非稳态时空预测

Memory In Memory: A Predictive Neural Network for Learning Higher-Order Non-Stationarity from Spatiotemporal Dynamics

一种从时空动力学中学习高阶非平稳性的预测神经网络

Yunbo Wang*, Jianjin Zhang*, Hongyu Zhu, Mingsheng Long (✉), Jianmin Wang, and Philip S. Yu

KLiss, MOE; BNRist; School of Software, Tsinghua University, China

Research Center for Big Data, Tsinghua University, China

Beijing Key Laboratory for Industrial Big Data System and Application

{wangyb15,zhang-jj16}@mails.tsinghua.edu.cn, {mingsheng,jimwang}@tsinghua.edu.cn

Abstract

Natural spatiotemporal processes can be highly non-stationary in many ways, e.g. the low-level non-stationarity such as spatial correlations or temporal dependencies of local pixel values; and the high-level variations such as the accumulation, deformation or dissipation of radar echoes in precipitation forecasting. From Cramér's Decomposition [4], any non-stationary process can be decomposed into deterministic, time-variant polynomials, plus a zero-mean stochastic term. By applying differencing operations appropriately, we may turn time-variant polynomials into a constant, making the deterministic component predictable.

*However, most previous recurrent neural networks for spatiotemporal prediction do not use the differential signals effectively, and their relatively simple state transition functions prevent them from learning too complicated variations in spacetime. We propose the **Memory In Memory (MIM)** networks and corresponding recurrent blocks for this purpose. The MIM blocks exploit the differential signals between adjacent recurrent states to model the non-stationary and approximately stationary properties in spatiotemporal dynamics with two cascaded, self-renewed memory modules. By stacking multiple MIM blocks, we could potentially handle higher-order non-stationarity. The MIM networks achieve the state-of-the-art results on four spatiotemporal prediction tasks across both synthetic and real-world datasets. We believe that the general idea of this work can be potentially applied to other time-series forecasting tasks.*

1. Introduction

Natural spatiotemporal processes exhibit complex non-stationarity in both space and time, where neighboring pixels exhibit local dependencies, and their joint distributions are

---
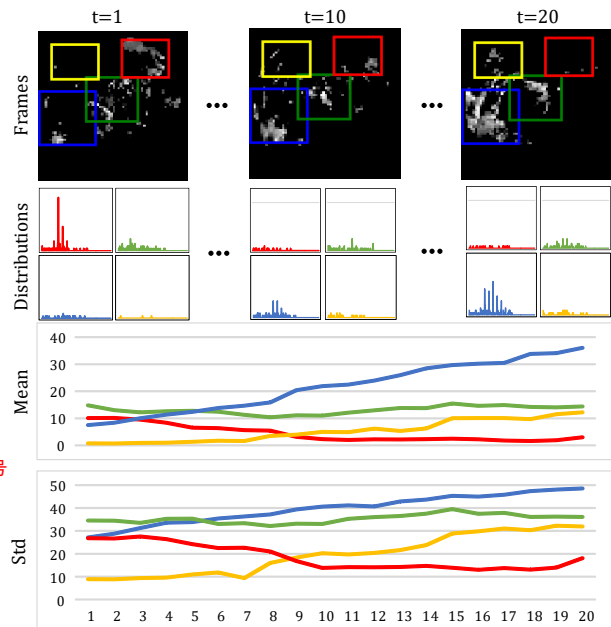*Equal contribution, in alphabetical order

Figure 1: An example of 20 consecutive radar maps to illustrate the complicated non-stationarity in precipitation forecasting. **First row:** radar maps, in which the whiter pixels show higher precipitation probability. **Second, third, last row:** pixel values' distributions, means and standard deviations for corresponding local regions that are identified by bounding boxes of different colors. Note that different regions have different variation trends, making the spatiotemporal prediction problem extremely challenging.

changing over time. Learning higher-order properties underlying the spatiotemporal non-stationarity is particularly significant for many video prediction tasks. Examples include modeling highly complicated real-world systems such as traffic flows [37, 35] and weather conditions [24, 32]. A well-performed predictive model is expected to learn the intrinsic variations in consecutive spatiotemporal context,

差分操作

MIM块利用相邻重复态之间的差分信号，通过两个级联的、自更新的记忆模块来模拟时空动力学中的非平稳和近似平稳特性

differential signals 差分信号

joint distributions 联合分布

高阶

连续时空环境中的内在变化

arXiv:1811.07490v2 [cs.LG] 8 Apr 2019

which can be seen as a combination of the stationary component and the deterministic non-stationary component.

A great challenge in non-stationary spatiotemporal prediction is how to effectively capture higher-order trends regarding each pixel and its local area. For example, when making precipitation forecasting, one should carefully consider the complicated and diverse local trends on the evolving radar maps, shown as Figure 1. But this problem is extremely difficult due to the complicated non-stationarity in both space and time. Most prior work handles trend-like non-stationarity with recursions of CNNs [37, 35] or relatively simple state transitions in RNNs [24, 32]. The lack of non-stationary modeling capability prevents reasoning about uncertainties in spatiotemporal dynamics and partially leads to the blurry effect of the predicted frames.

We attempt to resolve this problem by proposing a generic RNNs architecture that is more effective in non-stationarity modeling. We find that though the forget gates in the recurrent predictive models could deliver, select, and discard information in the process of memory state transitions, they are too simple to capture higher-order non-stationary trends in high-dimensional time series. In particular, the forget gates in the recent PredRNN model [32] does not work appropriately on precipitation forecasting: about 80% of them are saturated over all timestamps, implying almost time-invariant memory state transitions. In other words, future frames are predicted by approximately linear extrapolations.

In this paper, we focus on improving the memory transition functions of RNNs. Most statistical forecasting methods in classic time series analysis assume that the non-stationary trends can be rendered approximately stationary by performing suitable transformations such as *differencing*. We introduce this idea to RNNs and propose a new RNNs building block named *Memory In Memory* (MIM), which leverages the differential information between neighboring hidden states in the recurrent paths. MIM can be viewed as an improved version of LSTM [11], whose forget gate is replaced by another two embedded long short-term memories.

