribution.

Another way to represent data in a low-dimensional space is to use an auto-encoder. An auto-encoder is a deep neural network that is trained to reproduce its input as output, going through an intermediate latent representation. As such, the first part of the auto-encoder compresses data in a low-dimensional space. The second part decompresses the latent representation to evaluate how well it represents the raw data. A variety of complex non-linear auto-encoders have been proposed,[5,17] but for a fair comparison, in this article we only use a simple variant that mimics the principles of GFT.

In this work, much of the interest in the latent representation is motivated by its use for sequence time prediction. There are, again, several recent works in this area, based on diverse methods (dictionary learning,[18] source separation[19]). Again, we restrict the study to a simple question: what is the best linear, structure-based or data-driven, representation for predicting future frames using a LSTM?

3. METHODOLOGY

We consider spatially structured temporal sequences of data. These data are intra-connected. Let us take our two examples, images and fMRI signals. Images are spatially structured in the sense that each of their pixel correspond to a position inside the image. These positions can be neighbors or to the contrary far away. Similarly, fMRI signals are spatially structured, as these signals are associated with a group of neurons (called regions of interest (ROIs)). Each ROI is physically connected to other ROIs.

In both cases a simple way to represent data structure is to use a graph. Each variable (pixels, ROIs) is associated with a node of the graph. Then, connections are drawn between these variables and represented by edges in the graph. The value of each variable is seen as a signal (i.e. a vector) where each dimension corresponds to a node in the graph.

To perform (time) sequence prediction, a simple way is to train a Fully-Connected Long Short-Term Memory (FC-LSTM) neural network on it. But it can prove to be challenging (memory loss) as it does not take into account data structure. Here, we are interested in modifying the input of the FC-LSTM to make it reflect data structure. We begin with the simpler question of finding a linear spatial latent representation. Then, we investigate whether this latent representation helps the FC-LSTM provide more accurate predictions.

3.1 Comparing linear latent representations of spatially structured data

In this section, we forget the temporal aspect of the sequences. We only evaluate the ability of linear methods to effectively compress, and then reconstruct, the raw data.

Let $G = \langle V, \mathbf{W} \rangle$ be a weighted graph with n nodes. $V = \{1, \ldots, n\}$ is the finite set of n nodes and $\mathbf{W}$ is the weighted adjacency matrix. $\mathbf{W}_{ij}$ is the weight of the edge between nodes i and j, or 0 if no such edge exists. $\mathbf{x} \in \mathbb{R}^n$ is a signal on this graph, so that its i-th component corresponds to node i in G.

A structure-based approach used on data reposing on graphs is the *Graph Fourier Transform* (GFT).[6] This approach builds an analogy with the Fourier Transform. Let us consider the (combinatorial) Laplacian matrix $\mathbf{L} = \mathbf{D} - \mathbf{W}$, where $\mathbf{D}$ is the diagonal degree matrix: $\mathbf{D}_{ii} = \sum_{j=1}^{n} \mathbf{W}_{ij}$. Provided that $\mathbf{W}$ is real symmetric, it admits a set of orthonormal eigenvectors $\{\mathbf{u}_l\}_{l=1}^{n} \in \mathbb{R}^n$. We denote $\mathbf{U} = [\mathbf{u}_1, ..., \mathbf{u}_n] \in \mathbb{R}^{n \times n}$ and $\mathbf{U}^T$ the transpose of $\mathbf{U}$. By analogy with the FT, the GFT of the signal $\mathbf{x}$ is defined as: $\hat{\mathbf{x}} = \mathbf{U^T}\mathbf{x}$ and the inverse GFT is defined as: $\mathbf{x} = \mathbf{U}\,\hat{\mathbf{x}}$. From that, we want to compute a latent representation with m latent variables. By taking only a subset of eigenvectors, the ones associated with the m lowest eigenvalues, we build a latent representation of the data. The complexity of this method is in general cubic with the number of nodes in the graph, since it requires diagonalizing a matrix.

In the case of fMRI signals, we use a graph computed from the correlation matrix of the measurements between the ROIs over time, thresholded to keep only the edges with the 5 % largest weights. In the case of images, we consider two kind of graphs. the first one is a grid graph, completely agnostic of the data. In this graph, each pixel is connected to its neighbors pixels (left, right, up, down). The edge weight is either 0 (no

connection) or one (connection). The other graph is a semi-geometric graph. This graph has the same support as a grid but the edge weight is now the covariance between the pixels values over time.

