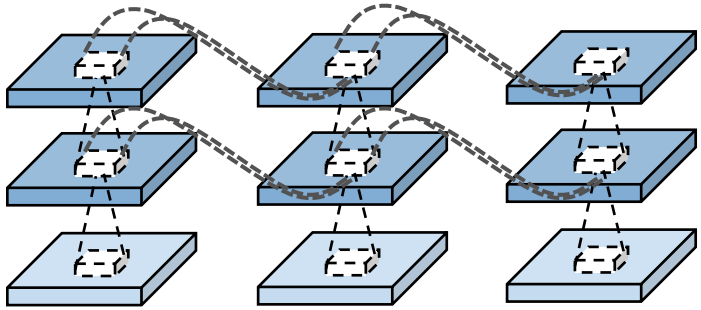
Predictive Learning

for Video Prediction

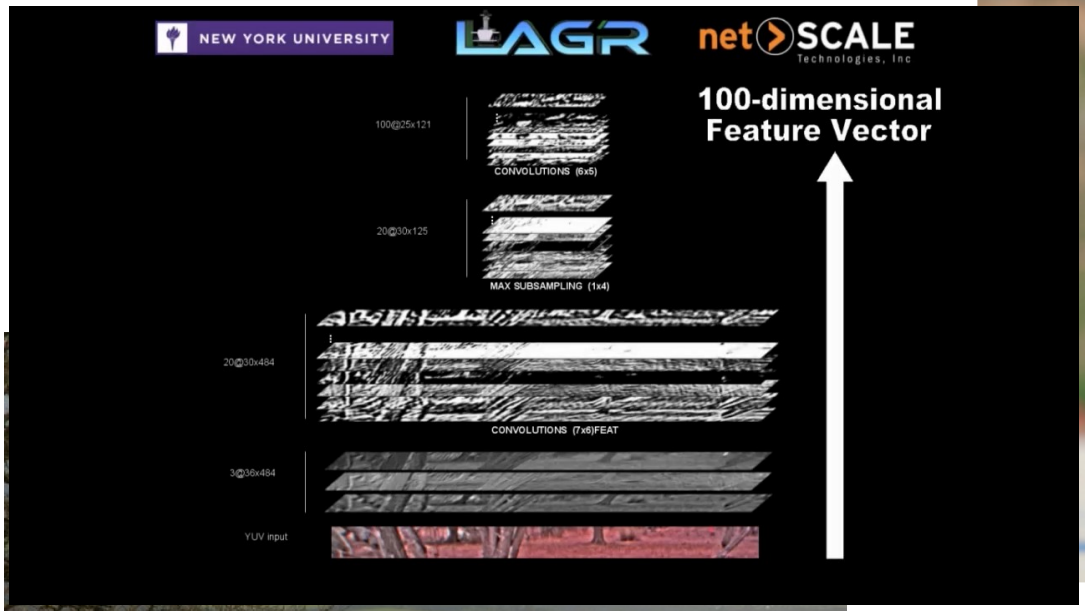
Mingsheng Long

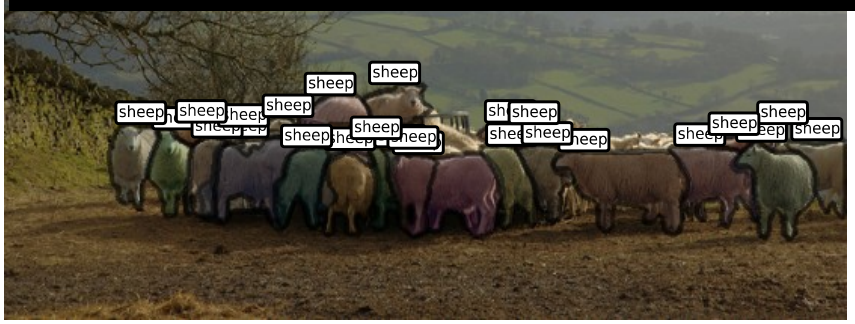
Tsinghua University

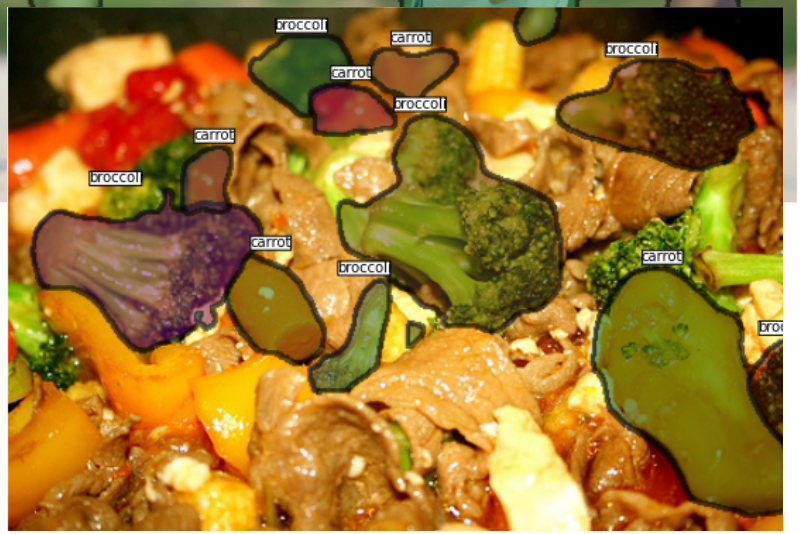


Today's AI

Today, AI = Supervised (Deep) Learning



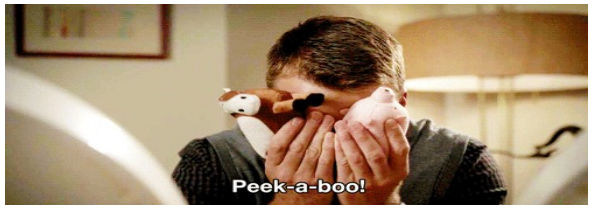




"Real" AI

But Supervised Learning is Insufficient for "Real" AI

- **Most of animals and humans learning is unsupervised, through interaction with the world**
- **We learn how the world works by observing it**
  - We learn many simple things: depth and 3-dimensionality, gravity, object permanence, …
- **We build models of the world through predictive unsupervised learning**
- **World models give us "common sense"**



Predictive Learning

How Much Information does the Machine Need to Predict?

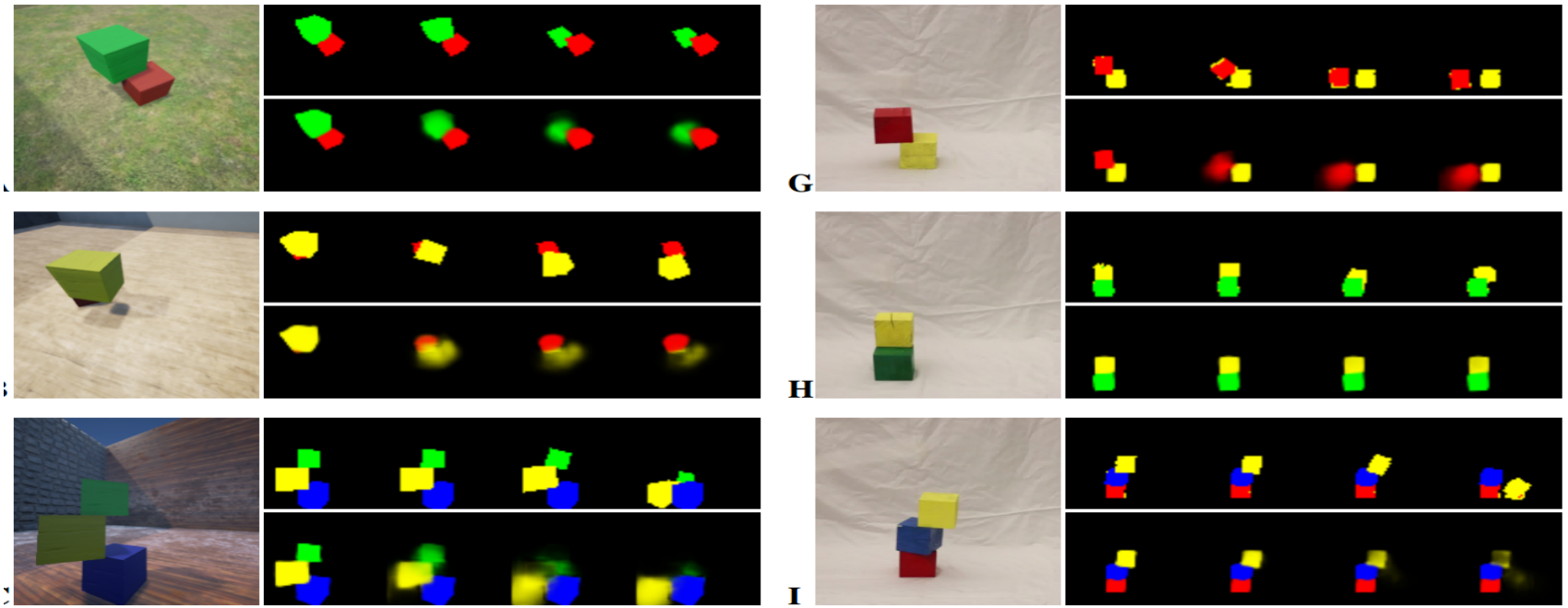
- **Pure Reinforcement Learning (cherry)**
  - The machine predicts a scalar reward given once in a while.
  - **A few bits for some samples**
- **Supervised Learning (icing)**
  - The machine predicts a category or a few numbers for each input
  - **10→10,000 bits per sample**
- **Unsupervised/Predictive Learning (génoise)**
  - The machine predicts any part of its input for any observed part.
  - Predicts future frames in videos
  - **Millions of bits per sample**

- **Unsupervised Learning is the Dark Matter (or Dark Energy) of AI**

Video Prediction: Learning Physics

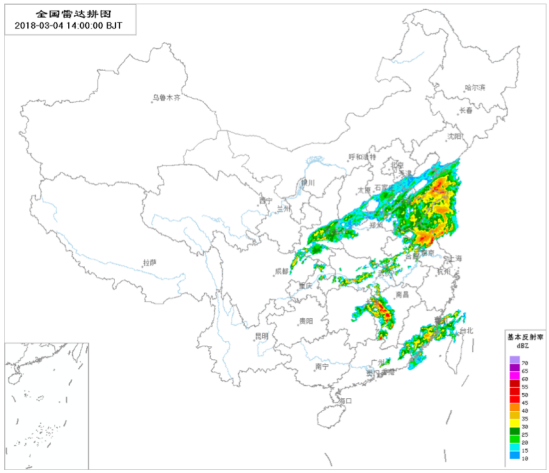
Learning Physics (PhysNet) [Lerer, Gross, Fergus, ICML'16]

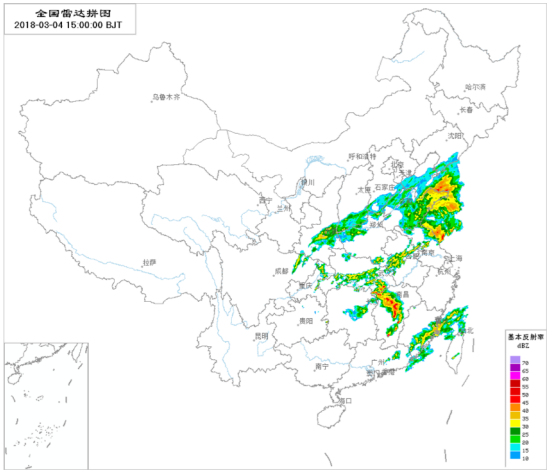
- ConvNet predicts the trajectories of falling blocks
- Uses the Unreal game engine hooked up to Torch.