MIM has the following characteristics: **(1)** It creates unified modeling for the spatiotemporal non-stationarity by differencing neighboring hidden states rather than raw images. **(2)** By stacking multiple MIM blocks, our model has a chance to gradually stationarize the spatiotemporal process and make it more predictable. **(3)** Note that over-differencing is no good for time series prediction, as it may inevitably lead to a loss of information. This is another reason that we apply differencing in memory transitions rather than all recurrent signals, e.g. the input gate and the input modulation gate. **(4)** MIM has one memory cell adopted from LSTMs as well as two additional recurrent modules with their own memories embedded in the transition path of the first memory. We use these modules to respectively model the higher-order non-stationary and approximately stationary components of the spatiotemporal dynamics. The proposed MIM networks achieve the state-of-the-art results on multiple prediction tasks, including a widely used synthetic dataset and three real-world datasets.

2. Related Work

2.1. ARIMA Models for Time Series Forecasting

Our model is inspired by the Autoregressive Integrated Moving Average (ARIMA) models. A time-series random variable whose power spectrum remains constant over time can be viewed as a combination of signal and noise. An ARIMA model aims to separate the signal from the noise. The obtained signal is then extrapolated into the future. In theory, it tackles time series forecasting by transforming the non-stationary process to stationary through differencing [3].

2.2. Deterministic Spatiotemporal Prediction

Spatiotemporal non-stationary processes are more complicated, as the joint distribution of neighboring pixel values is varying in both space and time. Like low-dimensional time series, they can also be decomposed into deterministic and stochastic components. Recent work in neural networks explored spatiotemporal prediction from these two aspects.

CNNs [17] and RNNs [27] have been widely used for learning the deterministic spatial correlations and temporal dependencies from videos. Ranzato *et al.* [23] defined a recurrent model predicting frames in a discrete space of patch clusters. Srivastava *et al.* [26] introduced the sequence to sequence LSTM network from language modeling to video prediction. But this model can only capture temporal variations. To learn spatial and temporal variations in a unified network structure, Shi *et al.* [24] integrated the convolution operator into recurrent state transition functions, and proposed the Convolutional LSTM. Finn *et al.* [9] developed an action-conditioned video prediction model that can be further used in robotics planning when combined with the model predictive control methods. Villegas *et al.* [29] and Patraucean *et al.* [21] presented recurrent models based on the convolutional LSTM that leverage optical flow guided features. Kalchbrenner *et al.* [14] proposed the Video Pixel Network (VPN) that encodes the time, space, color structures of videos as a four-dimensional dependency chain. It achieves sharp prediction results but suffers from a high computational complexity. Wang *et al.* [32, 31] extended the convolutional LSTM with zigzag memory flows, which provides a great modeling capability for short-term video dynamics. Adversarial learning [10, 7] has been increasingly used in video generation or prediction [19, 30, 8, 28, 34], as it aims to solve the multi-modal training difficulty of the future prediction and helps generate less blurry frames.

However, the high-order non-stationarity of video dynamics has not been thoroughly considered by the above work,
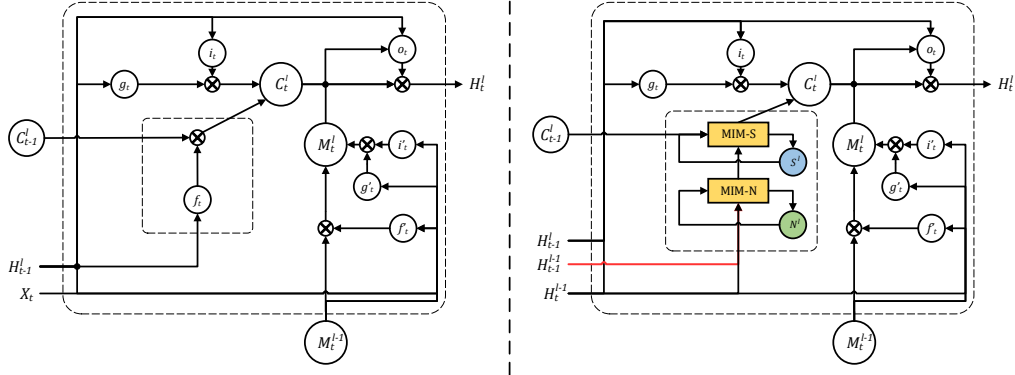
Figure 2: The ST-LSTM block [32] in the left plot and the proposed Memory In Memory (MIM) block in the right plot. MIM is designed to introduce two recurrent modules (yellow squares) to replace the forget gate (dashed box) in ST-LSTM. MIM-N is the non-stationary module and MIM-S is the stationary module. Note that the MIM block cannot be used in the first layer so the input $\mathcal{X}_t$ is replaced by $\mathcal{H}_t^{l-1}$.

whose temporal transition methods are relatively simple, either controlled by the recurrent gate structures or implemented by the recursion of the feed-forward network. By contrast, our model is characterized by exploiting high-order differencing to mitigate the non-stationary learning difficulty.

2.3. Stochastic Spatiotemporal Prediction

Some recent methods [36, 6, 18] attempted to model the stochastic component of video dynamics using Variational Autoencoder [16]. These methods increase the prediction diversity but are difficult to evaluate and require to run a great number of times for a satisfactory result. In this paper, we focus on the deterministic part of spatiotemporal non-stationarity. More specifically, this work attempts to stationarize the complicated spatiotemporal processes and make their deterministic components in the future more predictable by proposing new RNNs architecture for non-stationarity.

3. Methods

As mentioned above, the spatiotemporal non-stationarity remains under-explored and its differential features have not been fully exploited by previous methods using neural networks. In this section, we first present the *Memory In Memory* (MIM) blocks for learning about the higher-order non-stationarity from RNNs memory transitions. We then discuss a new RNN architecture, which interlinks multiple MIM blocks with diagonal state connections, for modeling the differential information in the spatiotemporal prediction. By stacking multiple MIM blocks, we could potentially learn higher-order non-stationarity from spatiotemporal dynamics. The proposed MIM state transition approach can be integrated into all LSTM-like units. We choose the Spatiotemporal LSTM (ST-LSTM) [32] as our base network for a trade-off between prediction accuracy and computation simplicity. ST-LSTM is characterized by a dual-memory structure, $\mathcal{C}_t^l$ and $\mathcal{M}_t^l$, as shown in Figure 2 (left). The corre-

sponding zigzag memory flow of $\mathcal{M}_t^l$, as illustrated by the black arrows in Figure 4, strengthens its short-term modeling capability by increasing the recurrent transition depth.