To be able to use GFT requires to have access to a graph describing the spatial domain of the data. Such graphs are not always accessible. An alternative is to use a data-driven approach. In this work we choose to use auto-encoders. For a fair comparison, we define a very basic version of an auto-encoder. Generally speaking, an auto-encoder is split into two parts: an encoder and a decoder. In this study, the encoder is a linear transformation. Formally, given the signal $\mathbf{x} \in \mathbb{R}^n$ and a weight matrix $\mathbf{A} \in \mathbb{R}^{n \times m}$, the encoder is defined as $\hat{\mathbf{x}} = \mathbf{A}^T \mathbf{x}$. The decoder uses the transposed matrix $\mathbf{A}$ to reconstruct the signal $\tilde{\mathbf{x}} = \mathbf{A}\, \hat{\mathbf{x}}$. Thus, its mathematical formulation is very similar to that of the GFT, with the noticeable difference that it does not require priors about the spatial structure of data.

The Mean Square Error (MSE) between raw images and reconstructed images is used to train the AE. It is also used to evaluate the accuracy of all methods.

3.2 Does the latent representation help in having better predictions?

Once latent representations have been obtained, using either the GFT or an auto-encoder (AE), the next step is to use them to predict the future of a sequence. In this work, we use a FC-LSTM to predict the evolution of sequences of data. Spatially compressed sequences are input into the FC-LSTM frame by frame. For each element $\mathbf{x}_t \in \mathbb{R}^m$, two functions are updated: the cell state c_t and the hidden state h_t. The cell state acts as the memory of FC-LSTM. The hidden state corresponds to the prediction of the next element. More precisely, at time t, the input, forget, cell and output gates are computed, respectively $\mathbf{i}_t$, $\mathbf{f}_t$, $\mathbf{c}_t$ and $\mathbf{o}_t$. All of them are vectors in $\mathbb{R}^m$. $\mathbf{W}_{..}$ and $\mathbf{b}_{..}$ are trainable parameters. σ is the sigmoid function. $*$ is the Hadamard product.

$$
\begin{aligned}
\mathbf{i}_t &= \sigma(\mathbf{W}_{ii}\mathbf{x}_t + \mathbf{b}_{ii} + \mathbf{W}_{hi}\mathbf{h}_{t-1} + \mathbf{b}_{hi}), \\
\mathbf{f}_t &= \sigma(\mathbf{W}_{if}\mathbf{x}_t + \mathbf{b}_{if} + \mathbf{W}_{hf}\mathbf{h}_{t-1} + \mathbf{b}_{hf}), \\
\mathbf{g}_t &= \tanh(\mathbf{W}_{ig}\mathbf{x}_t + \mathbf{b}_{ig} + \mathbf{W}_{hg}\mathbf{h}_{t-1} + \mathbf{b}_{hg}), \\
\mathbf{o}_t &= \sigma(\mathbf{W}_{io}\mathbf{x}_t + \mathbf{b}_{io} + \mathbf{W}_{ho}\mathbf{h}_{t-1} + \mathbf{b}_{ho}), \\
\mathbf{c}_t &= \mathbf{f}_t * \mathbf{c}_{t-1} + \mathbf{i}_t * \mathbf{g}_t, \\
\mathbf{h}_t &= \mathbf{o}_t * tanh(\mathbf{c}_t).
\end{aligned}
\tag{1}
$$

Sequences are split into two parts: the first is used to warm up the FC-LSTM and the second to evaluate its prediction ability. Indeed, at the beginning, the cell state and the hidden state are initialized at zero. Then, for each frame of the first part, the frame is input into the FC-LSTM which outputs a prediction. Finally, for each frame of the second part, the previous prediction is input into the FC-LSTM which again outputs a prediction.

The Mean Square Error (MSE) between frames and predictions is used to train the FC-LSTM. However, for the evaluation, only the MSE between frames and predictions of the second part is displayed in the results.

3.3 Datasets

In this study, we use three datasets, namely MovingSTL, MovingMNIST and a dataset of resting state fMRI.

MovingSTL is built from STL10.[20] STl10 images are 96×96 pixels. We transform these images into grey images, normalized between [-1, 1]. We use 5000 images. 3500 for training. 1500 for testing. Each image is used to define a sequence. A square, 45×45 pixels, is randomly cropped in the image. This square randomly moves by one pixel 20 times in a random direction. Thus, we define 5000 sequences of 20 images of 45×45 pixels.

