Deep Learning Models for Sequential Data Analysis

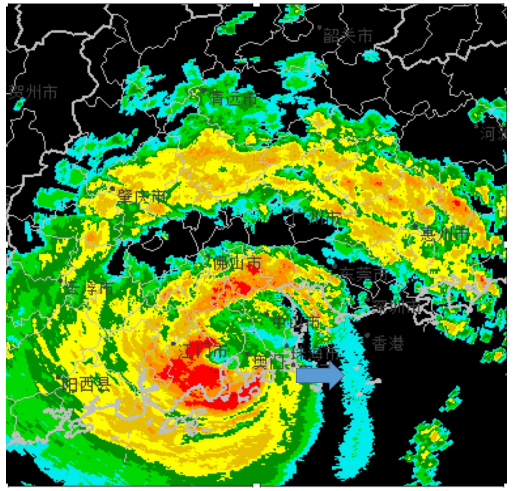
Mingsheng Long

School of Software, Tsinghua University

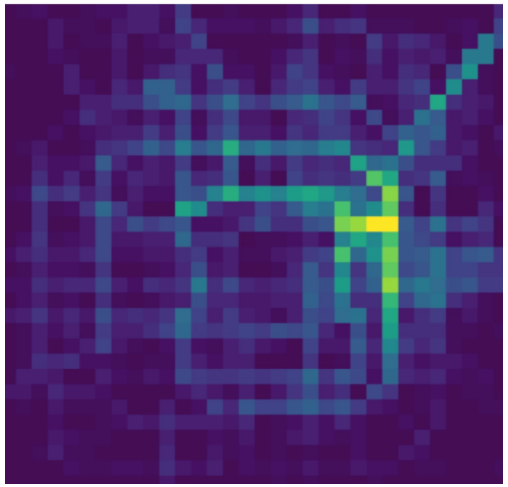
July 29, 2022

清華大學
Tsinghua University

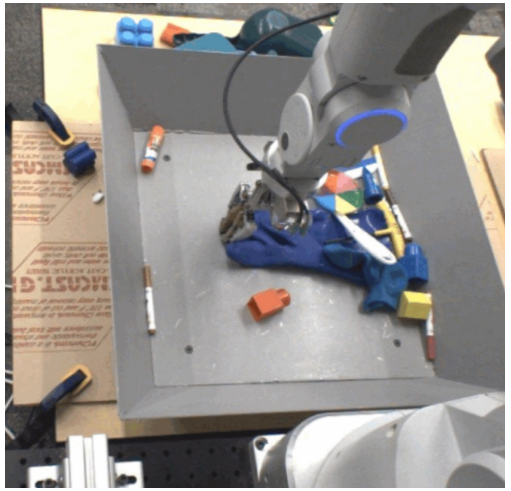
Spatiotemporal Data Analysis



Radar Echo

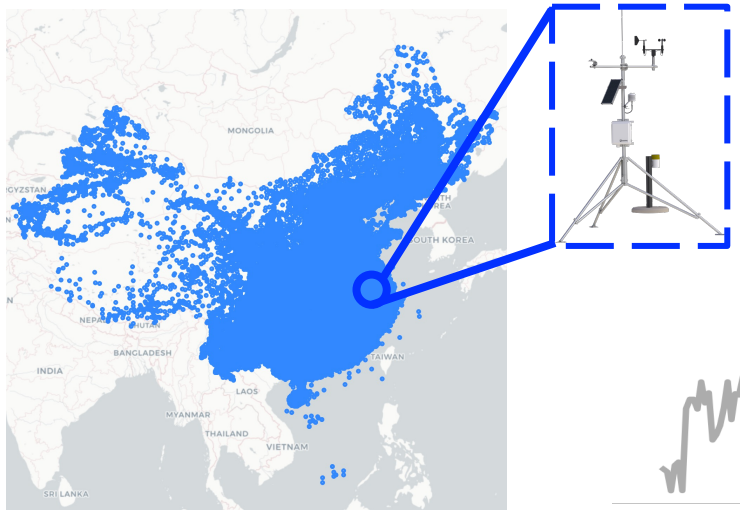


Traffic Map

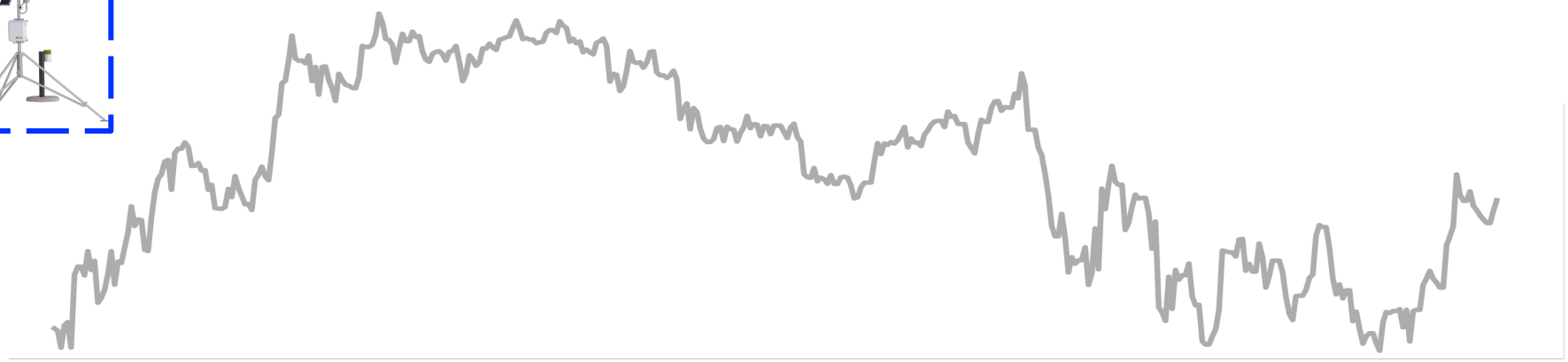


Robotics

Pedestrian Motion

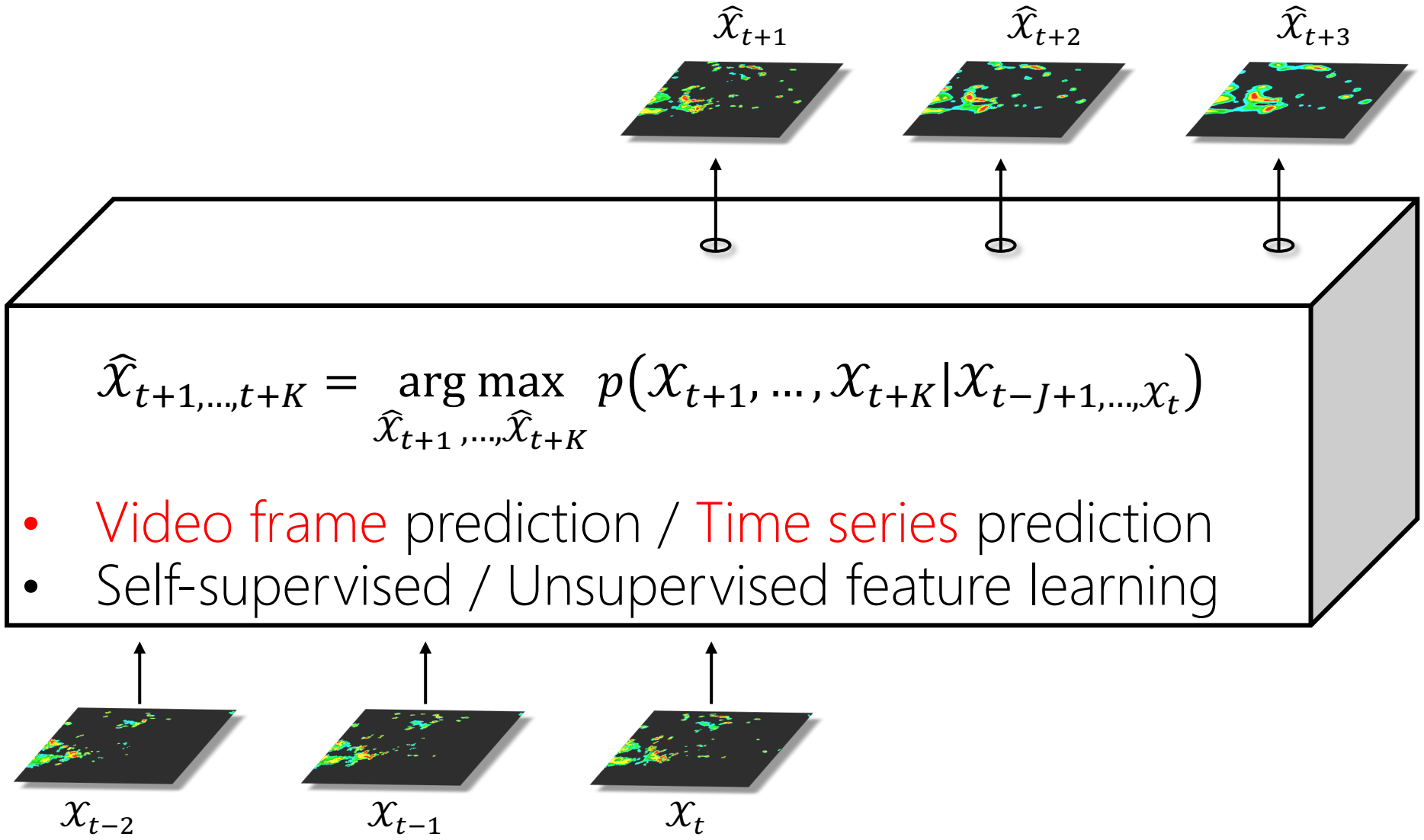


Automatic stations



Temperature, Wind speed, Precipitation…

Predictive Learning



Predictive Learning

Obstacles to Progress in AI

Y LeCun

- **Machines need to learn/understand how the world works**
  - Physical world, digital world, people,….
  - They need to acquire some level of common sense
- **They need to learn a very large amount of background knowledge**
  - Through observation and action
- **Machines need to perceive the state of the world**
  - So as to make accurate predictions and planning
- **Machines need to update and remember estimates of the state of the world**
  - Paying attention to important events. Remember relevant events
- **Machines neet to reason and plan**
  - Predict which sequence of actions will lead to a desired state of the world

Intelligence & Common Sense =

Perception + Predictive Model + Memory + Reasoning & Planning

How Much Information Does the Machine Need to Predict?