3.1. Memory In Memory Blocks

We observe that the complex dynamics in spatiotemporal sequences can be handled more effectively as a combination of stationary variations and non-stationary variations. Suppose we have a video sequence showing a person walking at a constant speed. The velocity can be seen as a stationary variable and the swing of the legs should be considered as a non-stationary process, which is apparently more difficult to predict. Unfortunately, the forget gate in previous LSTM-like models is a simple gating structure that struggles to capture the non-stationary variations in spacetime. In preliminary experiments, we find that the majority of forget gates in the recent PredRNN model [32] are saturated, implying that the units always remember stationary variations.

The Memory In Memory (MIM) block is enlightened by the idea of modeling the non-stationary variations using a series of cascaded memory transitions instead of the simple, saturation-prone forget gate in ST-LSTM. As compared in Figure 2 (the smaller dashed boxes), two cascaded temporal memory recurrent modules are designed to replace the temporal forget gate f_t in ST-LSTM. The first module additionally taking $\mathcal{H}_{t-1}^{l-1}$ as input is used to capture the non-stationary variations based on the differencing $(\mathcal{H}_t^{l-1} - \mathcal{H}_{t-1}^{l-1})$ between two consecutive hidden representations. So we name it the **non-stationary module** (shown as **MIM-N** in Figure 3). It generates differential features $\mathcal{D}_t^l$ based on the difference-stationary assumption [22]. The other recurrent module takes as inputs the output $\mathcal{D}_t^l$ of the MIM-N module and the outer temporal memory $\mathcal{C}_{t-1}^l$ to capture the approximately stationary variations in spatiotemporal sequences. So we call it the **stationary module** (shown as **MIM-S** in Figure 3). By replacing the forget gate with the
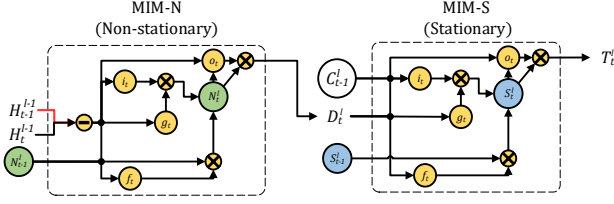
Figure 3: The non-stationary module (MIM-N) and the stationary module (MIM-S), which are interlinked in a cascaded structure in the MIM block. Non-stationarity is modeled by differencing.

非平稳动力学可以有效的被捕捉到

final output $\mathcal{T}_t^l$ of the cascaded non-stationary and stationary modules (as shown in Figure 2), the non-stationary dynamics can be captured more effectively. Key calculations inside a **MIM** block can be shown as follows:

$$
\begin{aligned}
g_t &= \tanh(W_{xg} * \mathcal{H}_t^{l-1} + W_{hg} * \mathcal{H}_{t-1}^l + b_g) \\
i_t &= \sigma(W_{xi} * \mathcal{H}_t^{l-1} + W_{hi} * \mathcal{H}_{t-1}^l + b_i) \\
\mathcal{D}_t^l &= \text{MIM-N}(\mathcal{H}_t^{l-1}, \mathcal{H}_{t-1}^{l-1}, \mathcal{N}_{t-1}^l) \\
\mathcal{T}_t^l &= \text{MIM-S}(\mathcal{D}_t^l, \mathcal{C}_{t-1}^l, \mathcal{S}_{t-1}^l) \\
\mathcal{C}_t^l &= \mathcal{T}_t^l + i_t \odot g_t \\
g_t' &= \tanh(W_{xg}' * \mathcal{H}_t^{l-1} + W_{mg} * \mathcal{M}_t^{l-1} + b_g') \\
i_t' &= \sigma(W_{xi}' * \mathcal{H}_t^{l-1} + W_{mi} * \mathcal{M}_t^{l-1} + b_i') \\
f_t' &= \sigma(W_{xf}' * \mathcal{H}_t^{l-1} + W_{mf} * \mathcal{M}_t^{l-1} + b_f') \\
\mathcal{M}_t^l &= f_t' \odot \mathcal{M}_t^{l-1} + i_t' \odot g_t' \\
o_t &= \sigma(W_{xo} * \mathcal{H}_t^{l-1} + W_{ho} * \mathcal{H}_{t-1}^l + W_{co} * \mathcal{C}_t^l + W_{mo} * \mathcal{M}_t^l + b_o) \\
\mathcal{H}_t^l &= o_t \odot \tanh(W_{1\times1} * [\mathcal{C}_t^l, \mathcal{M}_t^l]),
\end{aligned}
\tag{1}
$$

where $\mathcal{S}$ and $\mathcal{N}$ denote the horizontally-transited memory cells in the non-stationary module (MIM-N) and stationary module (MIM-S) respectively; $\mathcal{D}$ is the differential features learned by MIM-N and fed into MIM-S; $\mathcal{T}$ is the memory passing the virtual "forget gate"; and $*$ denotes convolution.

The cascaded structure enables end-to-end modeling of different orders of non-stationary dynamics. It is based on the difference-stationary assumption that differencing a non-stationary process repeatedly will likely lead to a stationary one [22]. A schematic of MIM-N and MIM-S is presented in Figure 3. We present the detailed calculations of **MIM-N** as follows:

$$
\begin{aligned}
g_t &= \tanh(W_{xg} * (\mathcal{H}_t^{l-1} - \mathcal{H}_{t-1}^{l-1}) + W_{ng} * \mathcal{N}_{t-1}^l + b_g) \\
i_t &= \sigma(W_{xi} * (\mathcal{H}_t^{l-1} - \mathcal{H}_{t-1}^{l-1}) + W_{ni} * \mathcal{N}_{t-1}^l + b_i) \\
f_t &= \sigma(W_{xf} * (\mathcal{H}_t^{l-1} - \mathcal{H}_{t-1}^{l-1}) + W_{nf} * \mathcal{N}_{t-1}^l + b_f) \\
\mathcal{N}_t^l &= f_t \odot \mathcal{N}_{t-1}^l + i_t \odot g_t \\
o_t &= \sigma(W_{xo} * (\mathcal{H}_t^{l-1} - \mathcal{H}_{t-1}^{l-1}) + W_{no} * \mathcal{N}_t^l + b_o) \\
\mathcal{D}_t^l &= \text{MIM-N}(\mathcal{H}_t^{l-1}, \mathcal{H}_{t-1}^{l-1}, \mathcal{N}_{t-1}^l) = o_t \odot \tanh(\mathcal{N}_t^l),
\end{aligned}
\tag{2}
$$

where all gates g_t, i_t, f_t and o_t are updated by incorporating the frame difference $(\mathcal{H}_t^{l-1} - \mathcal{H}_{t-1}^{l-1})$, which highlights the

non-stationary variations in the spatiotemporal sequence. The detailed calculations of **MIM-S** are shown as follows:

$$
\begin{aligned}
g_t &= \tanh(W_{dg} * \mathcal{D}_t^l + W_{cg} * \mathcal{C}_{t-1}^l + b_g) \\
i_t &= \sigma(W_{di} * \mathcal{D}_t^l + W_{ci} * \mathcal{C}_{t-1}^l + b_i) \\
f_t &= \sigma(W_{df} * \mathcal{D}_t^l + W_{cf} * \mathcal{C}_{t-1}^l + b_f) \\
\mathcal{S}_t^l &= f_t \odot \mathcal{S}_{t-1}^l + i_t \odot g_t \\
o_t &= \sigma(W_{do} * \mathcal{D}_t^l + W_{co} * \mathcal{C}_{t-1}^l + W_{so} * \mathcal{S}_t^l + b_o) \\
\mathcal{T}_t^l &= \text{MIM-S}(\mathcal{D}_t^l, \mathcal{C}_{t-1}^l, \mathcal{S}_{t-1}^l) = o_t \odot \tanh(\mathcal{S}_t^l),
\end{aligned}
\tag{3}
$$

which takes the memory cells $\mathcal{C}_{t-1}^l$ and the differential features $\mathcal{D}_t^l$ generated by MIM-N as input. As can be validated, the stationary module provides a gating mechanism to adaptively decide whether to trust the original memory $\mathcal{C}_{t-1}^l$ or the differential features $\mathcal{D}_t^l$. If the differential features vanish, indicating that the non-stationary dynamics is not prominent, then MIM-S will mainly reuse the original memory. Otherwise, if the differential features are prominent, then MIM-S will overwrite the original memory and focus more on the non-stationary dynamics.

3.2. Memory In Memory Networks

Stacking multiple MIM blocks, our model has a chance to capture **higher orders of non-stationarity**, gradually stationarizes the spatiotemporal process and makes the future sequence more predictable. The key idea of this architecture is to deliver necessary hidden states for generating differential features and best facilitating non-stationarity modeling.

A schematic of our proposed diagonal recurrent architecture is shown in Figure 4. We deliver the hidden states $\mathcal{H}_{t-1}^{l-1}$ and $\mathcal{H}_t^{l-1}$ to the *Memory In Memory* (MIM) block at timestamp $t \neq 1$ and layer $l \neq 1$ to generate the differenced features for further use. These connections are shown as diagonal arrows in Figure 4. As the first layer does not have any previous layer, we simply use the Spatiotemporal LSTM (ST-LSTM) [32] to generate its hidden presentations. Note that, the temporal differencing is performed by subtracting hidden state $\mathcal{H}_t^{l-1}$ from the hidden state $\mathcal{H}_{t-1}^{l-1}$ in MIM. Compared to differencing neighboring raw images directly, differencing temporally adjacent hidden states can reveal the non-stationarity more evidently, as the spatiotemporal variations in local areas have been encoded into the hidden representations through the bottom ST-LSTM layer.

与直接对相邻的原始图像进行差分相比，对时间上相邻的隐藏状态进行差分可以更明显地揭示非稳态性

Another distinctive feature of the MIM networks resides in the horizontal state transition paths. As the MIM blocks have two cascaded temporal memory modules to capture the non-stationary and stationary dynamics respectively, we further deliver the two temporal memories (denoted by $\mathcal{N}$ for the *non-stationary* memory and by $\mathcal{S}$ for the *stationary* memory) along the blue arrows in Figure 4.

The MIM networks generate one frame at one timestamp. Calculations of the entire model with one ST-LSTM and
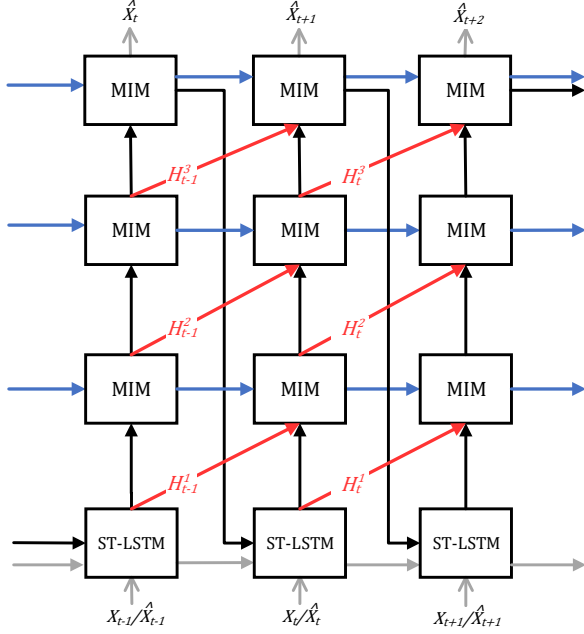
Figure 4: A MIM network with three MIMs and one ST-LSTM. **Red arrows:** the *diagonal* state transition paths of $\mathcal{H}$ for differential modeling. **Blue arrows:** the *horizontal* transition paths of the memory cells $\mathcal{C}$, $\mathcal{N}$ and $\mathcal{S}$. **Black arrows:** the *zigzag* state transition paths of $\mathcal{M}$. **Input:** the input can be either the ground truth frame for input sequence, or the generated frame at previous timestamp. **Output:** one frame is generated at each timestamp.