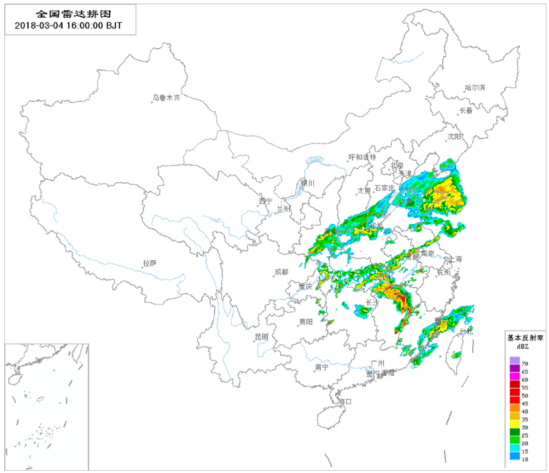


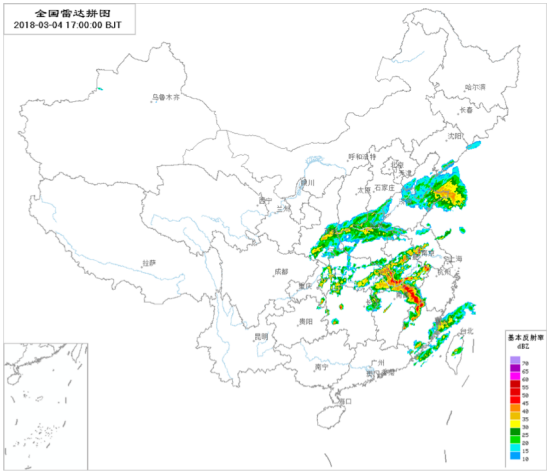
Video Prediction: Weather Nowcasting

March 4th, Jiangxi Province, China

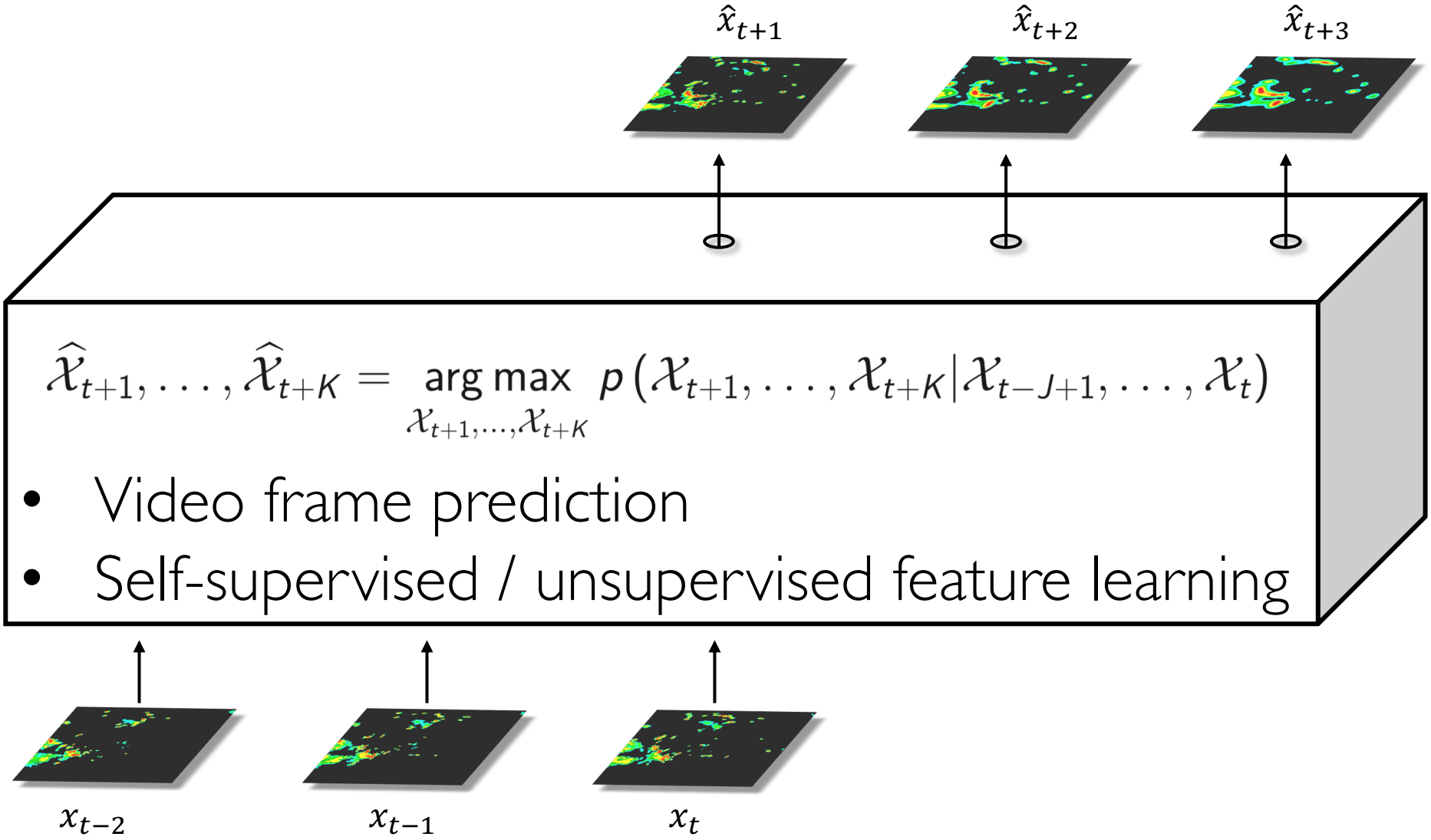




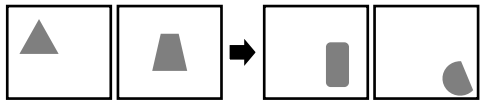
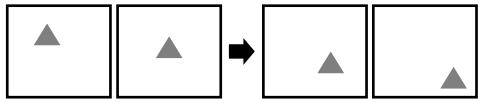




Predictive Learning



Technical Challenges

	Optical Flow	CNN	RNN	LSTM
Spatial deformation 	✗	✓	✗	✗
Short Term movement 	✗	✗	✓	✓
Long Term movement 	✗	✗	✗	✗

How?

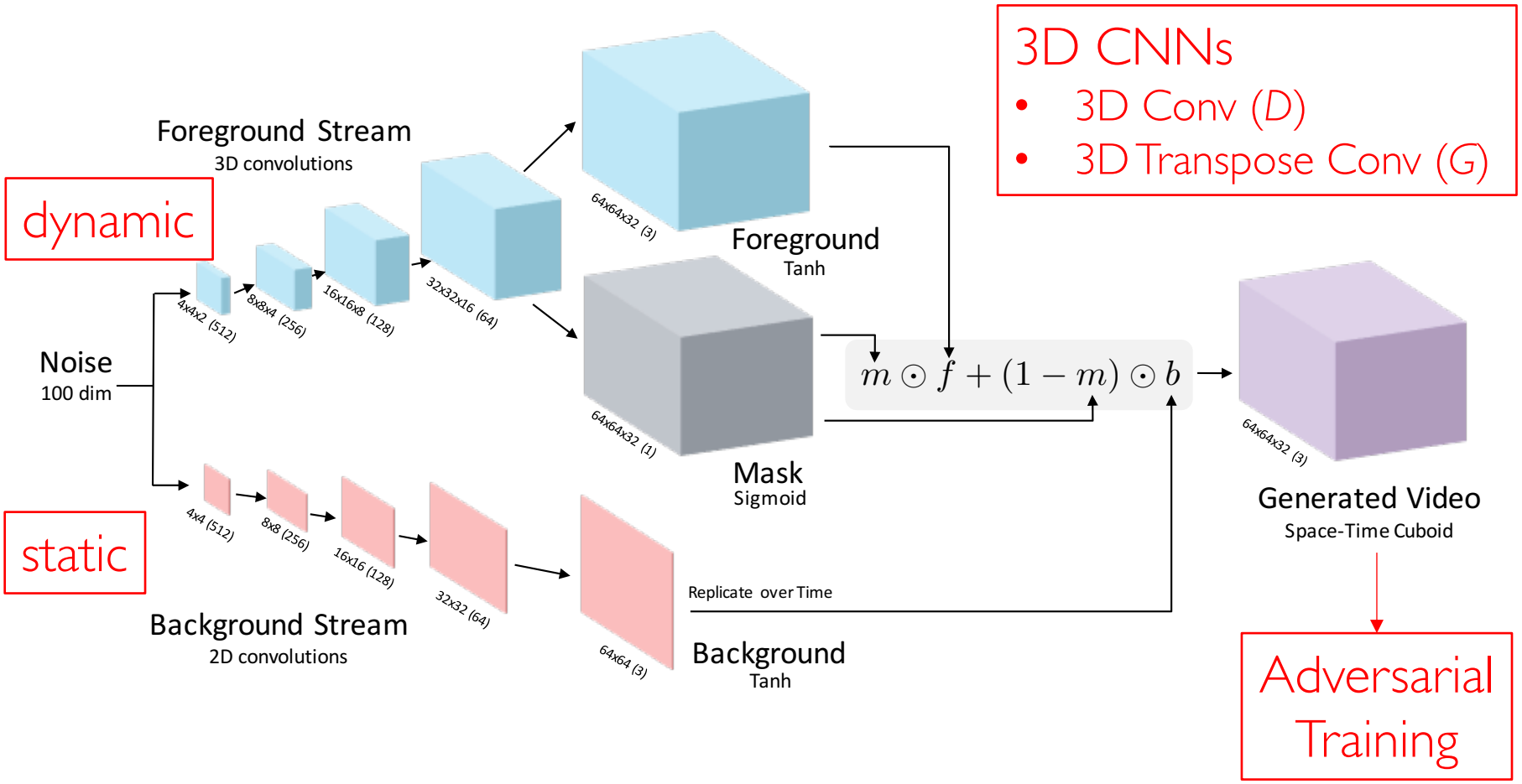
CNN-based methods

- 3D CNNs [Vondrick 2016]

RNN-based methods

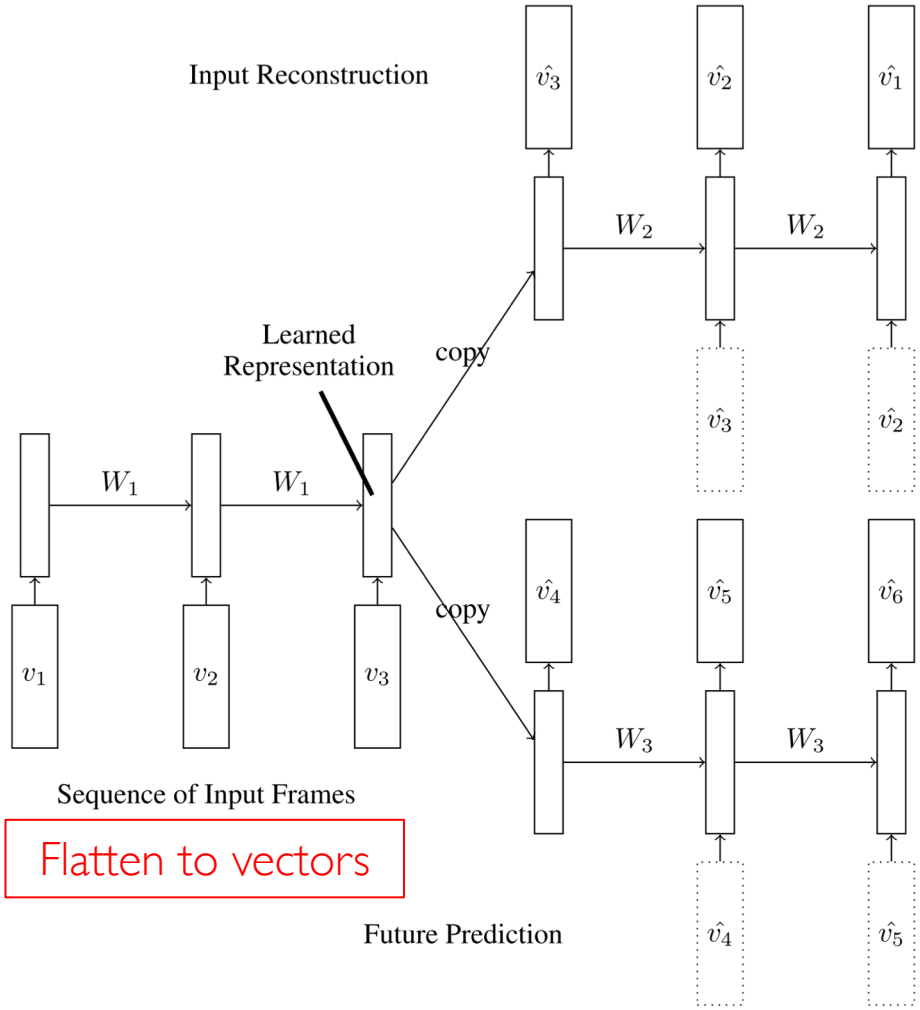
- ConvLSTM network [Shi 2015]
- Video Pixel Networks [Kalchbrenner 2017]
- PredRNN / PredRNN++ [Wang 2017, Wang 2018]

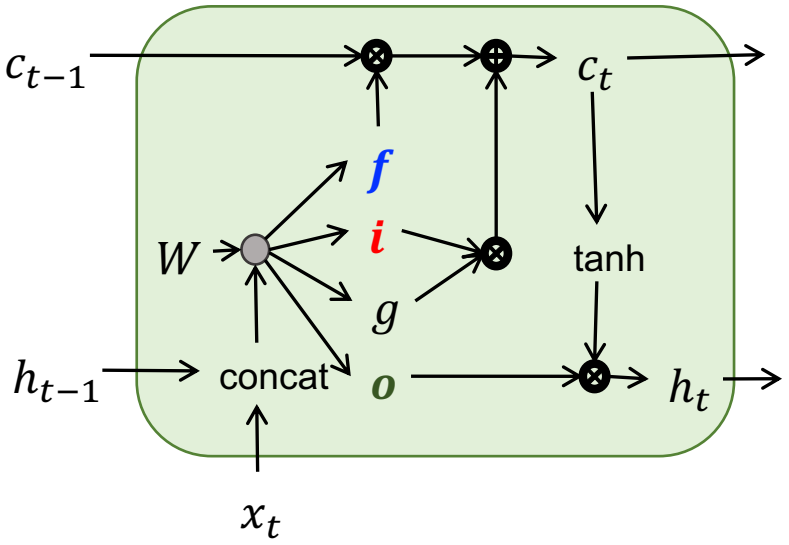
VideoGAN



Generating Videos with Scene Dynamics. Vondrick et al. NIPS 2016.

Warm-Up: Seq-to-Seq LSTMs

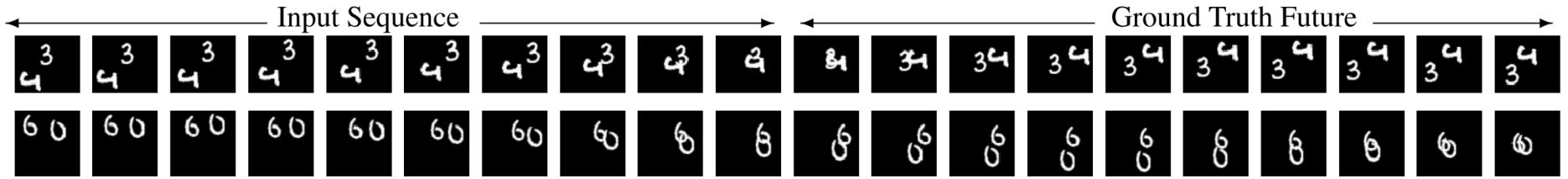


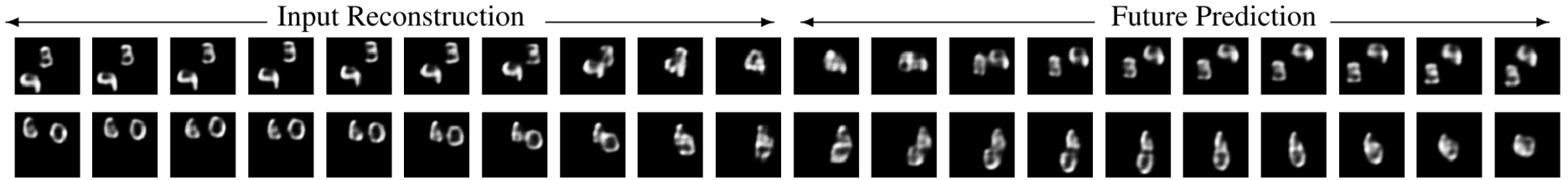


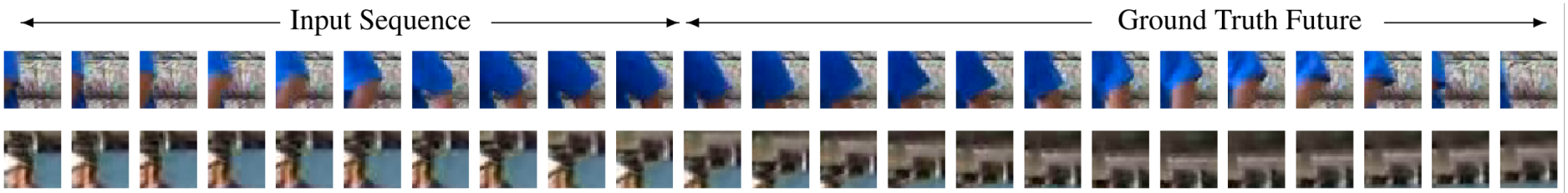
- Inputs, outputs and states at all timestamps are 1D vectors
- Dimensions of a hidden state can be swapped without affecting the overall structure