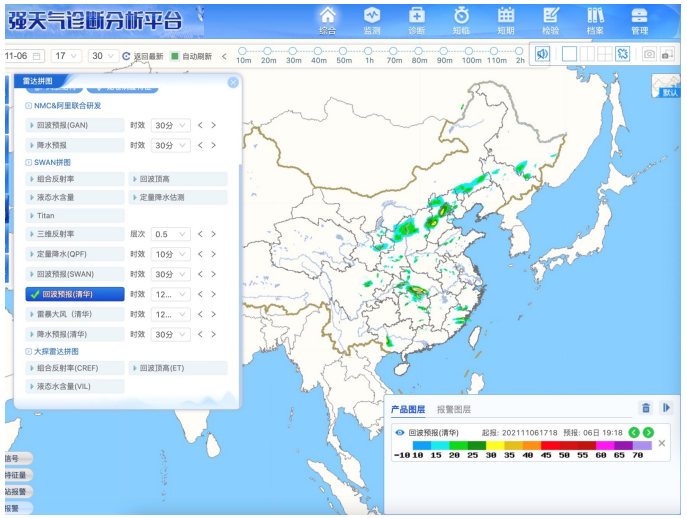
Y LeCun

- **"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
  - Predicting human-supplied data
  - **10→10,000 bits per sample**
- **Unsupervised/Predictive Learning (cake)**
  - The machine predicts any part of its input for any observed part.
  - Predicts future frames in videos
  - **Millions of bits per sample**

- **(Yes, I know, this picture is slightly offensive to RL folks. But I'll make it up)**

Credit: Yann LeCun [NIPS 2016]

Our Research



Spatiotemporal Predictive Learning

PredRNN-V2 for Precipitation Nowcasting

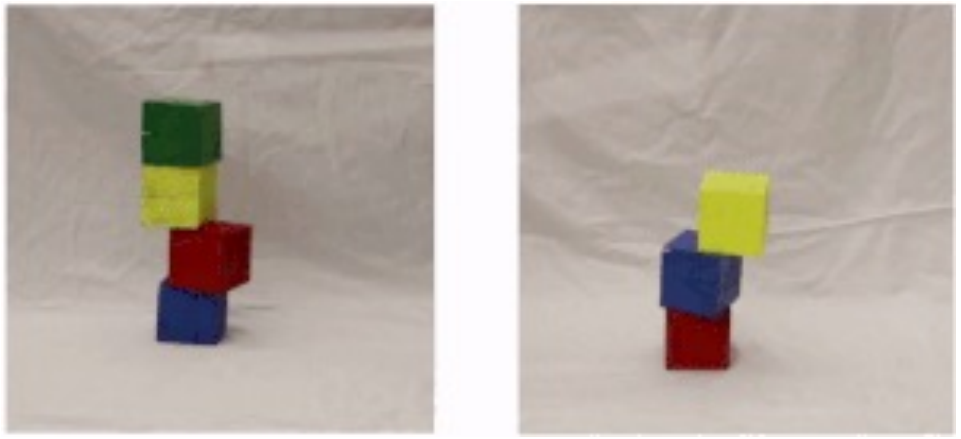
Time Series Forecasting

Autoformer for 2022 Beijing Olympics

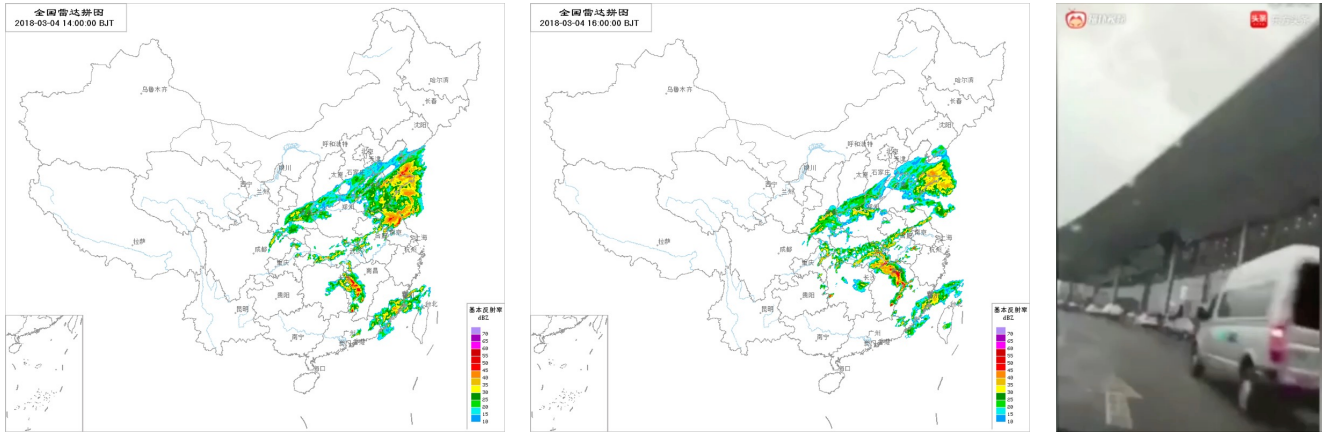
Flowformer

A Foundation Model for General Task

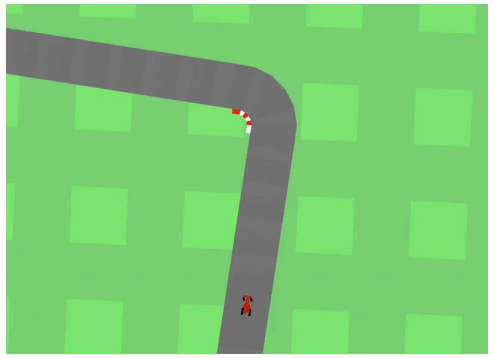
Spatiotemporal Predictive Learning

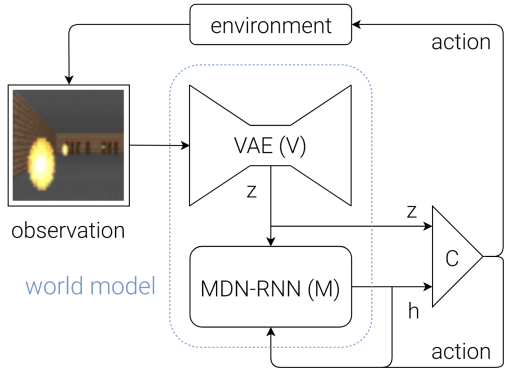


Physical world understanding
[Lerer et al. ICML16; Wu et al. NeurIPS17]