For MovingMNIST,[21] we pick 5000 sequences of 20 images. 3500 for training. 1500 for testing. Each image is 64×64 pixels. A sequence displays a moving white number on a black background.

The fMRI dataset is a subset of 62 subjects from the MPILMBB resting-state dataset.[22] Participants were measured four times at rest with eyes-open, fixating a cross during approximately 15 minutes (TR = 1400 ms; voxel size = 2.3 mm isotropic). Here, we only use the first measurement for each subject. Preprocessing and de-noising are described with more details here https://neuroanatomyandconnectivity.github.io/opendata/.

The denoised data was z-scored and normalized to MNI space and parcellated on 523 non-overlapping ROIs from the finest scale of BASC atlas (444 networks).[23] Two subjects were rejected because of corrupted data for one subject (the time series was shorter than expected), and because of excessive head motion for the other subject. 60 subjects remain, and we use 48 subjects for training and the 12 remaining subjects for testing.

On MovingSTL and MovingMNIST, the auto-encoder is trained for 400 epochs using 100 examples per mini-batch, with Adam, using a learning rate of 1e-5 with a weight decay factor that starts at 1e-5 and is divided by 10 at epochs 4 and 120. On the fMRI signals, the auto-encoder is trained for 400 epochs using 6 examples per mini-batch, with Adam, using a learning rate that starts at 1e-5, and is divided by 2 at epoch 200 with a weight decay factor of 1e-5. Only MovingSTL and the fMRI signals are used for evaluating whether the latent representation helps in having better predictions. On both, the FC-LSTM is trained for 600 epochs using 6 examples per mini-batch, with Adam, using a learning rate that starts at 0.001 and is divided by 2 at epochs 200 and 400, with no weight decay.

4. RESULTS

4.1 Linear latent representations of spatially structured data

In the case of images, we evaluate the ability of our three linear transforms to reconstruct raw images from their compressed representations. The three transforms are: the Graph Fourier Transform applied on a grid graph (GFT), the Graph Fourier Transform applied on a semi-geometric graph (GEO) and a linear auto-encoder (AE). We compute the MSE between raw images, from MovingSTL and MovingMNIST, and their reconstruction in function of the number of latent representations. See Fig. 1. For MovingSTL, the figure shows that, whatever the latent dimension, the AE reconstructs better the raw data than GFT. The results of AE and GEO are very similar. For MovingMNIST, the AE is also better but the GEO results are worse. In the case of fMRI data, we evaluate the ability of the Graph Fourier Transform applied on a correlation graph (GFT) and a linear auto-encoder (AE). We compute the MSE between raw data and their reconstruction in function of the number of latent dimensions for 523 regions of interest (ROI) and for 171 ROI. See Table 1. Just to notice it, the data for 171 ROI has been obtained after a hierarchical clustering of the 523 ROI. Again, the AE obtains a better MSE, whatever the latent dimension.
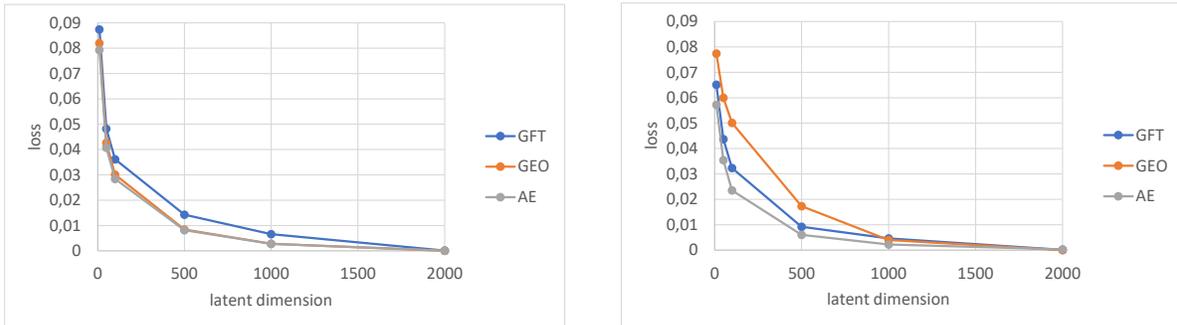
(a) MovingSTL test set   (b) MovingMNIST test set

Figure 1: MSE loss as a function of the number of latent dimensions m

	Loss	
Latent dimension	86	43
GFT	0.126	0.246
AE	0.065	0.134

(a) 171 ROI

	Loss	
Latent dimension	262	131
GFT	0.174	0.317
AE	0.102	0.189

(b) 523 ROI

Table 1: MSE loss on the fMRI test set as a function of the number of ROI and the number of latent dimensions.