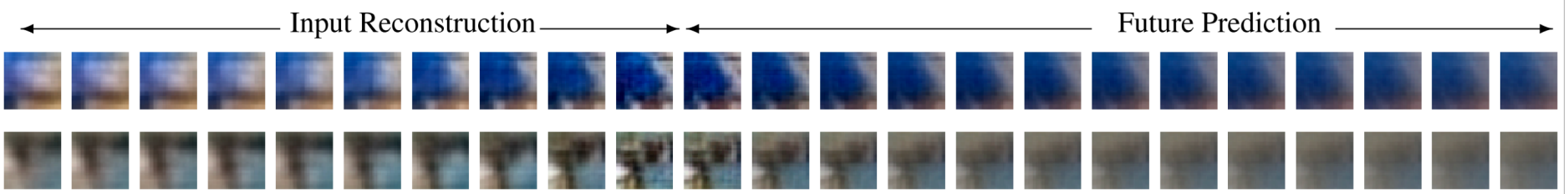
[Srivastava et al. Unsupervised Learning of Video Representations using LSTMs. ICML 2015]

Warm-Up: Seq-to-Seq LSTMs



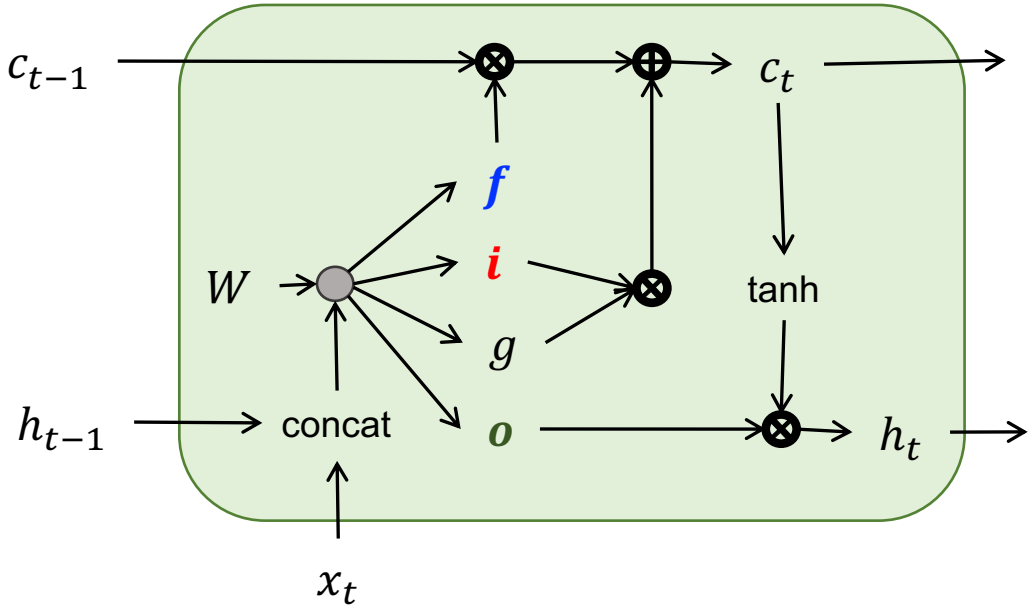






Two Layer Composite Model with 2048 LSTM units

Long Short-Term Memory (LSTM)



LSTM

Input, forget, output gates:

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + b_i)$$
$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + b_f)$$
$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + b_o)$$

Cell state:

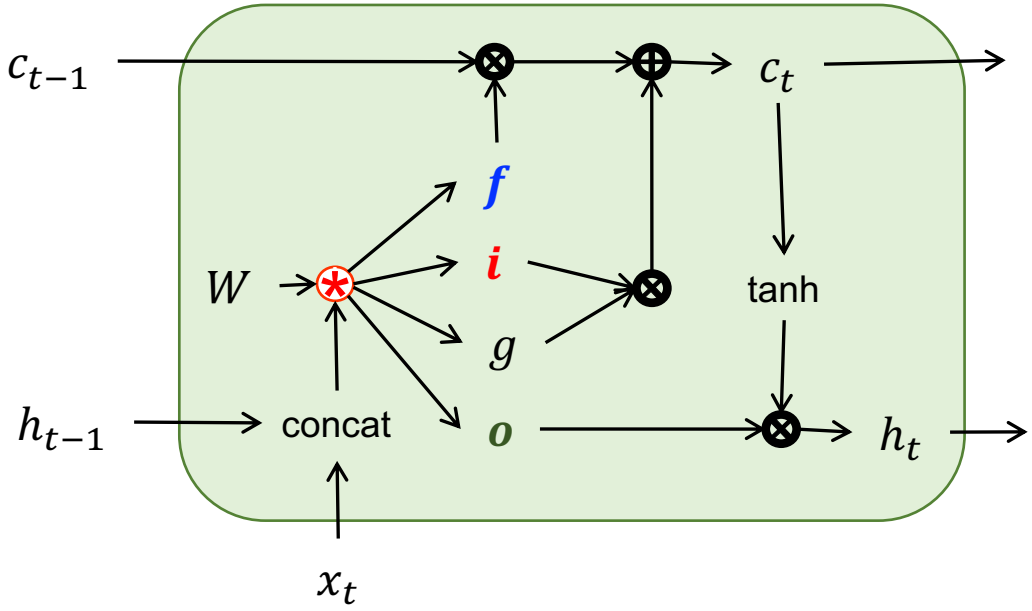
$$g_t = \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c)$$
$$c_t = f_t \odot c_{t-1} + i_t \odot g_t$$

1. All variables are 1D vectors
2. Dimensions are Permutable (Spatial Information Loss)

Hidden layer:

$$h_t = o_t \odot \tanh(c_t)$$

Convolutional LSTM (ConvLSTM)



ConvLSTM

Input, forget, output gates:

$$i_t = \sigma(W_{xi} * X_t + W_{hi} * H_{t-1} + b_i)$$
$$f_t = \sigma(W_{xf} * X_t + W_{hf} * H_{t-1} + b_f)$$
$$o_t = \sigma(W_{xo} * X_t + W_{ho} * H_{t-1} + b_o)$$

Cell state:

$$g_t = \tanh(W_{xc} * X_t + W_{hc} * H_{t-1} + b_c)$$
$$C_t = f_t \odot C_{t-1} + i_t \odot g_t$$

Hidden layer:

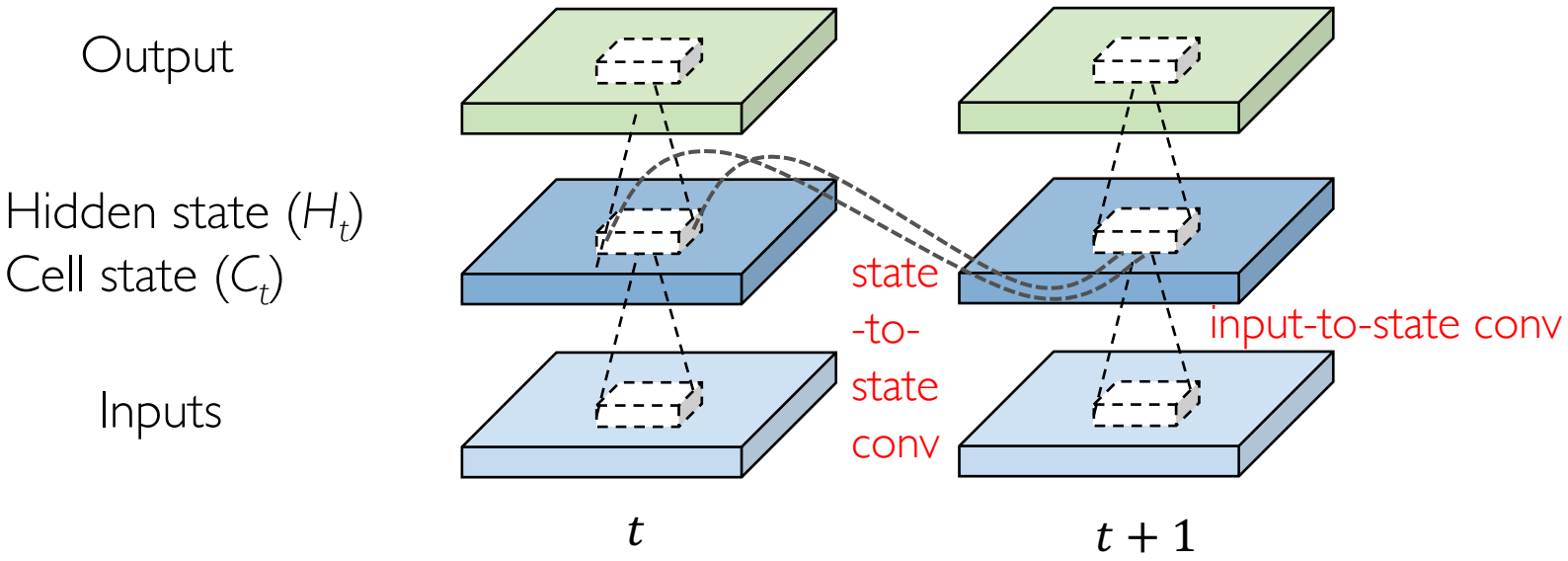
$$H_t = o_t \odot \tanh(C_t)$$

1. All variables are 3D tensors
2. Spatial Dimensions are Not Permutable (Spatial Preservation)

[Convolutional LSTM Network. Shi et al. NIPS 2015.]

ConvLSTM

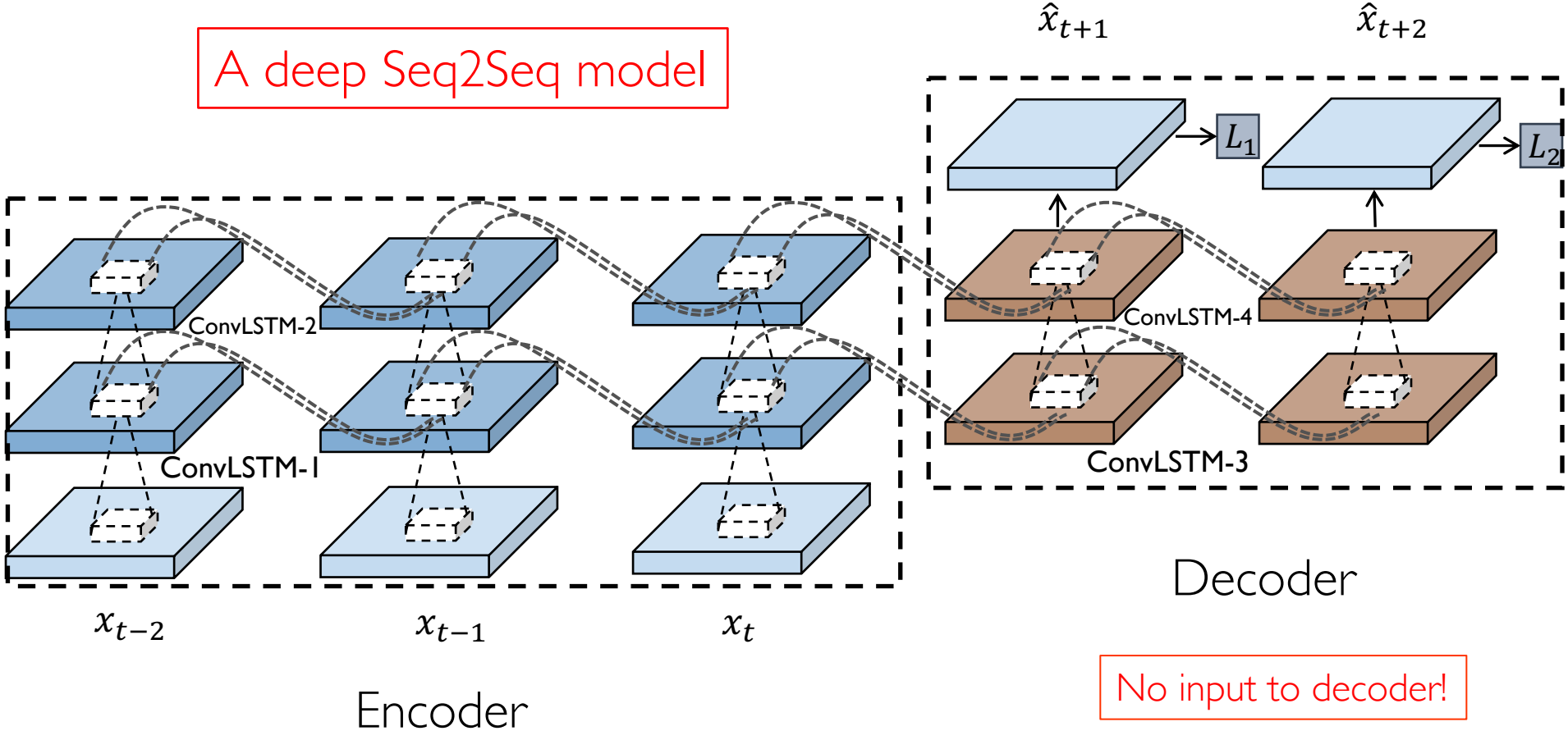
MatMul → Convolution



State-to-state Convolution: future states → larger receptive field

[Convolutional LSTM Network. Shi et al. NIPS 2015.]

ConvLSTM Network

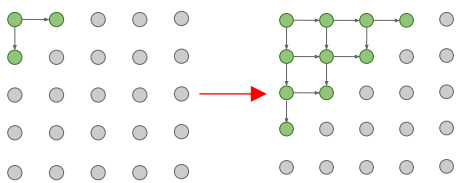


[Convolutional LSTM Network. Shi et al. NIPS 2015.]

PixelRNN

- Explicit density model
  - Use chain rule to decompose likelihood of an image $x = (x_1, \ldots, x_T)$ into product of 1-dim distributions:

Pixel Ordering



$$P_\theta(x) = \prod_{t=1}^{T} P_\theta(x_t \mid x_1, \ldots, x_{t-2}, x_{t-1})$$

Likelihood of image $x = (x_1, \ldots, x_T)$

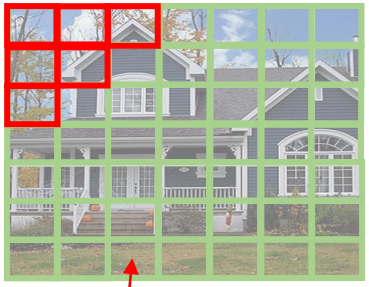
Probability of t-th pixel value given all previous pixels

  - Then maximize likelihood of training data $\{x\}$

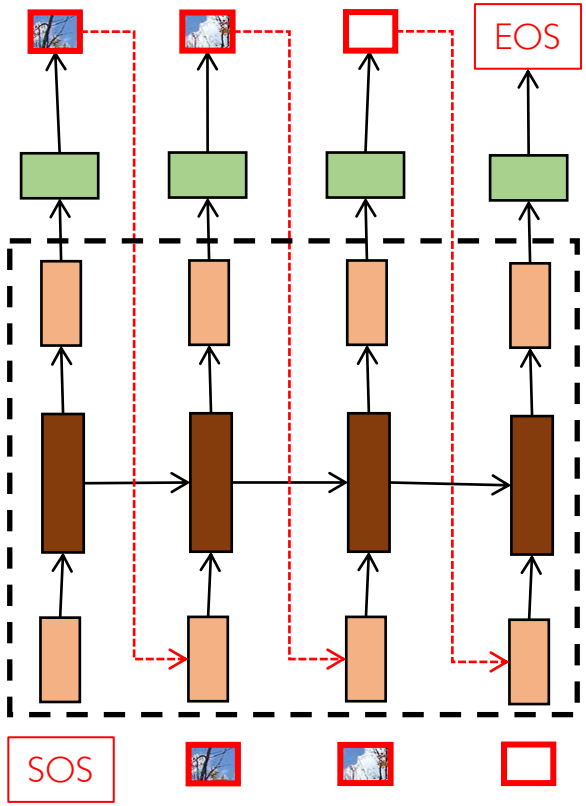
A Oord et al. Pixel Recurrent Neural Networks. ICML 2016.