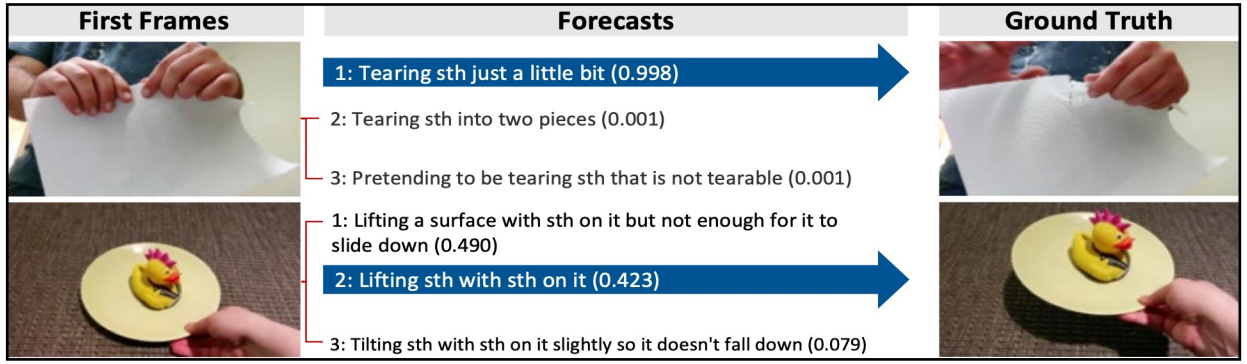


Precipitation Nowcasting
[Wang et al. NeurIPS17; CVPR19]