$(L-1)$ MIMs can be presented as follows (for $2 \leq l \leq L$). Note that there is no MIM block that is marked as MIM_1.

$$\mathcal{H}_t^1, \mathcal{C}_t^1, \mathcal{M}_t^1 = \text{ST-LSTM}_1(\mathcal{X}_t, \mathcal{H}_{t-1}^1, \mathcal{C}_{t-1}^1, \mathcal{M}_{t-1}^L)$$
$$\mathcal{H}_t^l, \mathcal{C}_t^l, \mathcal{M}_t^l, \mathcal{N}_t^l, \mathcal{S}_t^l = \text{MIM}_l(\mathcal{H}_t^{l-1}, \mathcal{H}_{t-1}^l, \mathcal{C}_{t-1}^l, \mathcal{M}_t^{l-1}, \quad (4)$$
$$\mathcal{N}_{t-1}^l, \mathcal{S}_{t-1}^l).$$

We formulate high-order non-stationarity as high-order polynomials based on Cramér's Decomposition [4]. In the space-time contexts, it refers to the varying trends of statistics of pixel values. The order of non-stationary polynomials can be reduced by a couple of differencing operations. We blend this idea from time-series analysis with deep learning: stacking MIMs with differential inputs layer by layer.

4. Experiments

In this section, we evaluate the proposed MIM model using four datasets for spatiotemporal prediction: a synthetic dataset with moving digits, a real traffic flow dataset, a real radar echo dataset, and a human action dataset. Here are some common settings all over these datasets. Our model has four layers in all experiments, including one ST-LSTM layer as the first layer and three MIMs. The number of feature channels in each MIM block is 64, as a trade-off of prediction
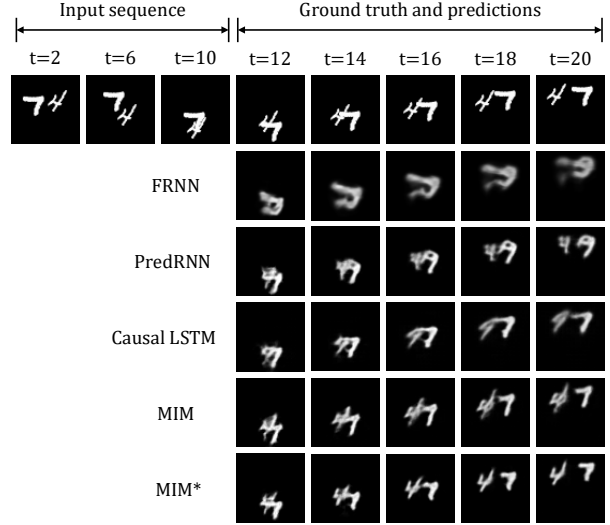

Figure 5: Prediction examples on the standard Moving MNIST. All models predict 10 frames into the future by observing 10 previous frames. The output frames are shown at two frames intervals.

accuracy and memory efficiency. All models are trained with the ℓ_2 loss, using the ADAM optimizer [15] with a learning rate of 0.001. The mini-batch size is set to 8. We apply the layer normalization [1] to the compared models in order to reduce the covariate shift problem [12]. Besides, we apply the scheduled sampling [2] to all models to stitch the discrepancy between training and inference. Code and models are available at https://github.com/Yunbo426/MIM.

4.1. Moving MNIST

The standard Moving MNIST is a synthetic dataset with grayscale image sequences of flying digits. We follow exactly the experimental settings in PredRNN [32].

We visualize a sequence of predicted frames on the standard Moving MNIST test set in Figure 5. This example is challenging, as severe occlusions exist near the junction of the input sequence and the output sequence. The occlusions can be viewed as information bottleneck, in which the mean and variance of the spatiotemporal process meet drastic changes, indicating the presence of a high-order non-stationarity. The generated images of MIM are more satisfactory, less blurry than those of other models. Actually, we cannot even tell the digits in the last frames generated by other models. We may conclude that MIM shows more capability in capturing complicated non-stationary variations.

We use the per-frame structural similarity index measure (SSIM) [33], the mean square error (MSE) and the mean absolute error (MAE) to evaluate our models. A lower MSE or MAE, or a higher SSIM indicates a better prediction. As shown in Table 1, our proposed MIM model approaches the state-of-the-art results on the standard Moving MNIST

Model	SSIM	MSE	MAE
FC-LSTM [26]	0.690	118.3	209.4
ConvLSTM [24]	0.707	103.3	182.9
TrajGRU [25]	0.713	106.9	190.1
CDNA [9]	0.721	97.4	175.3
DFN [5]	0.726	89.0	172.8
FRNN [20]	0.813	69.7	150.3
VPN baseline [14]	0.870	64.1	131.0
PredRNN [32]	0.867	56.8	126.1
Causal LSTM [31]	0.898	46.5	106.8
MIM	0.874	52.0	116.5
MIM*	**0.910**	**44.2**	**101.1**

Table 1: A comparison for predicting 10 frames on Moving MNIST dataset. All models have comparable numbers of parameters. **MIM*** is network using Causal LSTM [31] as the first layer, and integrating the cascaded MIM-N and MIM-S modules into the Causal LSTM memory cells. This result shows that MIM is a generic mechanism for improving recurrent memory transitions.

dataset. In particular, we construct another model named MIM* by using Causal LSTM [31] as the first layer, and integrating the cascaded MIM-N and MIM-S modules into the Causal LSTM memory cells, using them to replace the temporal forget gates in Causal LSTMs. This result shows that the *memory in memory* mechanism is not specifically designed for the ST-LSTM; instead, it is a generic mechanism for improving RNNs memory transitions. Though in other parts of this paper, we use ST-LSTM as our base structure for a trade-off between prediction accuracy and computational complexity, we can see that MIM performs better than its ST-LSTM (PredRNN) baseline, while MIM* also performs better than its Causal LSTM baseline. We also adopt the gradient-based sharpness metric from [19] to measure the sharpness of the generated images. As shown in Table 2, MIM rises the sharpness score by 16% over PredRNN.