PixelRNN

Sequential Ordering



Generate an image as pixel sequence → RNN



Markov process

$$P(x_1,\ldots,x_T)$$

$$=\prod_{t=1}^{T} P(x_t \mid x_1,\ldots,x_{t-2},x_{t-1})$$

$$=\prod_{t=1}^{T} g(s_{t-2},x_{t-1})$$

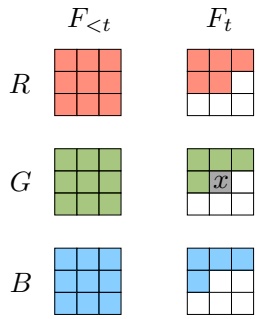
A pixel

(note: the box is used to indicate a pixel)

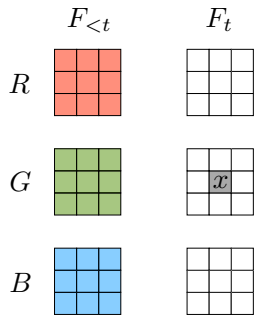
At test time, predict pixels recursively!

A Oord et al. Pixel Recurrent Neural Networks. ICML 2016.

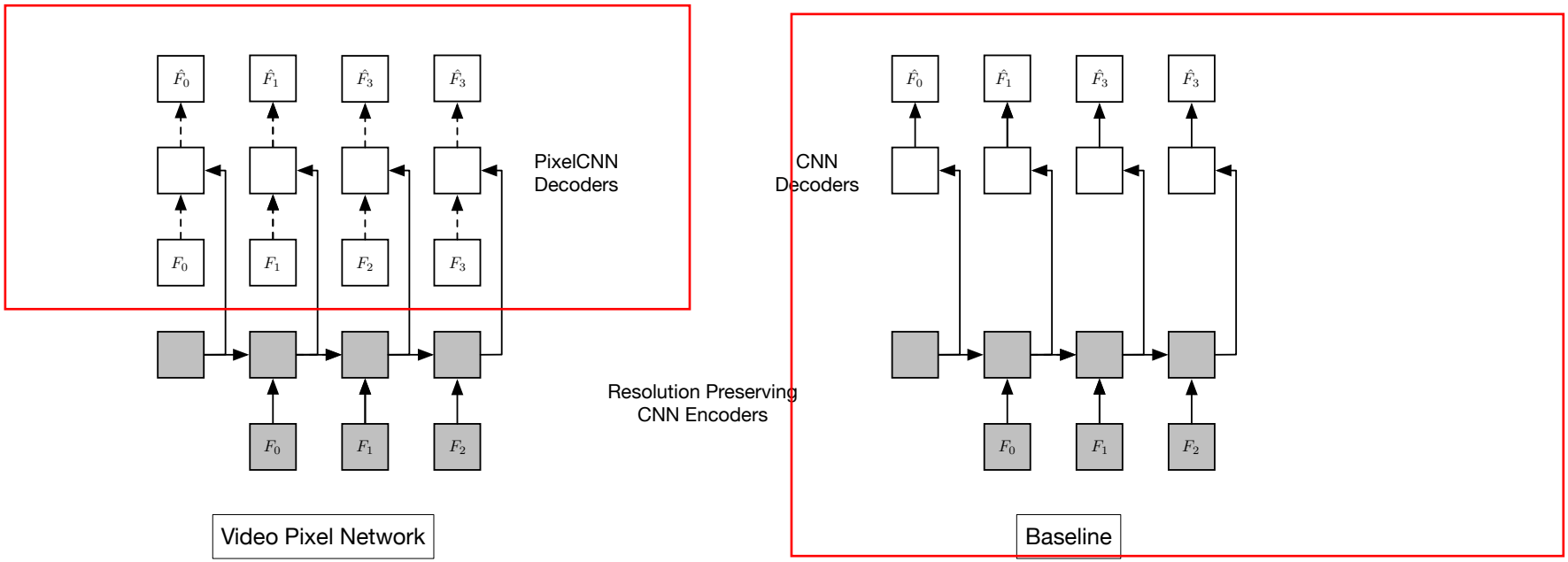
Video Pixel Networks (VPN)



PixelRNN Decoders



ConvLSTM

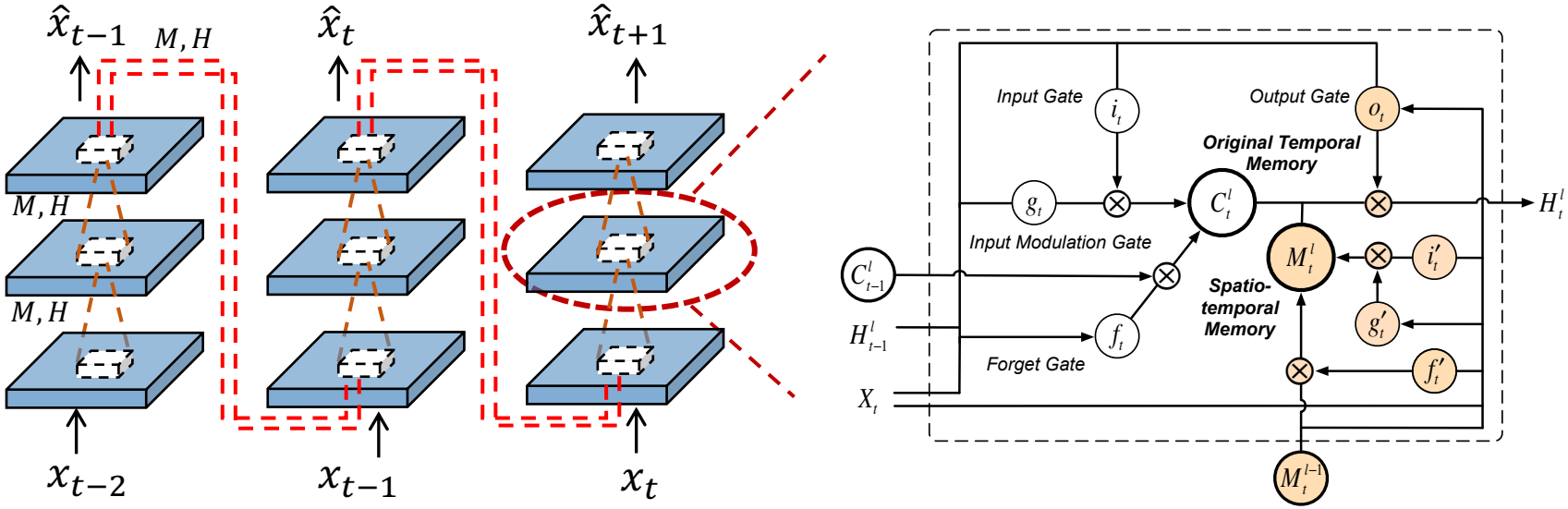


Video Pixel Networks. Kalchbrenner et al. ICML 2017.

PredRNN

Zigzag Info Flow

- Deeper transitions for short-term (fast flow) dependencies
- Larger receptive field across adjacent states for sudden change

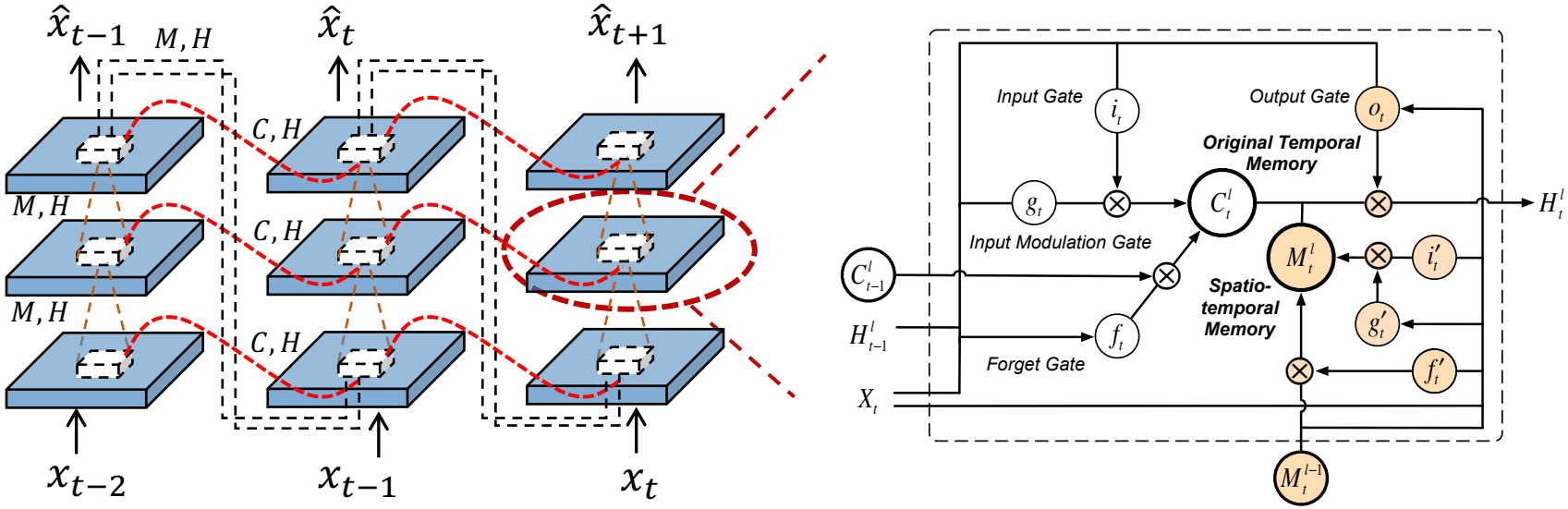


PredRNN: Recurrent Neural Networks for Predictive Learning using Spatiotemporal LSTMs. Yunbo Wang et al. NIPS 2017.

PredRNN

Zigzag Info Flow

- Zigzag flow makes network deeper in time → gradient vanishing
- Maintain temporal memory (C_t) → Preserve long-term gradients

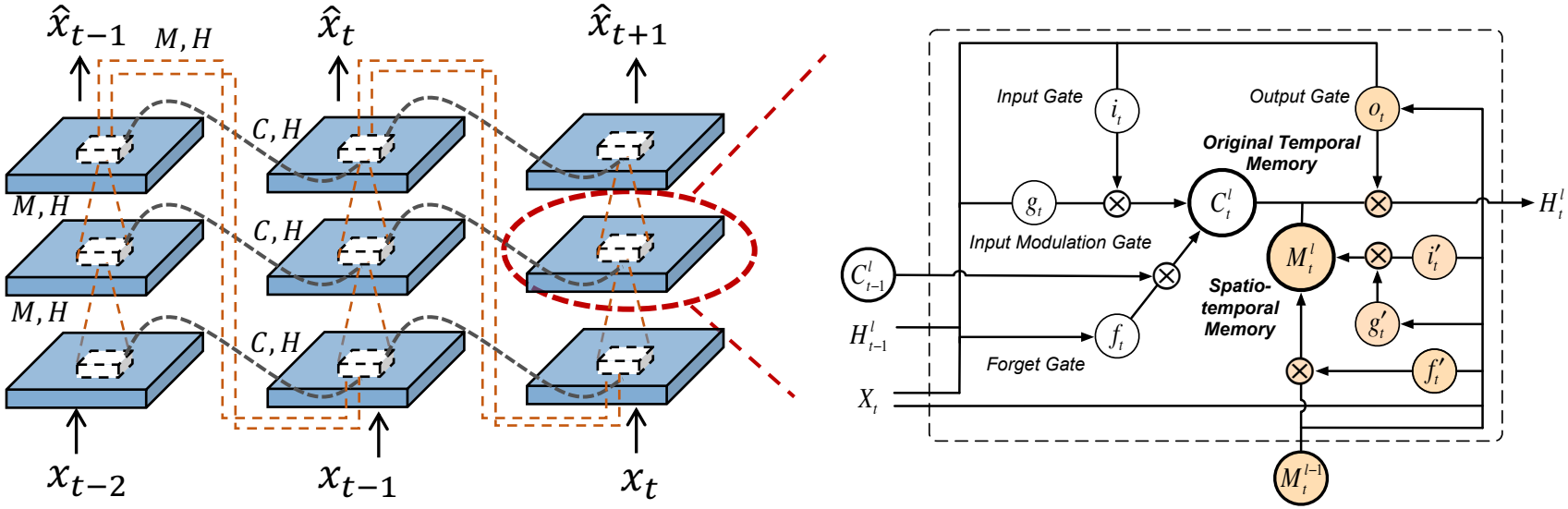


PredRNN: Recurrent Neural Networks for Predictive Learning using Spatiotemporal LSTMs. Yunbo Wang et al. NIPS 2017.