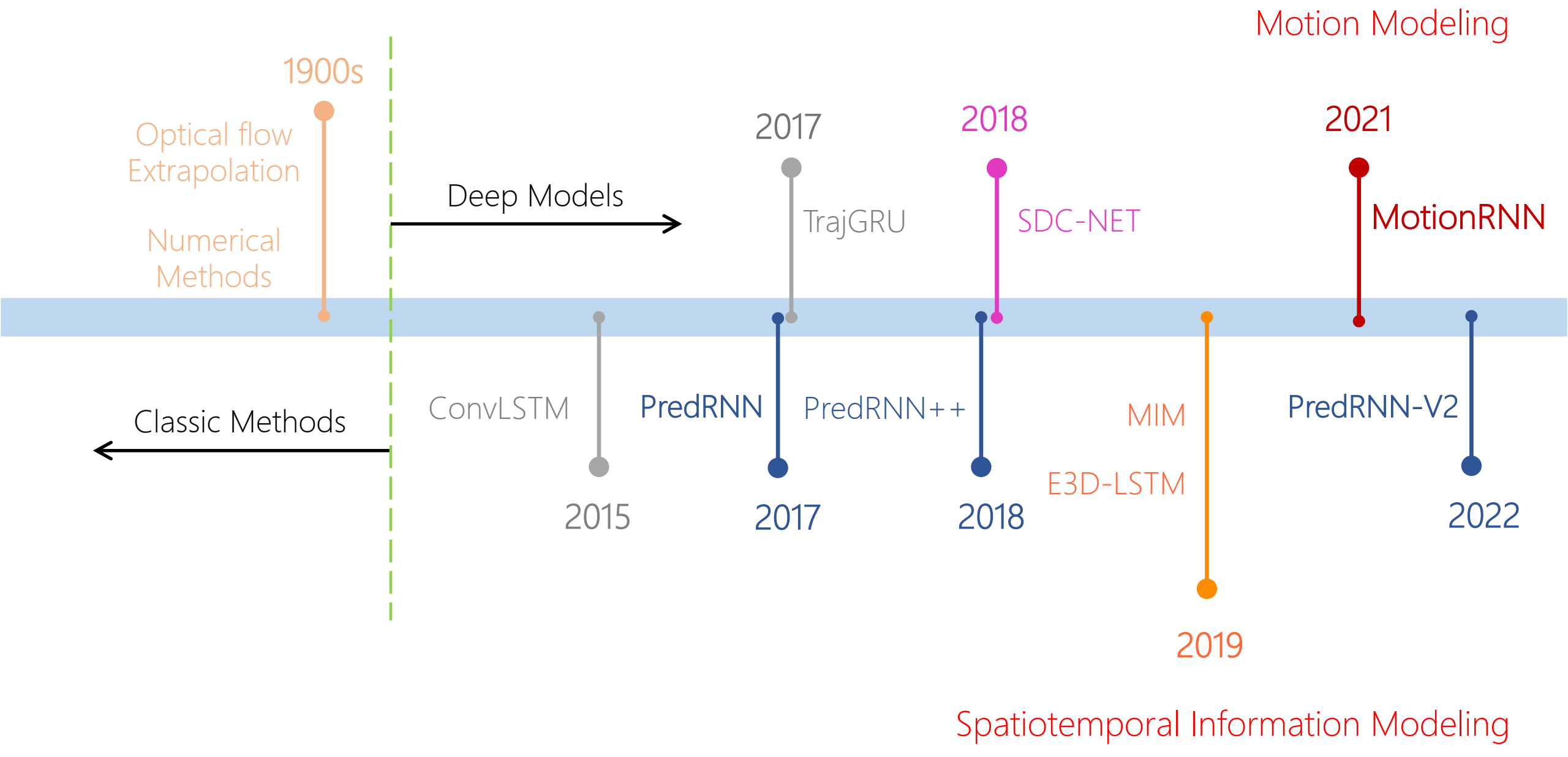


Robotics control
[Ha & Schmidhuber, NeurIPS18]



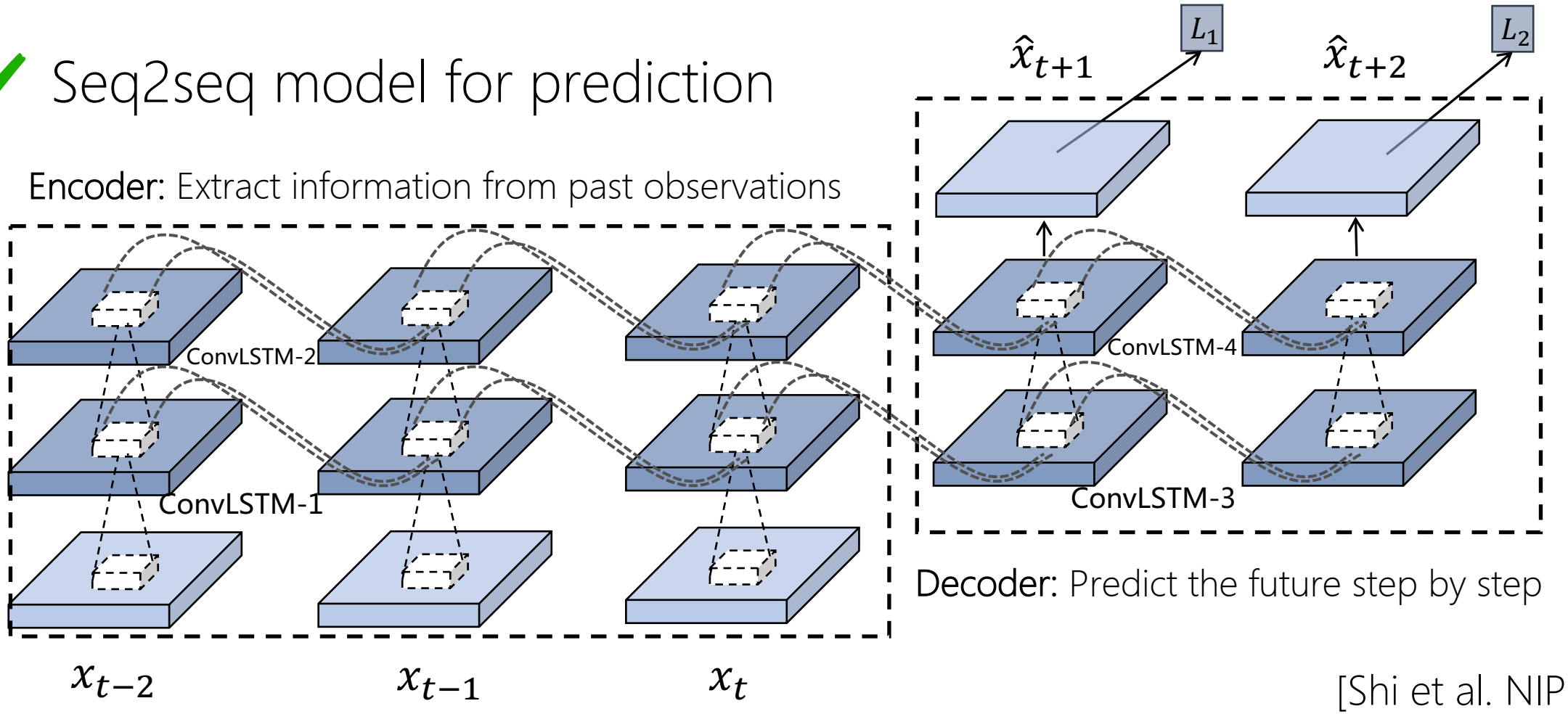
Action intention estimation
[Wang et al. ICLR19]

Timeline


Motion Modeling
1900s
Optical flow
Extrapolation
Numerical
Methods
Deep Models
2017
TrajGRU
2018
SDC-NET
2021
MotionRNN
Classic Methods
ConvLSTM
2015
PredRNN
2017
PredRNN++
2018
MIM
E3D-LSTM
2019
PredRNN-V2
2022
Spatiotemporal Information Modeling