Model	Gradient-Based Sharpness
FRNN [20]	24.99
PredRNN [32]	23.29
MIM	**27.05**

Table 2: Sharpness evaluation on Moving MNIST.

验证MIM-N和MIM-S 的作用

We further testify **the necessity of cascading inner recurrent modules** by respectively removing the stationary modules or non-stationary modules. As illustrated in Table 3, the MIM network without MIM-N works slightly better than that without MIM-S. Also, either of them has significant improvements over the PredRNN model in MSE/MAE, showing the necessity of cascading them in a unified network. When MIM-N and MIM-S are interlinked, the entire MIM model achieves the best performance.

Model	SSIM	MSE	MAE
MIM (without MIM-N)	0.858	54.4	124.8
MIM (without MIM-S)	0.853	55.7	125.5
MIM	**0.874**	**52.0**	**116.5**

Table 3: Ablation study with respect to the MIM block.

MIM层数实验

We study **the sensitivity of our model to the number of MIM blocks**. As stacking 2–4 recurrent layers is a common practice, we evaluate models with $2/3/4$ MIMs on Moving MNIST. The proposed model with 3 MIMs performs best. It is a trade-off: applying too few MIMs leads to inadequate non-stationary modeling capability while the excessively deep recurrent model leads to training difficulty.

# MIM blocks	SSIM	MSE
2	0.870	54.6
3	**0.874**	**52.0**
4	0.859	53.7

Table 4: The sensitivity of our model to the number of MIM blocks.

遗忘门饱和实验

The forget gates are easily saturated in PredRNN as well as other LSTM-based spatiotemporal prediction models such as ConvLSTM [24]. As shown in Figure 6, a great portion of forget gates of these models are close to 0, indicating that the *long short-term memories* do not work properly. As the non-stationary signals within short-term variations are hard to be captured, these models have to refresh the memory states to convey these short-term hidden representations. Our model mitigates this problem and makes better use of 减轻缓和 long-term variations through the proposed MIM block. As MIM-N mainly reduces the non-stationarity, short-term tendencies become easier to be captured and more neurons in MIM-S can be used for handling long-term variations.
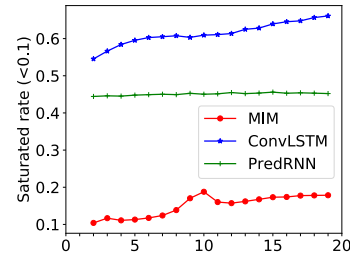

Figure 6: The saturated rate of forget gates that corresponds to $|\mathcal{T}_t/\mathcal{C}_{t-1}| < 0.1$ for MIM, and $f_t < 0.1$ for other models.

4.2. TaxiBJ Traffic Flow

Traffic flows are collected from the chaotic real-world environment. They will not vary uniformly over time, and there are strong temporal dependencies between the traffic conditions at neighboring timestamps. Each frame in TaxiBJ

is a $32 \times 32 \times 2$ grid of image. Two channels represent the traffic flow entering and leaving the same district. We follow the experimental settings of ST-ResNet [37], which yields the previous state-of-the-art results on this dataset. Each sequence contains 8 consecutive frames, 4 for the inputs and 4 for the predictions. We show the quantitative results in Table 5 and the qualitative results in Figure 7. To make the comparisons conspicuous, we also visualize the difference between the predictions and the ground truth images. Obviously, MIM shows the best performance in all predicted frames among all compared models, with the lowest difference intensities.

Model	Frame 1	Frame 2	Frame 3	Frame 4
ST-ResNet [37]	0.688	0.939	1.130	1.288
VPN [14]	0.744	1.031	1.251	1.444
FRNN [20]	0.682	0.823	0.989	1.183
PredRNN [32]	0.634	0.934	1.047	1.263
Causal LSTM [31]	0.641	0.855	0.979	1.158
MIM	**0.554**	**0.737**	**0.887**	**0.999**

Table 5: Per-frame MSE on the TaxiBJ dataset. All compared models take 4 historical traffic flow images as inputs, and predict the next 4 images (traffic conditions for the next two hours).
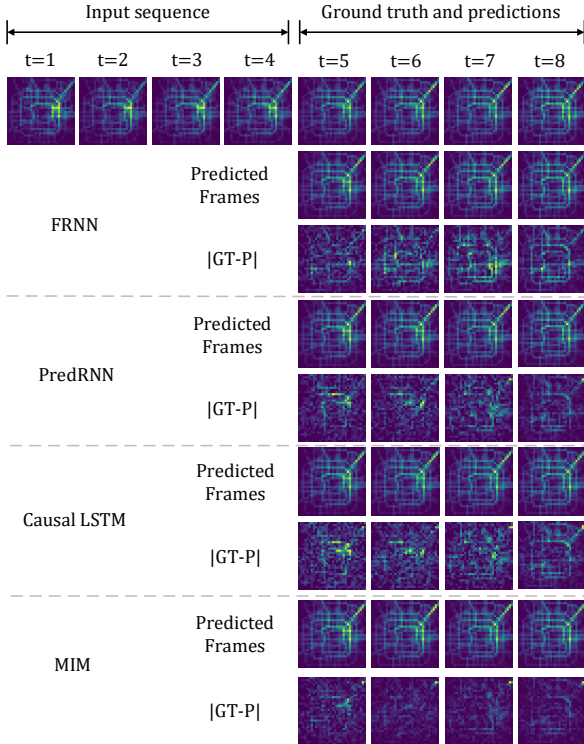

Figure 7: Prediction examples on TaxiBJ dataset. For ease of comparison, we also visualize the difference between the ground truth frames (GT) and predicted frames (P).