PredRNN

Dual memory

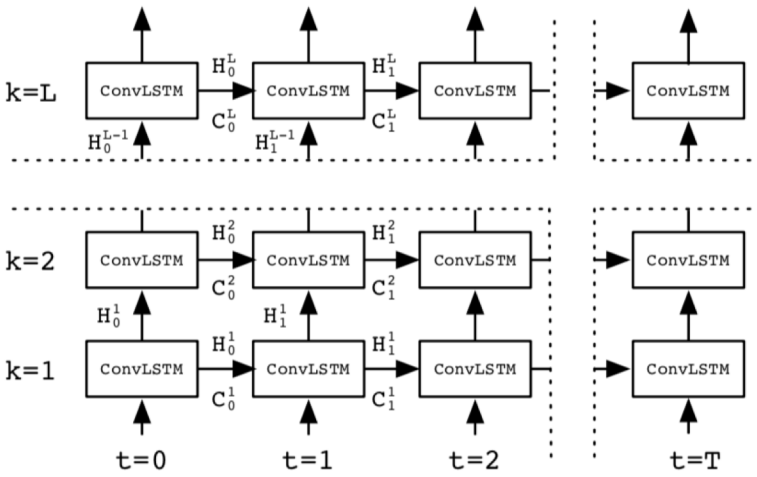
- Spatiotemporal Memory (in yellow): an external memory (M_t^l)
- Temporal Memory (in black): the conventional memory cell (C_t^l)



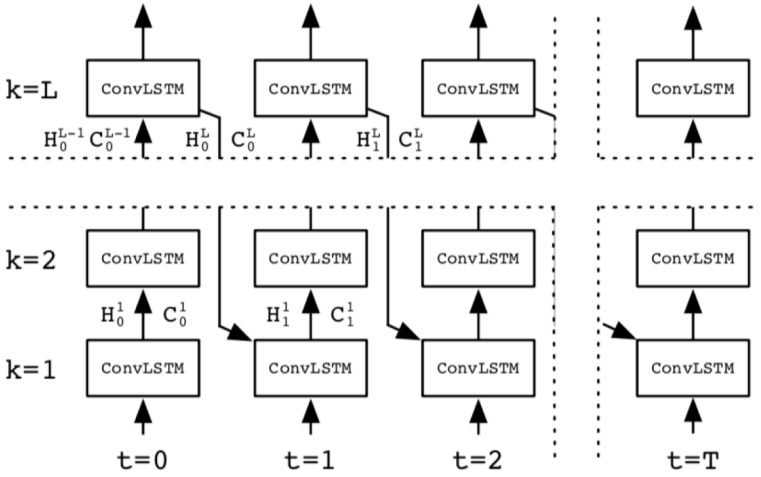
PredRNN: Recurrent Neural Networks for Predictive Learning using Spatiotemporal LSTMs. Yunbo Wang et al. NIPS 2017.

PredRNN++

Zigzag flow: more straightforward supervisions to the memory



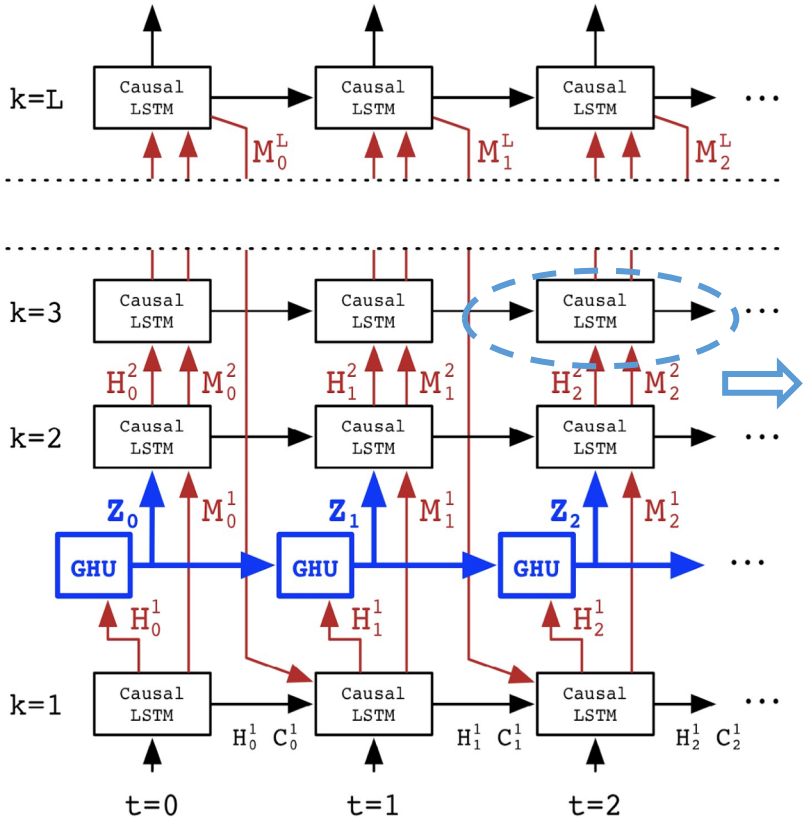
(a) Stacked LSTMs



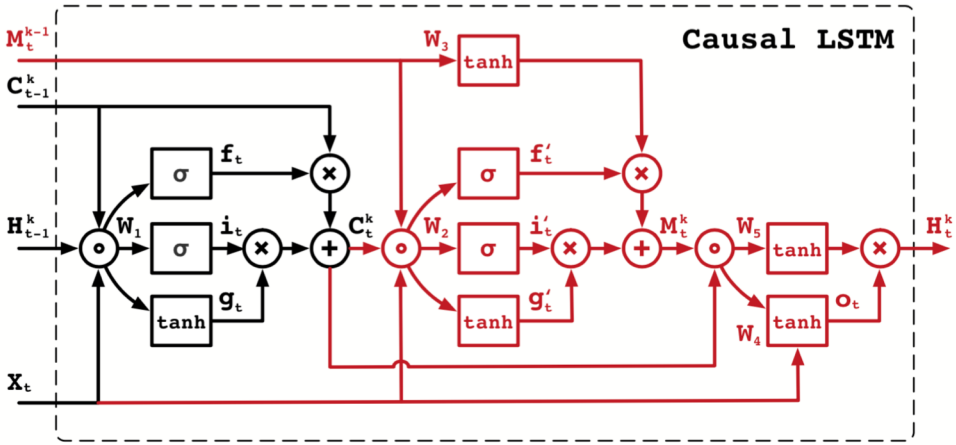
(b) Deep Transition LSTMs

But, for the short term -> deeper-in-time networks -> gradient vanishing -> bad long-term modeling capability

PredRNN++: Longer path for short-term dynamics

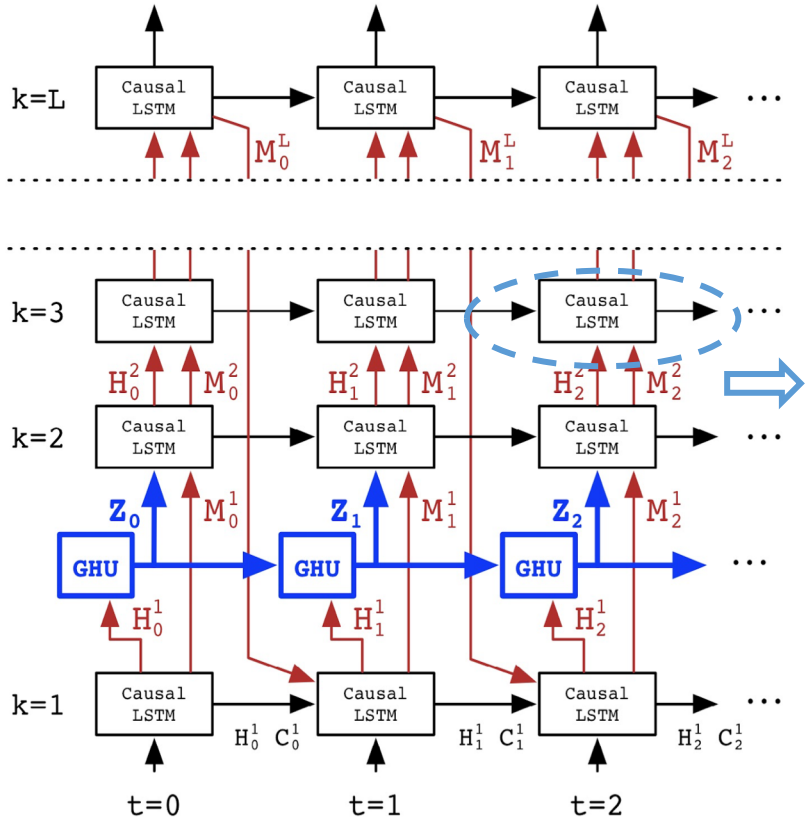


- A dual memory structure, C_t^k: the conventional LSTM memory
- M_t^k: a zigzag delivered memory (vertically and horizontally)
- $C_t^k \rightarrow M_t^k$: add more non-linear layers to recurrent transitions from one time step to the next.



[PredRNN++. Yunbo Wang et al. ICML 2018.]

PredRNN++: Longer path for short-term dynamics



- A dual memory structure, C_t^k: the conventional LSTM memory
- M_t^k: a zigzag delivered memory (vertically and horizontally)
- $C_t^k \rightarrow M_t^k$: add more non-linear layers to recurrent transitions from one time step to the next.

$$\begin{pmatrix} g_t \\ i_t \\ f_t \end{pmatrix} = \begin{pmatrix} \tanh \\ \sigma \\ \sigma \end{pmatrix} W_1 * \left[\mathcal{X}_t, \mathcal{H}_{t-1}^k, \mathcal{C}_{t-1}^k\right]$$

$$\mathcal{C}_t^k = f_t \odot \mathcal{C}_{t-1}^k + i_t \odot g_t$$

$$\begin{pmatrix} g_t' \\ i_t' \\ f_t' \end{pmatrix} = \begin{pmatrix} \tanh \\ \sigma \\ \sigma \end{pmatrix} W_2 * \left[\mathcal{X}_t, \mathcal{C}_t^k, \mathcal{M}_t^{k-1}\right]$$

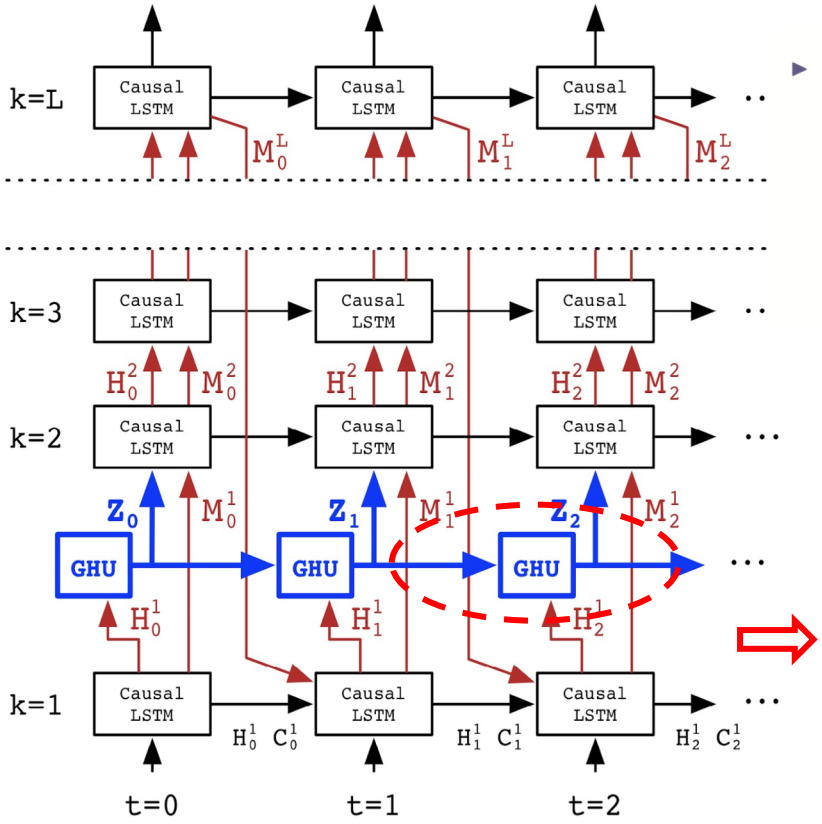
$$\mathcal{M}_t^k = f_t' \odot \tanh\left(W_3 * \mathcal{M}_t^{k-1}\right) + i_t' \odot g_t'$$

$$o_t = \tanh\left(W_4 * \left[\mathcal{X}_t, \mathcal{C}_t^k, \mathcal{M}_t^k\right]\right)$$

$$\mathcal{H}_t^k = o_t \odot \tanh\left(W_5 * \left[\mathcal{C}_t^k, \mathcal{M}_t^k\right]\right)$$

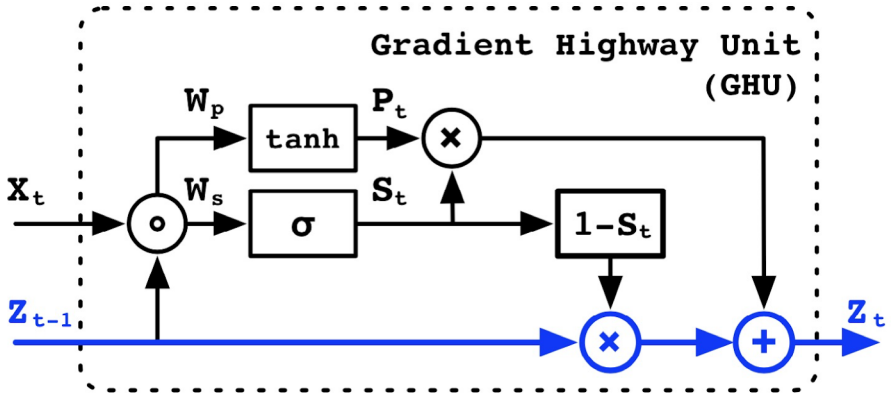
[PredRNN++. Yunbo Wang et al. ICML 2018.]