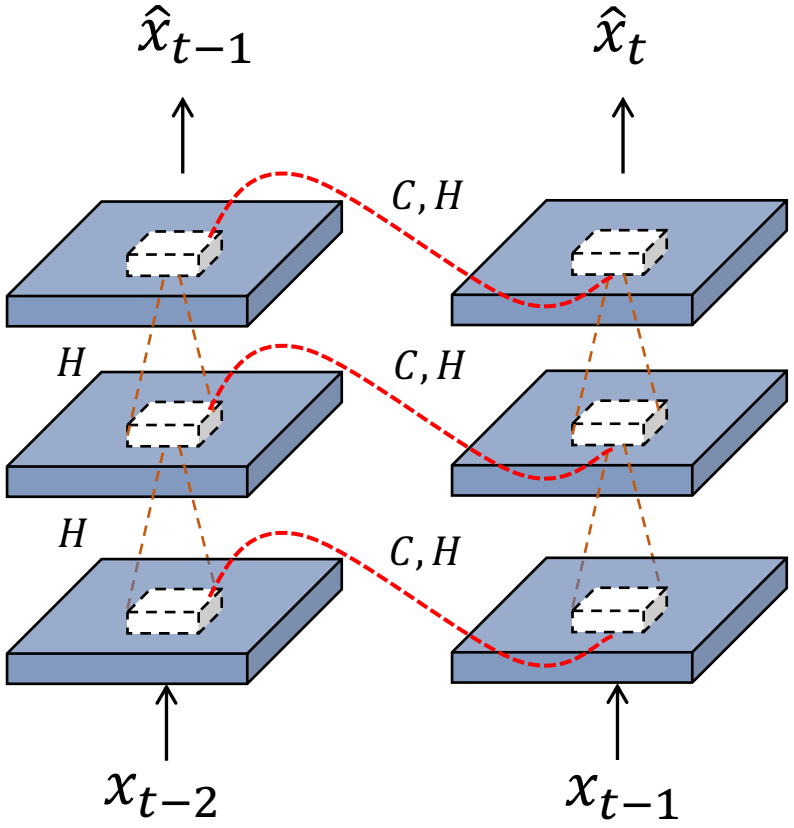
Spatiotemporal Modeling in ConvLSTM

- ✓ CNN for spatial information, RNN for temporal information
- ✓ Seq2seq model for prediction



[Shi et al. NIPS 2015]

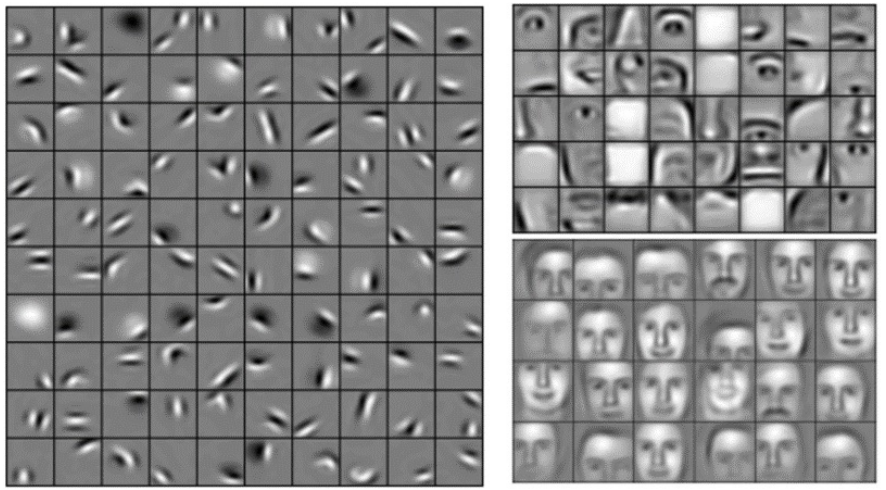
Spatiotemporal Modeling in ConvLSTM



(1) Spatial Dimension

Hierarchical

visual features



[Zeiler & Fergus. ECCV 2014]

(2) Temporal Dimension

$$g_t = \tanh(w_{xg} * X_t + w_{hg} * H_{t-1} + b_g)$$

$$i_t = \sigma(w_{xi} * X_t + w_{hi} * H_{t-1} + w_{ci} \odot C_{t-1} + b_i)$$

$$f_t = \sigma(w_{xf} * X_t + w_{hf} * H_{t-1} + w_{cf} \odot C_{t-1} + b_f)$$

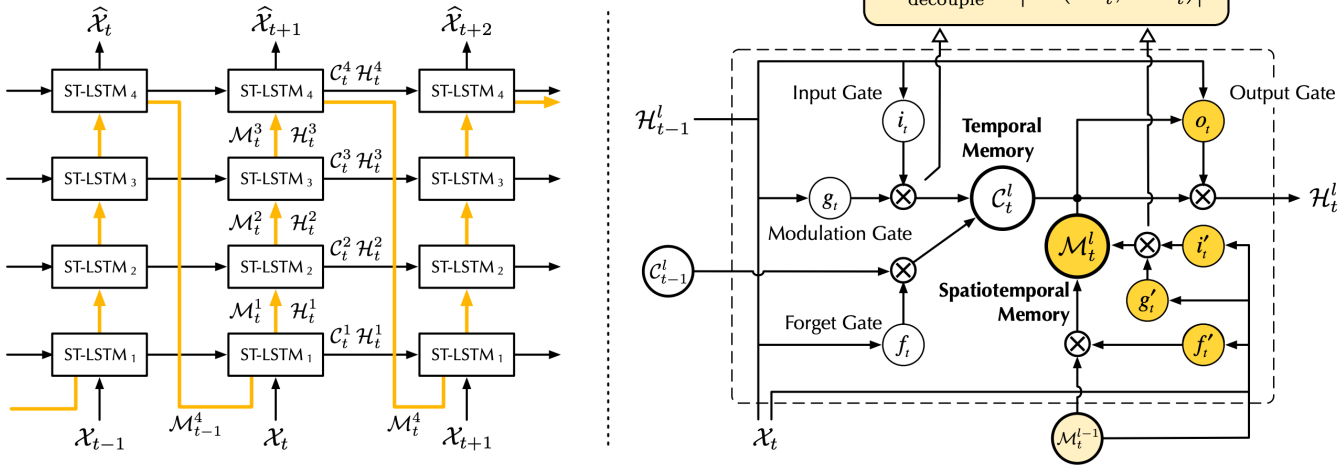
$$C_t = f_t \odot C_{t-1} + i_t \odot g_t$$