	Prediction MSE	Reconstruction MSE
GFT	0.094	0.007
GEO	0.093	0.003
AE	0.096	0.003

(a) Latent dimension: 1000

	Prediction MSE	Reconstruction MSE
GFT	0.093	0.014
GEO	0.091	0.008
AE	0.095	0.008

(b) Latent dimension: 500

Table 2: FC-LSTM on MovingSTL test set: prediction of 10 images when the 9 previous are given. Reconstruction MSE: MSE between raw images and reconstructed images, computed independently of the FC-LSTM predictions. Prediction MSE: MSE between raw images and reconstructed predictions. For information, the MSE obtained without using compressed representations, with a FC-LSTM trained with the same number of epochs, is 0.106.

4.2 Predicting the evolution of the sequences with latent representations

In this section, we evaluate the ability of a FC-LSTM to predict the dynamics of GTS from compressed representations. In the case of MovingSTL, for a given number of latent dimensions, the prediction MSE for the three linear transforms is similar. See Table 2. As the reconstruction errors of the three methods are not the same, it raises questions. We discuss this result in the next section. We also notice that their predictions are better than the ones obtained without latent representations. Appendix A contains two examples of predictions. In the case of the fMRI measurements, unfortunately, we were not able to obtain accurate predictions. Whatever the method, the prediction converges quickly towards the average signal value. It is not surprising as our FC-LSTM is only able to predict the structure of a few frames before it starts to converge towards an average value as well.

5. DISCUSSION

Let us recall that we wanted to compare several linear spatial latent representations for spatial compression and time prediction.

As far as spatial compression is concerned, it is not surprising that the learned representation gives the best results. Indeed, this is the only representation optimised for the task. Interestingly, in the case of MovingSTL, the semi-geometric graph is very competitive. Again, this result was expected because STL is made of natural images that are likely to be low frequency on the grid graph. In the case of MovingMNIST, the grid graph poorly captures the regularities of the graph signals, leading to a strong advantage for the learned representation. This may come from the fact that MovingMNIST images may not be smooth on the graph, as a consequence low frequencies might not capture efficiently the structure of the actual signals. Previous work on brain data has indeed shown that the high graph frequencies can also be important in capturing latent representations.[2,24]

As far as time prediction is concerned, we observed an interest in using spatial latent representations to predict the future of sequences. Another interesting finding of our study lies in the fact that all representations perform similarly. We see three possible explanations. A first hypothesis is that FC-LSTMs trained for the task contain enough parameters to handle both spatial and temporal predictions. This would explain why the initial advantage of learned representations is lost. A second hypothesis is that by construction GFT transforms are likely to be sparse, providing a much easier task. On the contrary, we did not impose any constraints on the latent representations trained using the auto-encoder. Finally, the third hypothesis is that our FC-LSTMs are not able to capture the complex dynamics of our dataset.

Obviously, this comparison is highly theoretical. In practice, having access to the underlying spatial domain of the data, or to a larger number of samples to train the latent representation, is rare. But we found interesting that our three considered approaches ended performing so similarly in the end. The exact reasons remain unclear.

6. CONCLUSION

In this study, we were interested in predicting spatially structured temporal sequences of data. More precisely, we wondered whether linear transforms of such data help a FC-LSTM make more accurate predictions. We explored three linear transforms: a structured-based transform with the Graph Fourier Transform, a data-driven