PredRNN++: Shorter path for long-term relations



- Enables an adaptive learning between the quickly updated $\mathcal{H}_t(\mathcal{X}_t)$ and the lazily updated $\mathcal{Z}_{t-1}$. $\mathcal{Z}_t = \mathrm{GHU}(\mathcal{X}_t, \mathcal{Z}_{t-1})$:

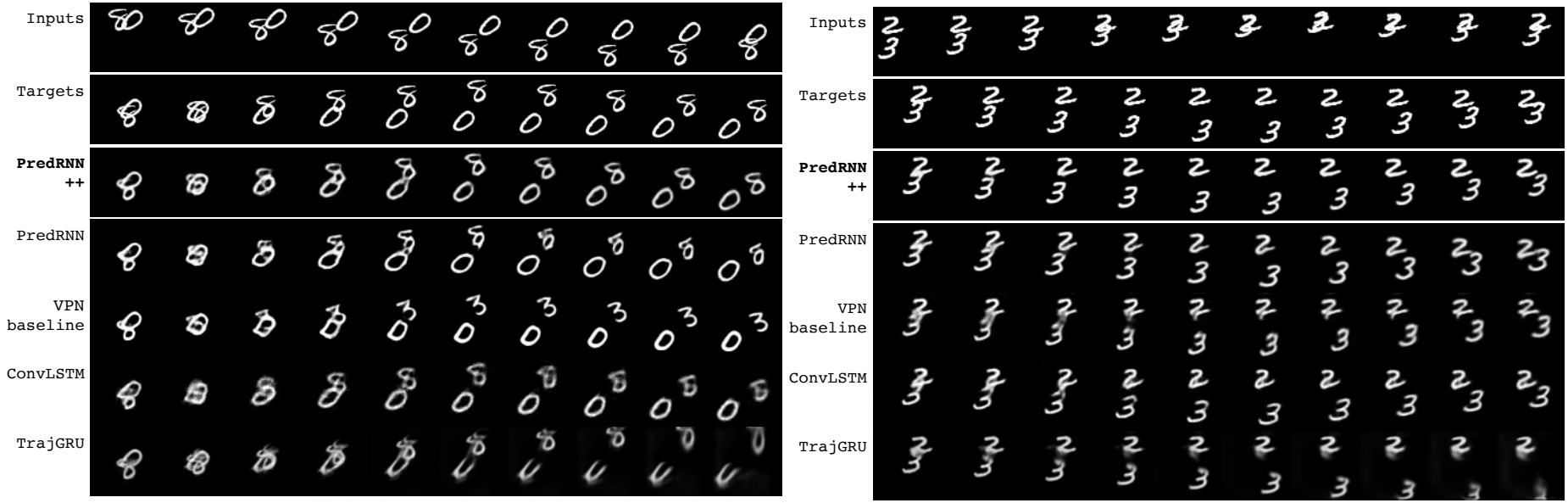
$$\mathcal{P}_t = \tanh\left(W_{px} * \mathcal{X}_t + W_{pz} * \mathcal{Z}_{t-1}\right)$$
$$\mathcal{S}_t = \sigma\left(W_{sx} * \mathcal{X}_t + W_{sz} * \mathcal{Z}_{t-1}\right)$$
$$\mathcal{Z}_t = \mathcal{S}_t \odot \mathcal{P}_t + (1 - \mathcal{S}_t) \odot \mathcal{Z}_{t-1}$$



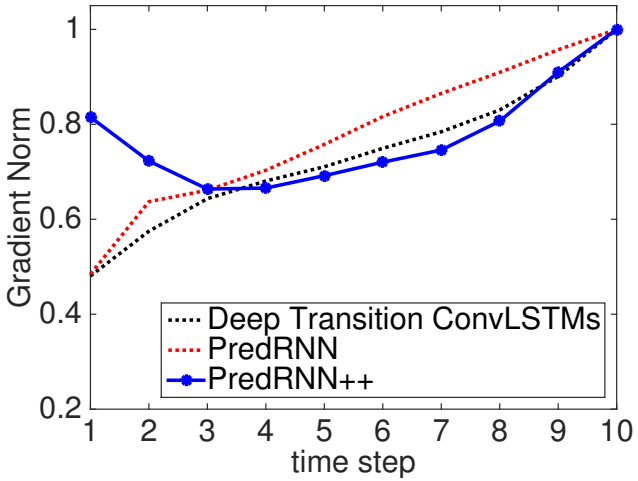
[PredRNN++. Yunbo Wang et al. ICML 2018.]

Moving MNIST: Frequent Occlusions

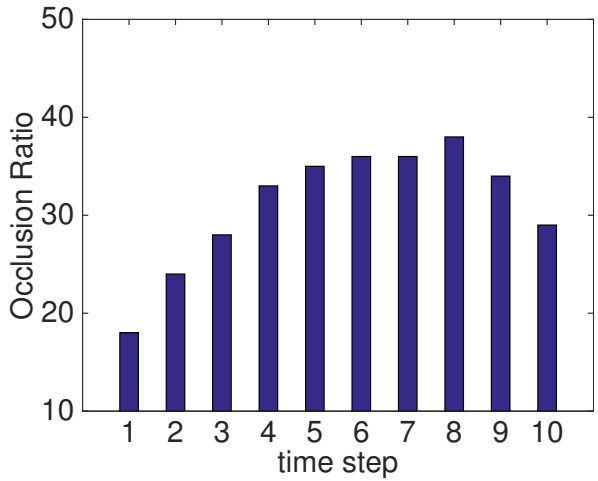
Model	MNIST-2				MNIST-3	
	10 time steps		30 time steps		10 time steps	
	SSIM	MSE	SSIM	MSE	SSIM	MSE
FC-LSTM (Srivastava et al., 2015a)	0.690	118.3	0.583	180.1	0.651	162.4
ConvLSTM (Shi et al., 2015)	0.707	103.3	0.597	156.2	0.673	142.1
TrajGRU (Shi et al., 2017)	0.713	106.9	0.588	163.0	0.682	134.0
CDNA (Finn et al., 2016)	0.721	97.4	0.609	142.3	0.669	138.2
DFN (De Brabandere et al., 2016)	0.726	89.0	0.601	149.5	0.679	140.5
VPN* (Kalchbrenner et al., 2017)	0.870	64.1	0.620	129.6	0.734	112.3
PredRNN (Wang et al., 2017)	0.867	56.8	0.645	112.2	0.782	93.4
Causal LSTM	0.882	52.5	0.685	100.7	0.795	89.2
Causal LSTM (variant: spatial-to-temporal)	0.875	54.0	0.672	103.6	0.784	91.8
PredRNN + GHU	0.886	50.7	0.713	98.4	0.790	88.9
Causal LSTM + GHU (Final)	**0.898**	**46.5**	**0.733**	**91.1**	**0.814**	**81.7**



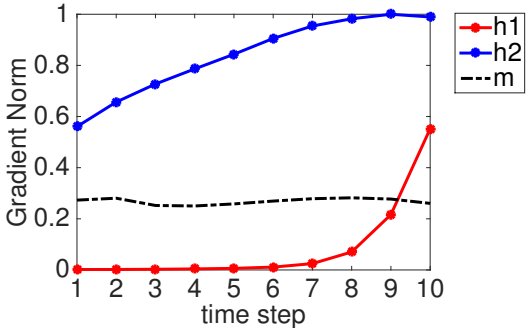
Moving MNIST: Frequent Occlusions



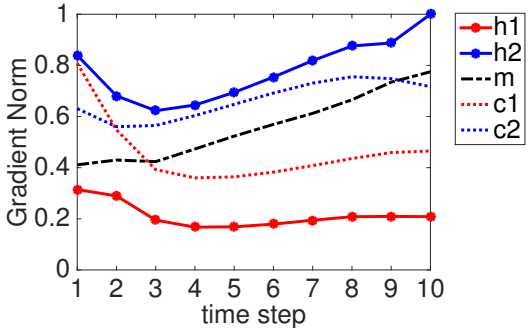
(a) $\|\nabla_{\mathcal{X}_t}\mathcal{L}_{20}\|$



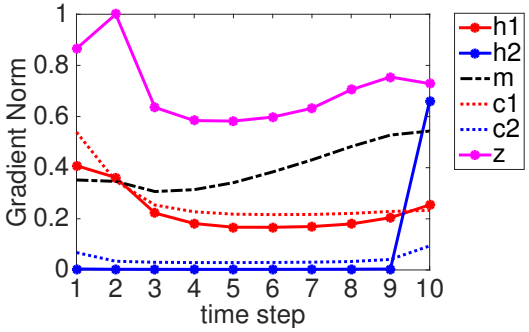
(b) Occlusion Frequency



(a) Deep Transition ConvLSTMs



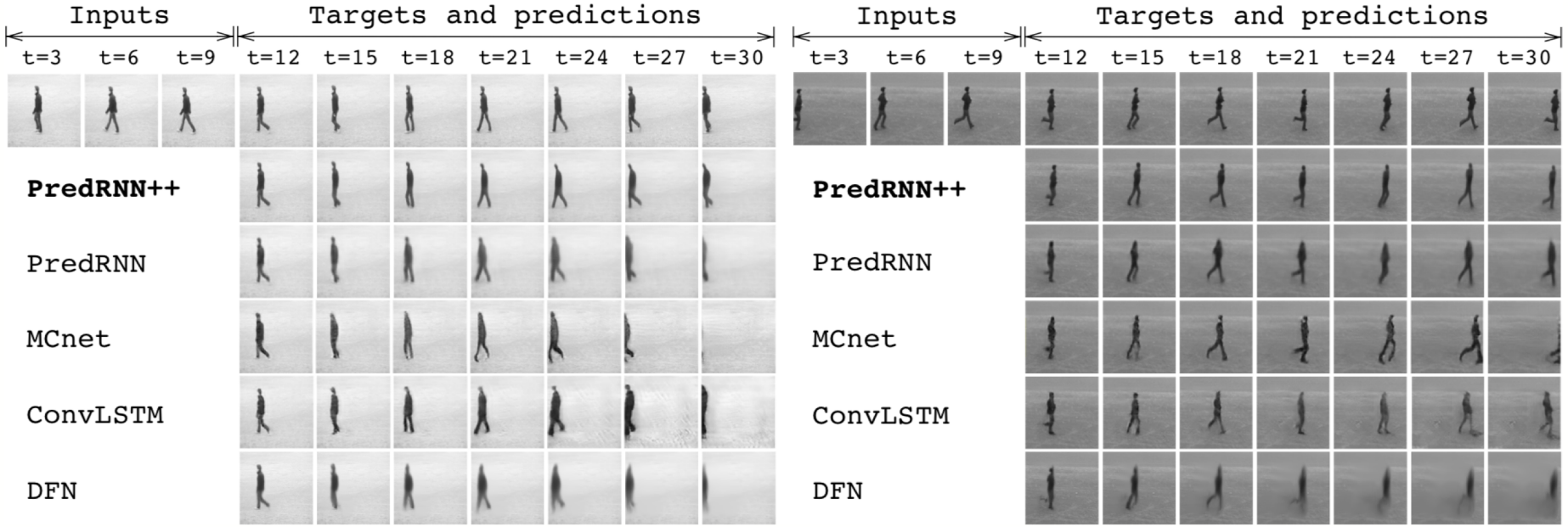
(b) PredRNN



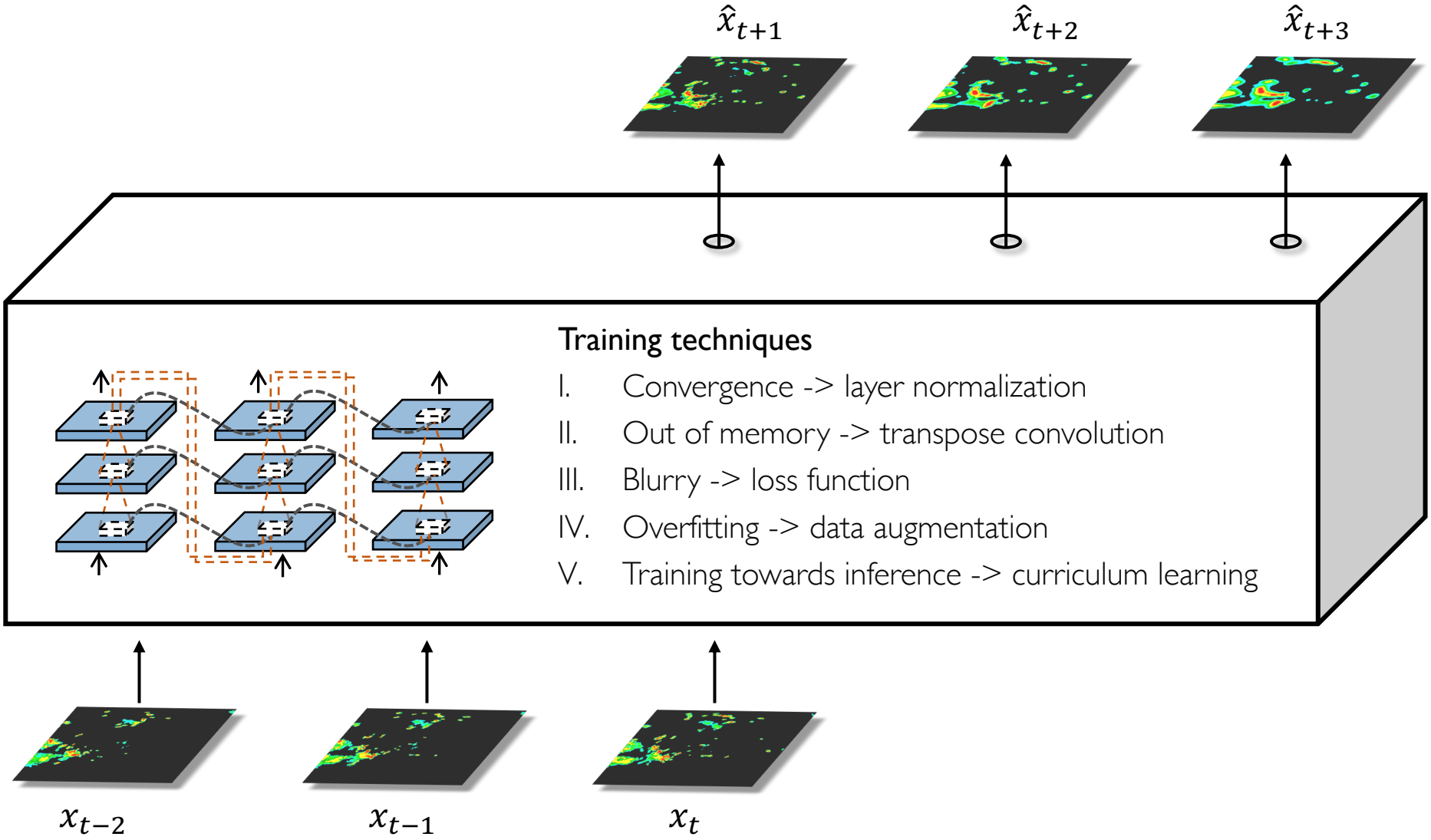
(c) PredRNN++

Real Videos: Predictable Movement

Model	PSNR	SSIM
Ours	**28.47**	**0.865**
PredRNN [Wang et al., 2017]	27.55	0.839
MCnet [Villegas et al., 2017]	25.95	0.804
ConvLSTM [Shi et al., 2015]	23.58	0.712
DFN [De Brabandere et al., 2016]	27.26	0.794