Capture the long- and short-term dynamics with C_t

$$o_t = \sigma(w_{xo} * X_t + w_{ho} * H_{t-1} + w_{co} \odot C_t + b_o)$$

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

Timeline



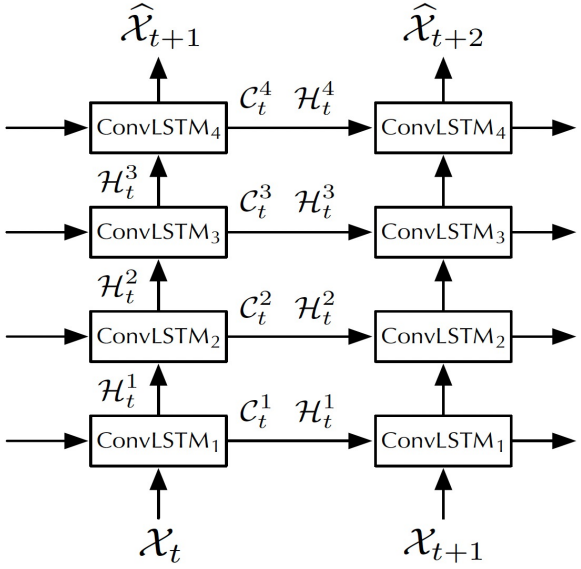
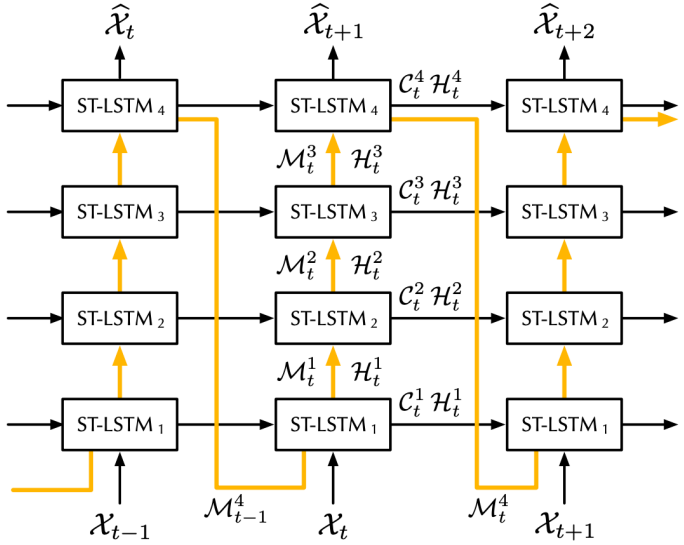
PredRNN-V2

2022

- Architecture: Spatiotemporal Memory Flow to enhance state transition
- Core Module: ST-LSTM with Memory Decoupling to cover long- and short-term dynamics
- Training Strategy: Reverse Scheduled Sampling to bridge the gap between encoder and decoder

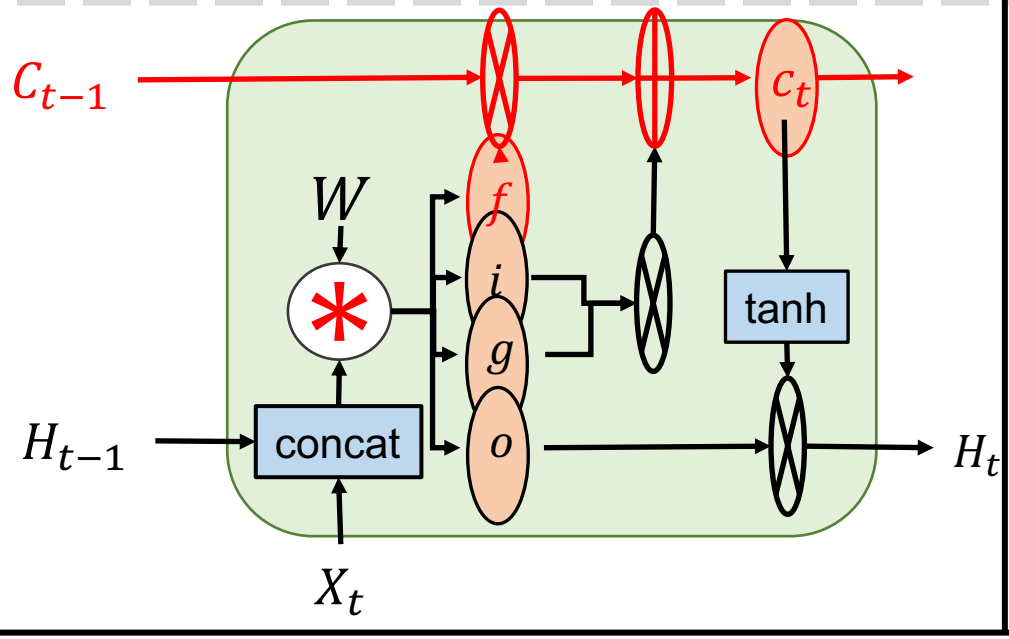
Yunbo Wang, Haixu Wu, Jianjin Zhang, Zhifeng Gao, Jianmin Wang, Philip S. Yu, Mingsheng Long✉. PredRNN: A Recurrent Neural Network for Spatiotemporal Predictive Learning. **TPAMI**, 2022.