approach with a linear auto-encoder, and a mixed approach using a semi-geometric graph. We also introduced a dataset for this study. Our experiments confirmed the interest of a prior spatial compression of the data before predicting future frames. They also showed that domain-agnostic and data-agnostic solutions are on par on this task. In future work, we would like to investigate further the fundamental reasons why all compared methods behaved similarly. We would also like to encompass more diverse representations of data, including nonlinear variations of auto-encoders and wavelets transforms on graphs.
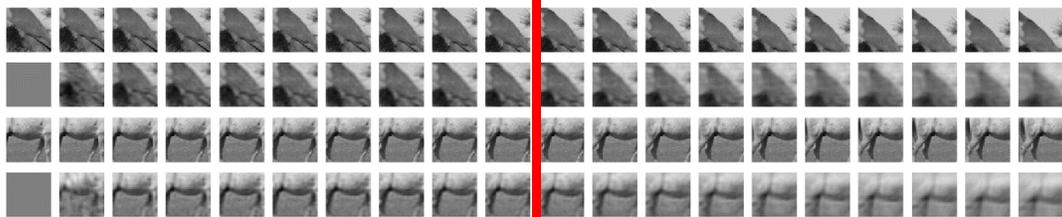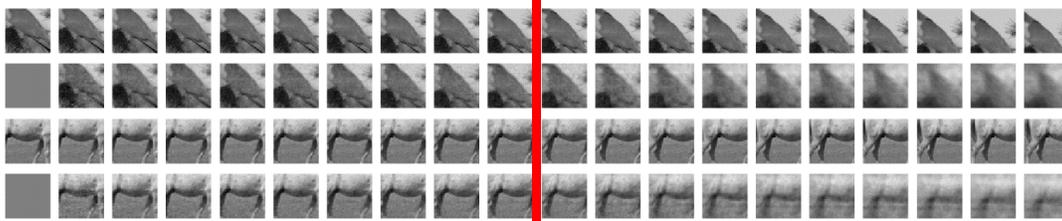
APPENDIX A. PREDICTIONS OBTAINED ON MOVINGSTL
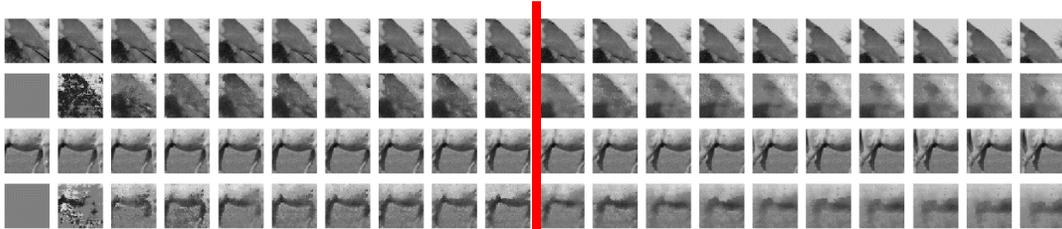
(a) FC-LSTM trained on raw images.
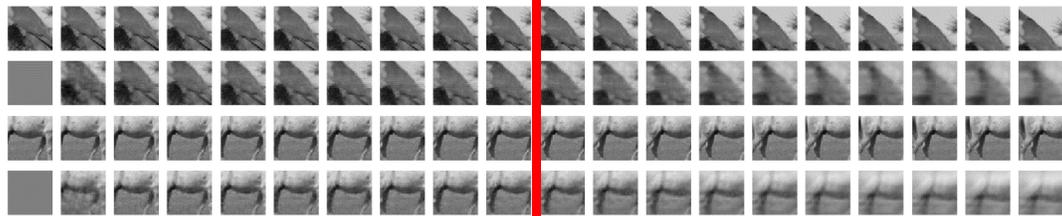
(b) FC-LSTM trained on images tranformed with GFT.

(c) FC-LSTM trained on images transformed with GEO.

(d) FC-LSTM trained on images transformed with AE.

Figure 2: FC-LSTM on MovingSTL. Two examples are plotted for each method: (a) without any method, (b) with the GFT on a grid graph, (c) with the GFT on a semi-geometric graph, (d) with a linear auto-encoder. The first row of each example displays the real images. The second row displays the reconstructed output of the FC-LSTM. The 10 first images are used for warming up the FC-LSTM memory. The 10 last images (after the red line) are predictions (the FC-LSTM no longer have access to the previous real image).

REFERENCES

[1] Ogawa, S., Lee, T.-M., Kay, A. R., and Tank, D. W., "Brain magnetic resonance imaging with contrast dependent on blood oxygenation," *proceedings of the National Academy of Sciences* **87**(24), 9868–9872 (1990).

[2] Pilavci, Y. Y. and Farrugia, N., "Spectral graph wavelet transform as feature extractor for machine learning in neuroimaging," in [*ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*], 1140–1144, IEEE (2019).

[3] Mairal, J., Bach, F., Ponce, J., and Sapiro, G., "Online dictionary learning for sparse coding," in [*Proceedings of the 26th annual international conference on machine learning*], 689–696, ACM (2009).

[4] Lee, D. D. and Seung, H. S., "Learning the parts of objects by non-negative matrix factorization," *Nature* **401**(6755), 788 (1999).

[5] Bengio, Y., Courville, A., and Vincent, P., "Representation learning: A review and new perspectives," *IEEE transactions on pattern analysis and machine intelligence* **35**(8), 1798–1828 (2013).