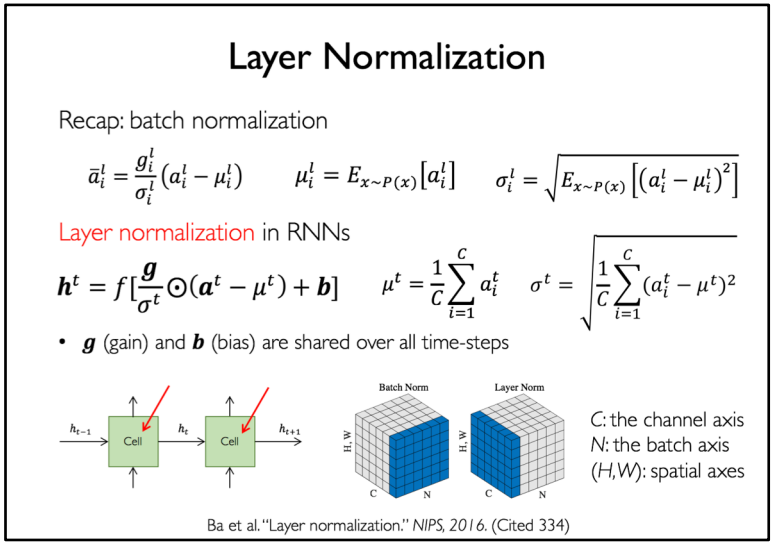
Weather Forecasting: A Case Study

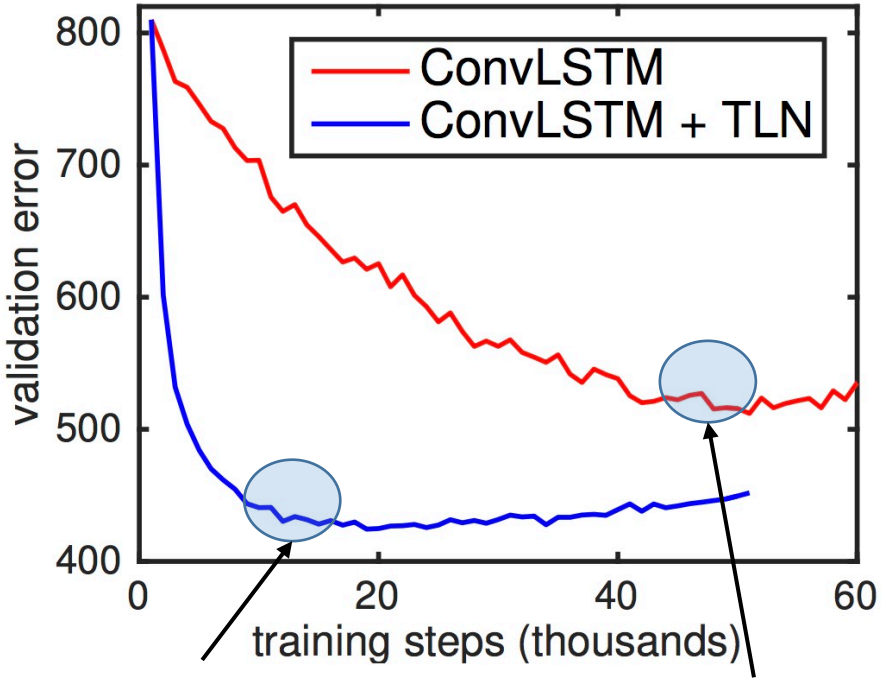


I. Layer Normalization -> Convergence

After convolutions but right before nonlinearity

- Cutting memory usage 87.5% by saving mini-batch
- Cutting training time 75%
- Increase accuracy 20%



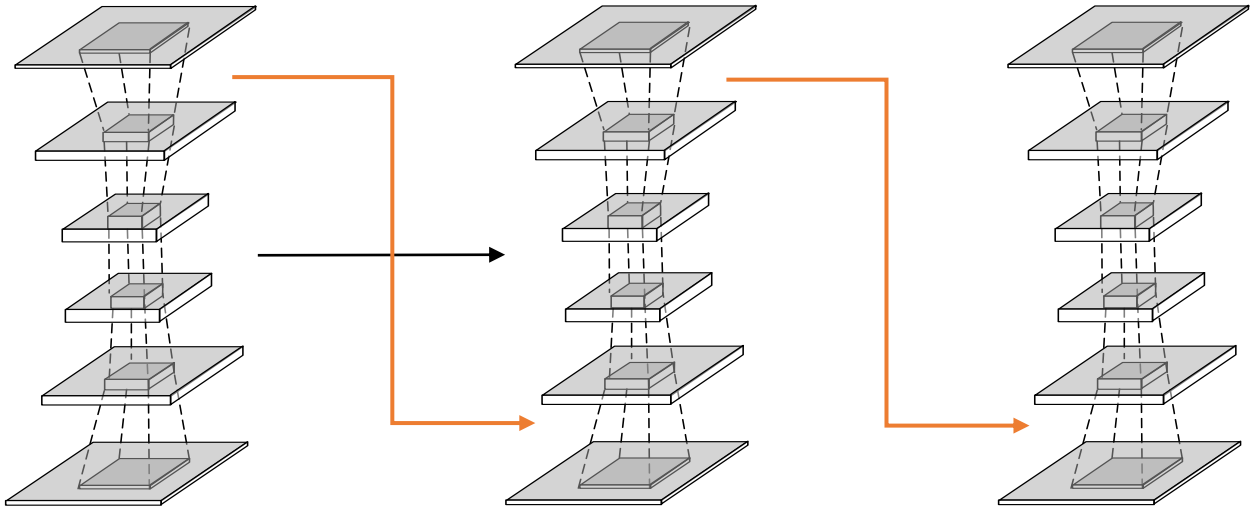


II. Transpose Convolution -> OOM

Many intermediate activations -> memory explosion

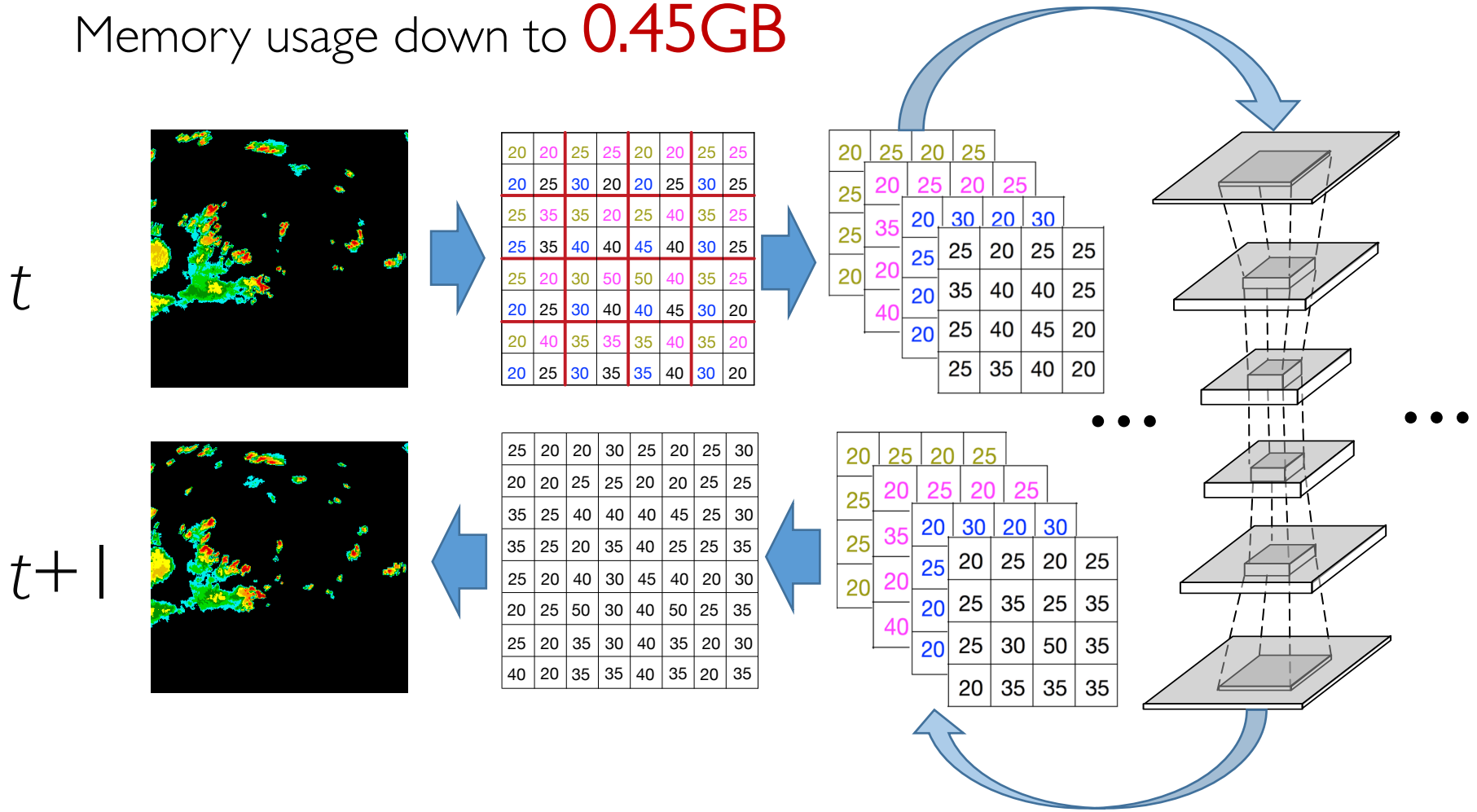
Resolution 400*400 -> Memory usage 18.31GB