4.3. Radar Echo

The radar echo dataset contains evolving radar maps that were collected every 6 minutes, from May 1st, 2014 to June 30th, 2014. Each frame is a $64 \times 64 \times 1$ grid of image, covering 64×64 square kilometers. We predict 10 timestamps into the future at a time interval of 6 minutes, covering the next hour. We visualize the generated radar maps in Figure 8. We can see that the evolution of radar echoes is a highly non-stationary process. The accumulation, deformation, and dissipation of the radar echoes are happening at every moment. In this showcase, the echoes in the bottom left corner aggregate while those in the upper right corner dissipate. Only MIM captures the movement of the echoes correctly.
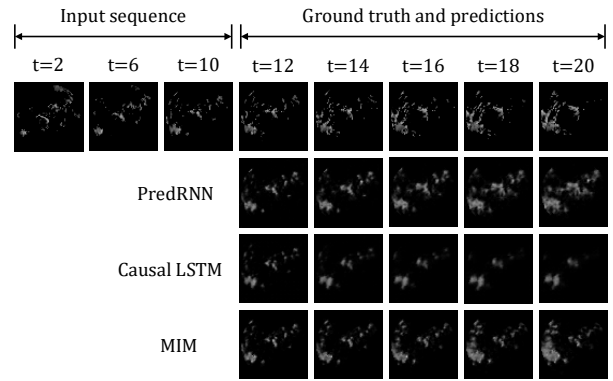

Figure 8: Examples of the next-hour predictions of radar echoes, where higher pixel values indicate higher precipitation probabilities.

Model	MSE	CSI-30	CSI-40	CSI-50
FRNN [20]	52.5	0.254	0.203	0.163
PredRNN [32]	31.8	0.401	0.378	0.306
Causal LSTM [31]	29.8	0.362	0.331	0.251
MIM	**27.8**	**0.429**	**0.399**	**0.317**

Table 6: A comparison for predicting 10 frames on the subsets of the radar dataset. All of the models are also trained with 10 target frames and made to predict 10 future frames at test time.

We evaluate the generated radar echoes by MSE in Table 6, and then convert pixel values to radar echo intensities in dBZ. We respectively choose 30 dBZ, 40 dBZ and 50 dBZ as thresholds to calculate the Critical Success Index (CSI). CSI is defined as $\text{CSI} = \frac{\text{hits}}{\text{hits}+\text{misses}+\text{falsealarms}}$, where hits corresponds to *true positive*, misses corresponds to *false positive*, and false alarms corresponds to *false negative*. A higher CSI denotes a better prediction result. MIM consistently outperforms other models in both MSE and CSI. Figure 9 shows the frame-wise MSE/CSI. CSI-40 and CSI-50 indicate the probabilities of severe weather conditions. MIM performs best though predicting severe weather is non-trivial due to
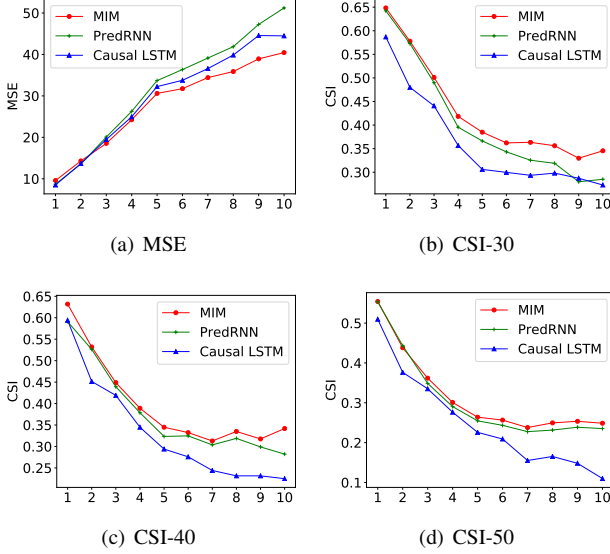
Figure 9: Frame-wise comparisons of the next 10 generated radar maps. Lower MSE curves or higher CSI curves indicate better forecasting results. The MIM network is the most high-performing method over all timestamps in the forecasting horizon.
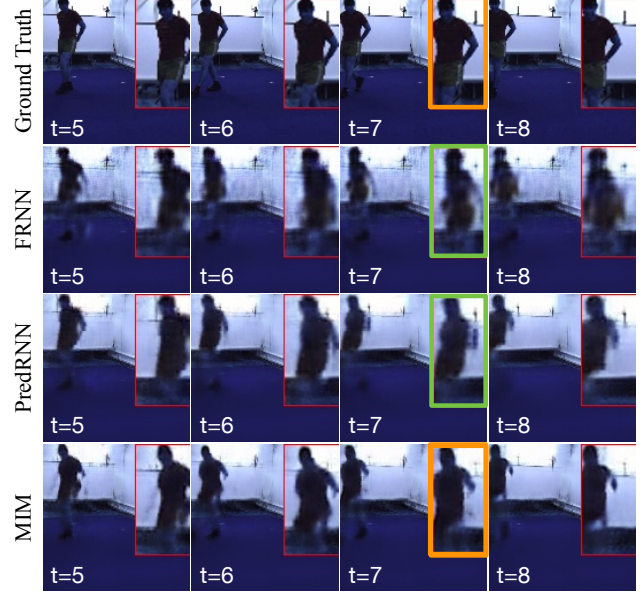

Figure 10: Examples of the generated images on the Human3.6M dataset. We zoom in to show the details of images.

the long tail distributions of the pixel values.

4.4. Human3.6M

The Human3.6M dataset [13] contains human actions of 17 scenarios, including 3.6 million poses and corresponding images. We train the models using only the "Walking" scenario. The RGB images in Human3.6M dataset are originally $1000 \times 1000 \times 3$, and resized to $128 \times 128 \times 3$ in our experiments. We generate 4 future frames given the previous 4 RGB frames. We use subjects S1, S5, S6, S7, S8 for training, and S9, S11 for testing.

As shown in Table 7 and Figure 10, the MIM network outperforms the previous state-of-the-art models in both numerical metrics and visual effects. The generated frames by MIM are more accurate in motion positions, as indicated by the orange boxes (versus the green boxes). We notice that some work performs well on human-body datasets by modeling the structures of human joints, while the MIM network is designed for non-structural, general-purpose spacetime data such as RGB videos. The structural and non-structural methods can be integrated and jointly used in a complementary manner, which is left for future work.

Model	SSIM	MSE	MAE
FRNN [20]	0.771	497.7	1901.1
PredRNN [32]	0.781	484.1	1895.2
MIM	**0.790**	**429.9**	**1782.8**

Table 7: Quantitative results on the Human3.6M dataset.