[6] Shuman, D. I., Narang, S. K., Frossard, P., Ortega, A., and Vandergheynst, P., "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," *arXiv preprint arXiv:1211.0053* (2012).

[7] Lassance, C. E. R. K., Vialatte, J.-C., and Gripon, V., "Matching convolutional neural networks without priors about data," in [*2018 IEEE Data Science Workshop (DSW)*], 234–238, IEEE (2018).

[8] Hochreiter, S. and Schmidhuber, J., "Long short-term memory," *Neural computation* **9**(8), 1735–1780 (1997).

[9] Xingjian, S., Chen, Z., Wang, H., Yeung, D.-Y., Wong, W.-K., and Woo, W.-c., "Convolutional lstm network: A machine learning approach for precipitation nowcasting," in [*Advances in neural information processing systems*], 802–810 (2015).

[10] Wang, Y., Long, M., Wang, J., Gao, Z., and Philip, S. Y., "Predrnn: Recurrent neural networks for predictive learning using spatiotemporal lstms," in [*Advances in Neural Information Processing Systems*], 879–888 (2017).

[11] Wang, Y., Jiang, L., Yang, M.-H., Li, L.-J., Long, M., and Fei-Fei, L., "Eidetic 3d lstm: A model for video prediction and beyond," (2018).

[12] Li, Y., Yu, R., Shahabi, C., and Liu, Y., "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," *arXiv preprint arXiv:1707.01926* (2017).

[13] Zhang, J., Shi, X., Xie, J., Ma, H., King, I., and Yeung, D.-Y., "Gaan: Gated attention networks for learning on large and spatiotemporal graphs," *arXiv preprint arXiv:1803.07294* (2018).

[14] Pasdeloup, B., Gripon, V., Mercier, G., Pastor, D., and Rabbat, M. G., "Characterization and inference of graph diffusion processes from observations of stationary signals," *IEEE Transactions on Signal and Information Processing over Networks* **4**(3), 481–496 (2017).

[15] Dong, X., Thanou, D., Frossard, P., and Vandergheynst, P., "Learning laplacian matrix in smooth graph signal representations," *IEEE Transactions on Signal Processing* **64**(23), 6160–6173 (2016).

[16] Marques, A. G., Segarra, S., Leus, G., and Ribeiro, A., "Stationary graph processes and spectral estimation," *IEEE Transactions on Signal Processing* **65**(22), 5911–5926 (2017).

[17] Chen, X., Kingma, D. P., Salimans, T., Duan, Y., Dhariwal, P., Schulman, J., Sutskever, I., and Abbeel, P., "Variational lossy autoencoder," *arXiv preprint arXiv:1611.02731* (2016).

[18] La Tour, T. D., Moreau, T., Jas, M., and Gramfort, A., "Multivariate convolutional sparse coding for electromagnetic brain signals," in [*Advances in Neural Information Processing Systems*], 3292–3302 (2018).

[19] Hjelm, R. D., Damaraju, E., Cho, K., Laufs, H., Plis, S. M., and Calhoun, V. D., "Spatio-temporal dynamics of intrinsic networks in functional magnetic imaging data using recurrent neural networks," *Frontiers in neuroscience* **12**, 600 (2018).

[20] Coates, A., Ng, A., and Lee, H., "An analysis of single-layer networks in unsupervised feature learning," in [*Proceedings of the fourteenth international conference on artificial intelligence and statistics*], 215–223 (2011).

[21] Szeto, R., Stent, S., Ros, G., and Corso, J. J., "A dataset to evaluate the representations learned by video prediction models," in [*International Conference on Learning Representations (Workshop Track)*], (Apr 2018).

[22] Mendes, N., Oligschlaeger, S., Lauckner, M. E., Golchert, J., Huntenburg, J. M., Falkiewicz, M., Ellamil, M., Krause, S., Baczkowski, B. M., Cozatl, R., et al., "A functional connectome phenotyping dataset including cognitive state and personality measures," *Scientific data* **6**, 180307 (2019).

[23] Bellec, P., Rosa-Neto, P., Lyttelton, O. C., Benali, H., and Evans, A. C., "Multi-level bootstrap analysis of stable clusters in resting-state fMRI," *NeuroImage* **51**(3), 1126–1139 (2010).

[24] Ménoret, M., Farrugia, N., Pasdeloup, B., and Gripon, V., "Evaluating graph signal processing for neuroimaging through classification and dimensionality reduction," in [*2017 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*], 618–622, IEEE (2017).