Transpose convolution -> 7.15GB



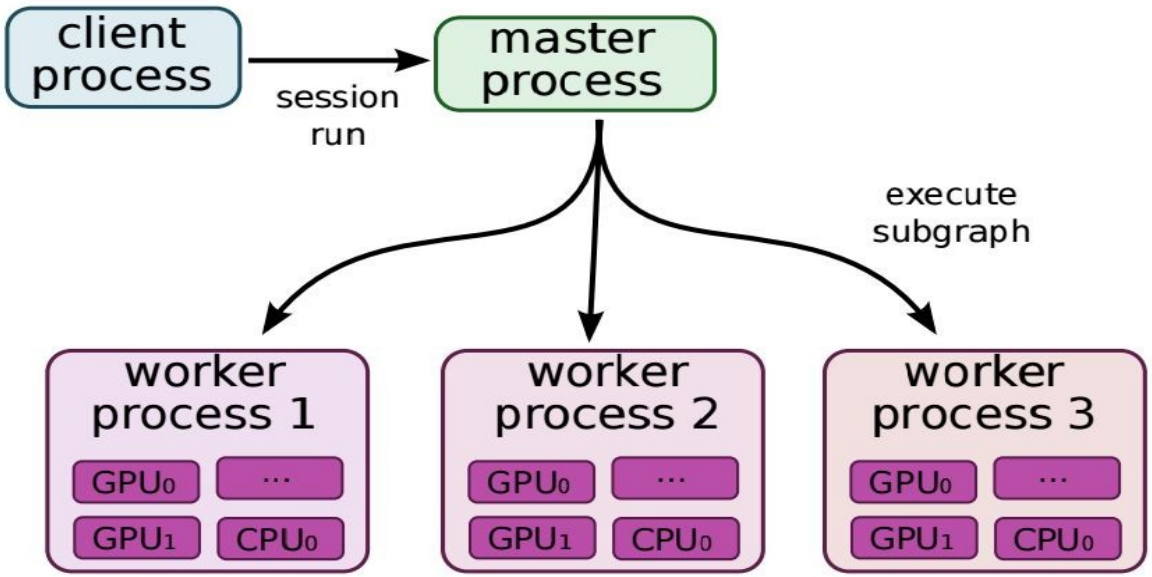
II. Sampling -> OOM

Memory usage down to 0.45GB

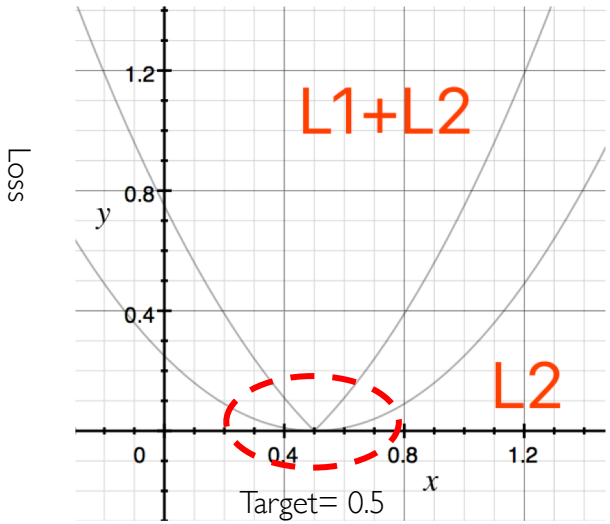


II. Multiple GPUs -> OOM

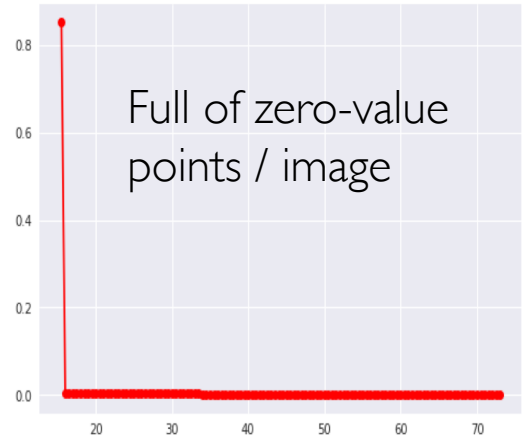
Cutting training time 60%

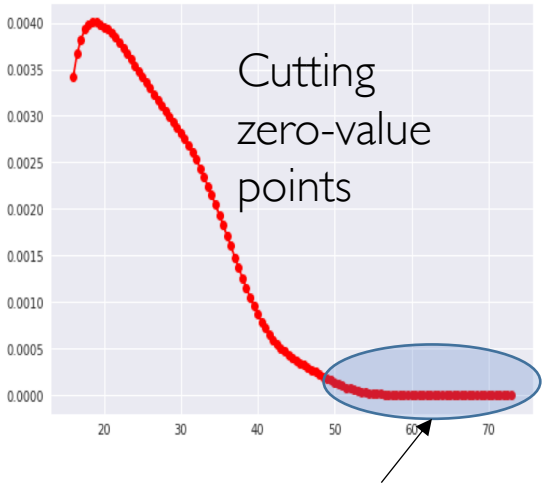


III. Loss Function -> Blurry



Proportion

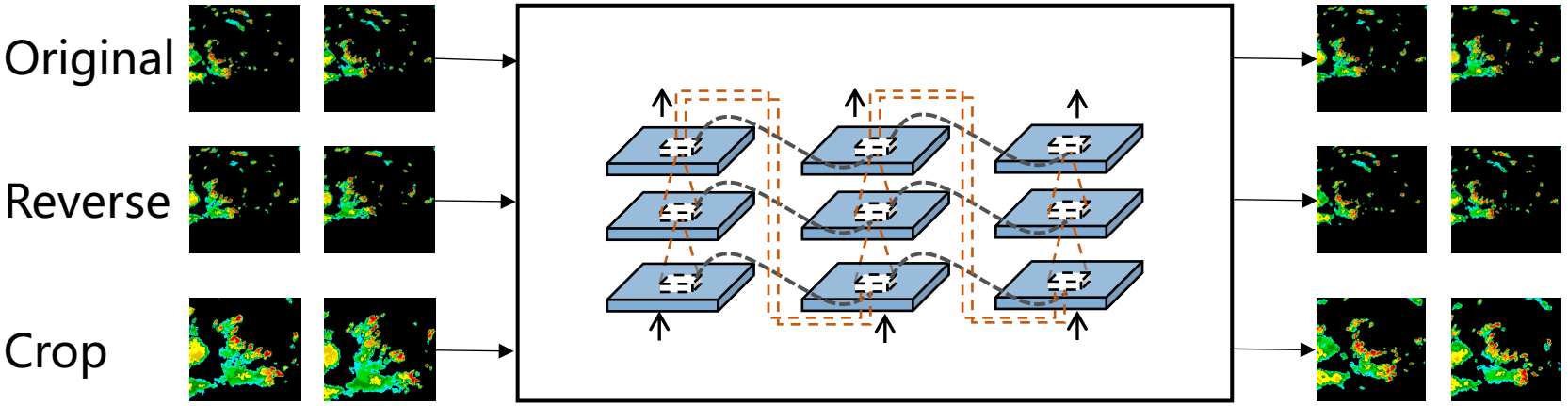




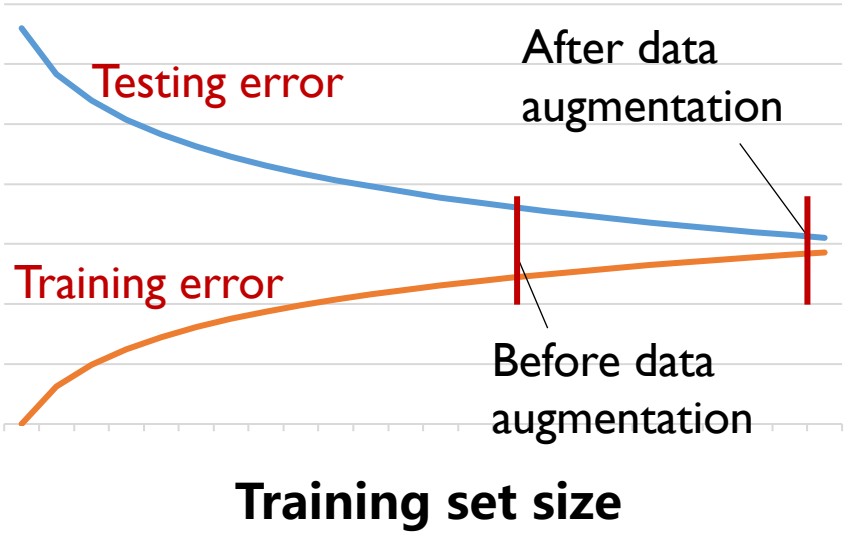
$$L = L1 + L2 = \sum_{i=0}^{m*n}(|x_i - \hat{x_i}|_1 + |x_i - \hat{x_i}|_2^2)$$

$$cost = \sum \begin{pmatrix} W_{20dBZ} * L + \\ W_{30dBZ} * L + \\ W_{40dBZ} * L + \\ W_{50dBZ} * L \end{pmatrix}$$

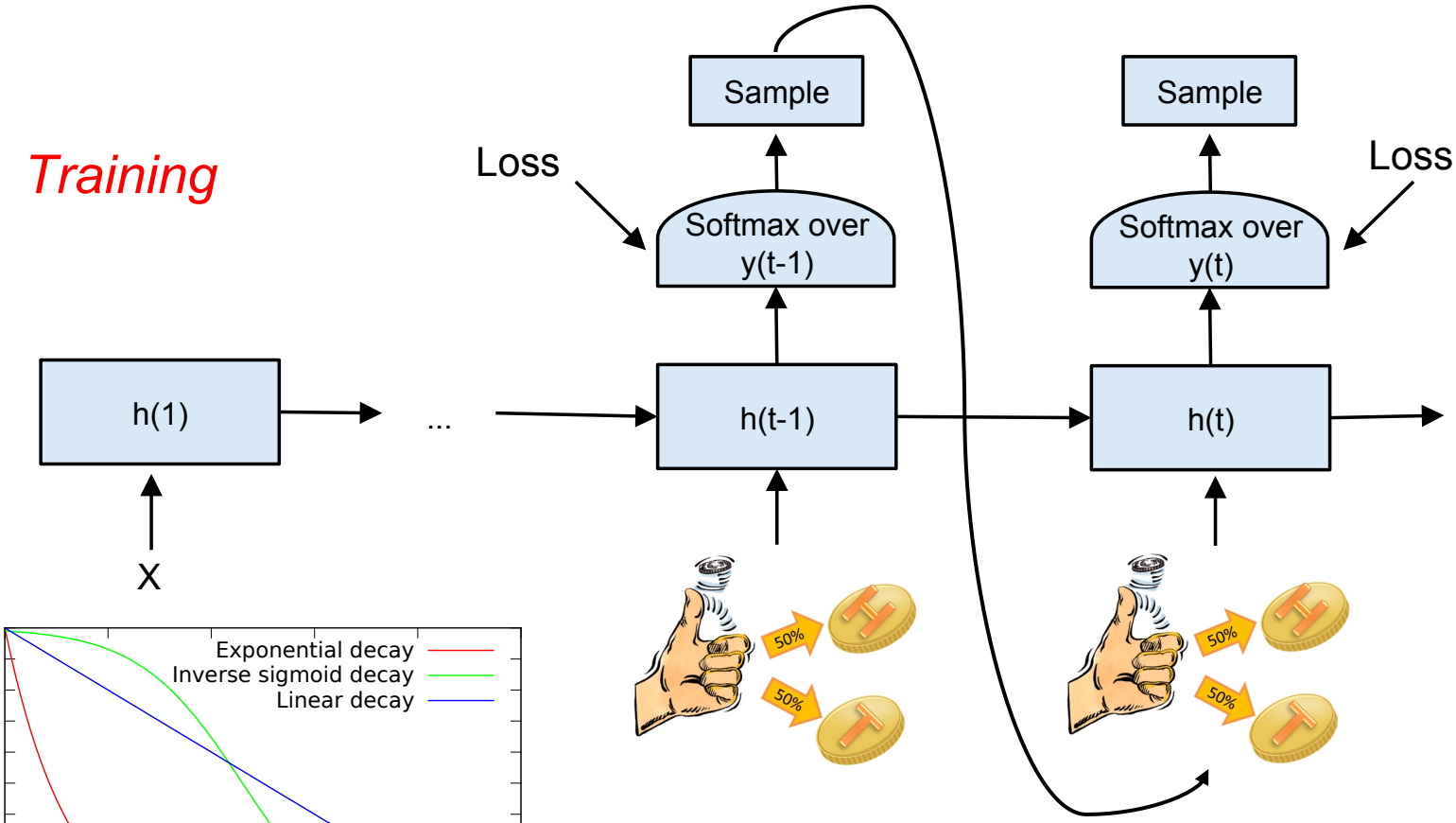
IV. Data Augmentation -> Overfitting

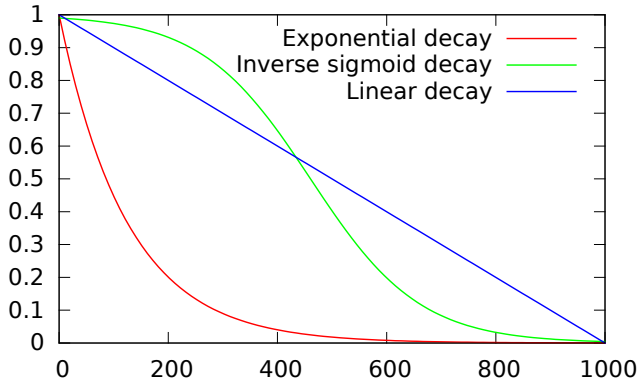


- **10** times training set
- **10%** accuracy boost



V. Scheduled Sampling



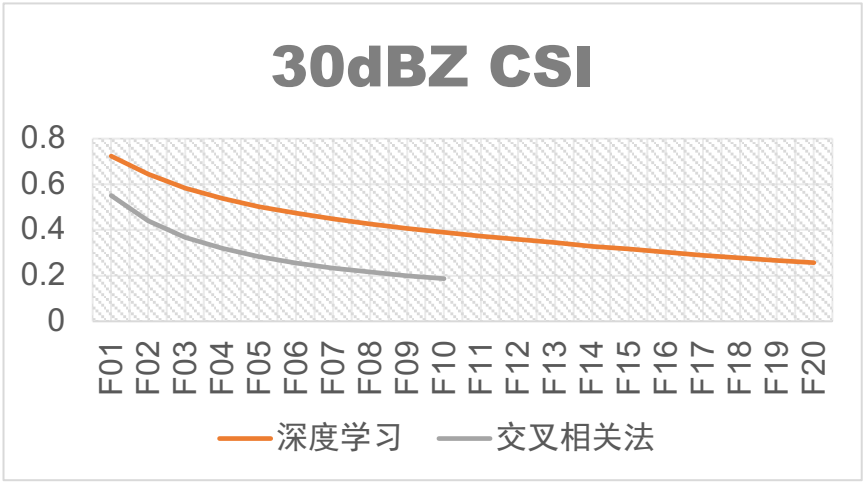


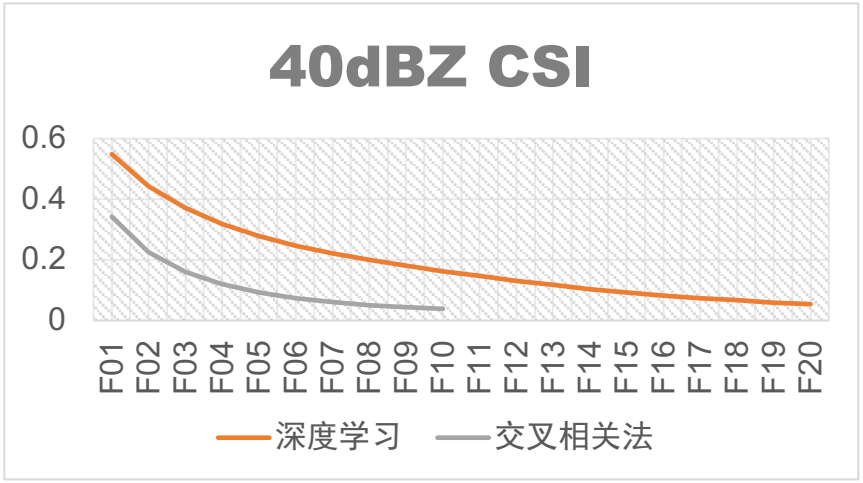
- Scheduled sampling rate
- Incremental learning (a.k.a. curriculum learning)

Bengio, S. et al. "Scheduled Sampling for Sequence Prediction with Recurrent Neural Networks." *NIPS 2015*.

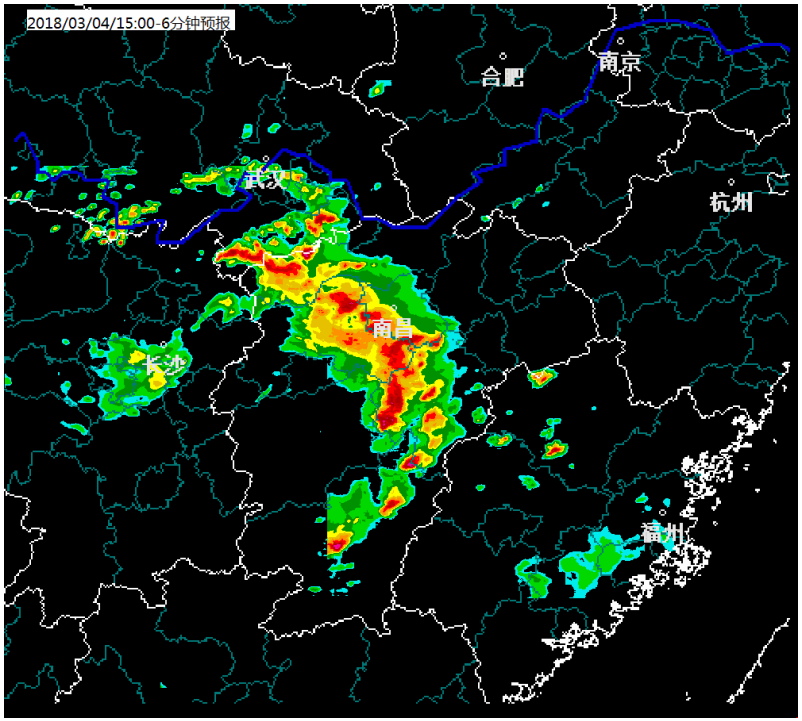
Weather Forecasting

Algorithm	Inventor	Year	MSE/frame
ConvLSTM	CUHK	2015	10.48
CDNA	OpenAI	2016	9.11
VPN	DeepMind	2017	8.49
TrajGRU	CUHK	2017	8.43
PredRNN	Tsinghua	2017	7.54
PredRNN++	Tsinghua	2018	6.38

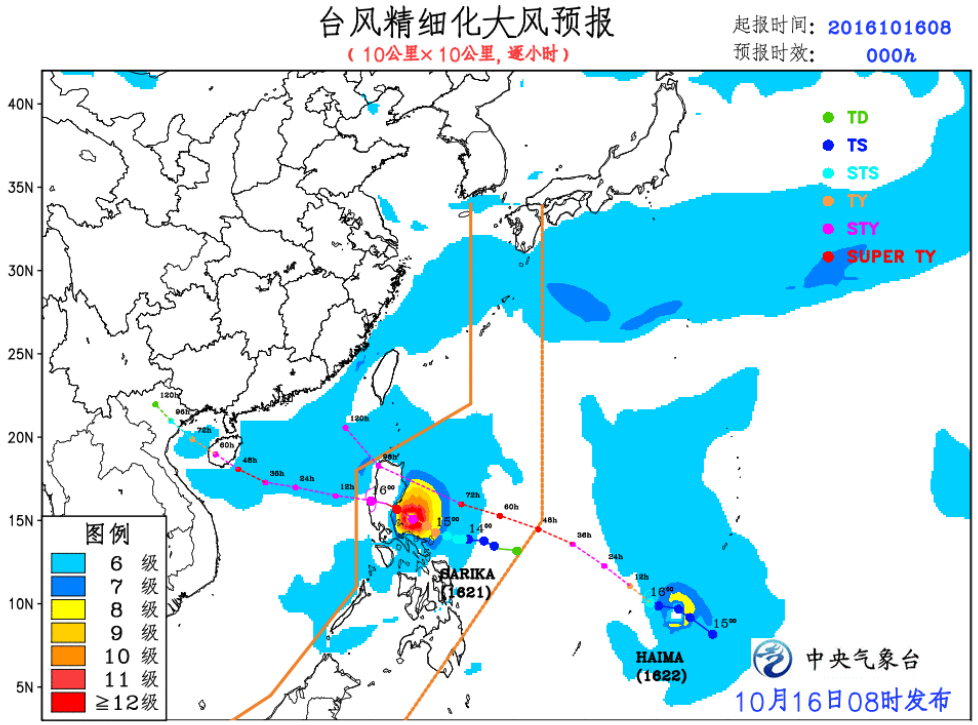




Weather Forecasting



Precipitation Nowcasting



Typhoon Monitoring

Thank You

Questions?

Mingsheng Long
mingsheng@tsinghua.edu.cn
http://ise.thss.tsinghua.edu.cn/~mlong