5. Conclusions

We investigate the underlying non-stationarity that forms one of the main obstacles in spatiotemporal prediction. Existing LSTM-based models for spatiotemporal prediction are powerful in modeling difference-stationary sequences, whose capability for modeling high-order non-stationary process is limited by their relatively weak forget gates. This paper proposes a new recurrent neural network to enable non-stationary modeling in the spacetime contexts. We formulate high-order non-stationarity as high-order polynomials with respect to the statistically varying trends of pixel intensities. The order of non-stationary polynomials can be reduced by a couple of differencing operations. We leverage this idea from time-series analysis: stacking MIMs with differential inputs layer by layer. The Memory In Memory (MIM) block is derived to model the complicated variations, which uses two cascaded recurrent modules to handle the non-stationary and approximately stationary components in the spatiotemporal dynamics. MIM achieves the state-of-the-art prediction performance on four datasets: a synthetic dataset of flying digits, a traffic flow prediction dataset, a weather forecasting dataset, and a human pose video dataset.

Acknowledgements

References

[1] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[2] Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. Scheduled sampling for sequence prediction with recurrent neural networks. In *NeurIPS*, pages 1171–1179, 2015.

[3] George EP Box, Gwilym M Jenkins, Gregory C Reinsel, and Greta M Ljung. *Time series analysis: forecasting and control.* John Wiley & Sons, 2015.

[4] Harald Cramér. On some classes of nonstationary stochastic processes. In *Berkeley Symposium on Mathematical Statistics and Probability*, volume 2, pages 57–78. University of Los Angeles Press Berkeley and Los Angeles, 1961.

[5] Bert De Brabandere, Xu Jia, Tinne Tuytelaars, and Luc Van Gool. Dynamic filter networks. In *NeurIPS*, 2016.

[6] Emily Denton and Rob Fergus. Stochastic video generation with a learned prior. In *ICML*, pages 1174–1183, 2018.

[7] Emily L Denton, Soumith Chintala, Rob Fergus, et al. Deep generative image models using a laplacian pyramid of adversarial networks. In *NeurIPS*, pages 1486–1494, 2015.

[8] Emily L Denton et al. Unsupervised learning of disentangled representations from video. In *NeurIPS*, 2017.

[9] Chelsea Finn, Ian Goodfellow, and Sergey Levine. Unsupervised learning for physical interaction through video prediction. In *NeurIPS*, 2016.

[10] Ian J. Goodfellow, Jean Pougetabadie, Mehdi Mirza, Bing Xu, David Wardefarley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. In *NeurIPS*, volume 3, pages 2672–2680, 2014.

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

[12] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, 2015.

[13] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(7):1325–1339, jul 2014.

[14] Nal Kalchbrenner, Aaron van den Oord, Karen Simonyan, Ivo Danihelka, Oriol Vinyals, Alex Graves, and Koray Kavukcuoglu. Video pixel networks. In *ICML*, 2017.

[15] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

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

[17] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NeurIPS*, pages 1097–1105, 2012.

[18] Alex X Lee, Richard Zhang, Frederik Ebert, Pieter Abbeel, Chelsea Finn, and Sergey Levine. Stochastic adversarial video prediction. *arXiv preprint arXiv:1804.01523*, 2018.

[19] Michael Mathieu, Camille Couprie, and Yann LeCun. Deep multi-scale video prediction beyond mean square error. In *ICLR*, 2016.

[20] Marc Oliu, Javier Selva, and Sergio Escalera. Folded recurrent neural networks for future video prediction. In *ECCV*, 2018.

[21] Viorica Patraucean, Ankur Handa, and Roberto Cipolla. Spatio-temporal video autoencoder with differentiable memory. In *ICLR Workshop*, 2016.

[22] Donald B. Percival and Andrew T. Walden. *Spectral Analysis for Physical Applications.* Cambridge University Press, 1993.

[23] MarcAurelio Ranzato, Arthur Szlam, Joan Bruna, Michael Mathieu, Ronan Collobert, and Sumit Chopra. Video (language) modeling: a baseline for generative models of natural videos. *arXiv preprint arXiv:1412.6604*, 2014.

[24] 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 *NeurIPS*, 2015.

[25] Xingjian Shi, Zhihan Gao, Leonard Lausen, Hao Wang, Dit-Yan Yeung, Wai-kin Wong, and Wang-chun Woo. Deep learning for precipitation nowcasting: A benchmark and a new model. In *NeurIPS*, 2017.

[26] Nitish Srivastava, Elman Mansimov, and Ruslan Salakhutdinov. Unsupervised learning of video representations using lstms. In *ICML*, 2015.

[27] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks. In *NeurIPS*, 2014.

[28] Sergey Tulyakov, Ming-Yu Liu, Xiaodong Yang, and Jan Kautz. Mocogan: Decomposing motion and content for video generation. In *CVPR*, 2018.

[29] Ruben Villegas, Jimei Yang, Seunghoon Hong, Xunyu Lin, and Honglak Lee. Decomposing motion and content for natural video sequence prediction. In *ICLR*, 2017.

[30] Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. Generating videos with scene dynamics. In *NeurIPS*, 2016.

[31] 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 *ICML*, pages 5123–5132, 2018.

[32] Yunbo Wang, Mingsheng Long, Jianmin Wang, Zhifeng Gao, and S Yu Philip. Predrnn: Recurrent neural networks for predictive learning using spatiotemporal lstms. In *NeurIPS*, 2017.

[33] Zhou Wang, A. C Bovik, H. R Sheikh, and E. P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600, 2004.

[34] Nevan Wichers, Ruben Villegas, Dumitru Erhan, and Honglak Lee. Hierarchical long-term video prediction without supervision. In *ICML*, 2018.

[35] Ziru Xu, Yunbo Wang, Mingsheng Long, Jianmin Wang, and MOE KLiss. Predcnn: Predictive learning with cascade convolutions. In *IJCAI*, 2018.

[36] Tianfan Xue, Jiajun Wu, Katherine Bouman, and Bill Freeman. Visual dynamics: Probabilistic future frame synthesis via cross convolutional networks. In *NeurIPS*, pages 91–99, 2016.

[37] Junbo Zhang, Yu Zheng, and Dekang Qi. Deep spatiotemporal residual networks for citywide crowd flows prediction. In *AAAI*, 2017.