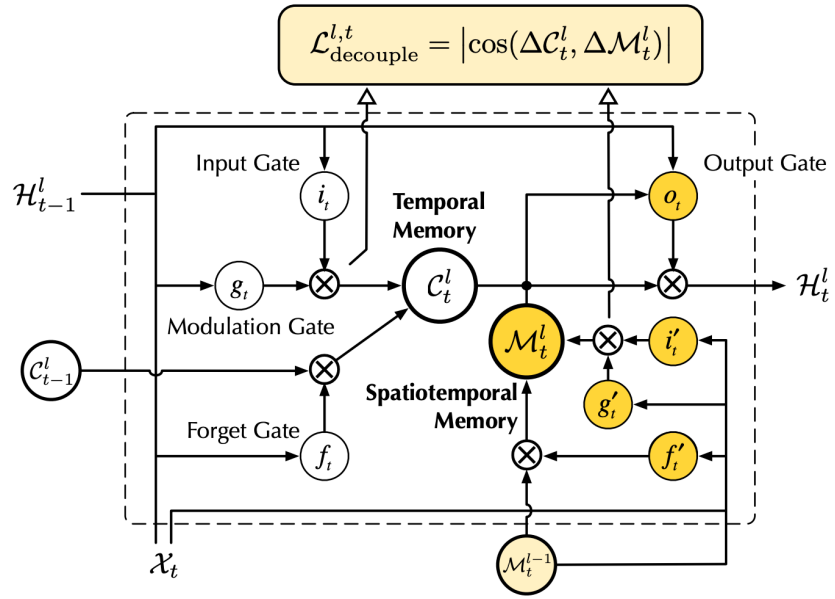
Comparison with ConvLSTM

	ConvLSTM	PredRNN-V2
State Transitions	Memory state C_t is only updated along the temporal dimension, which wastes the hierarchical visual features ✗	Spatiotemporal Memory Flow forms a zig-zag transition to unify the visual features in different layers ✓
		

Comparison with ConvLSTM

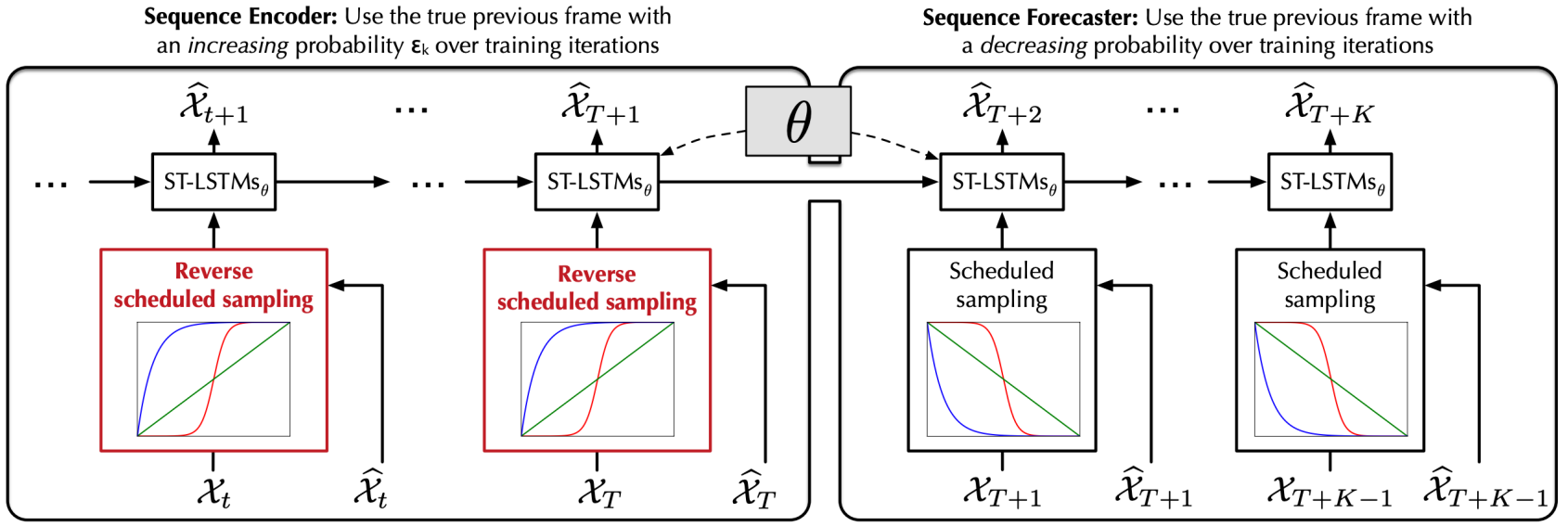
	ConvLSTM	PredRNN-V2
Long Short-term Dynamics Modeling	Memory state C_t is forced to capture long- and short-term dynamics simultaneously ✗	ST-LSTM with C_t and M_t that are jointly learned and explicitly decoupled to cover long- and short-term dynamics ✓



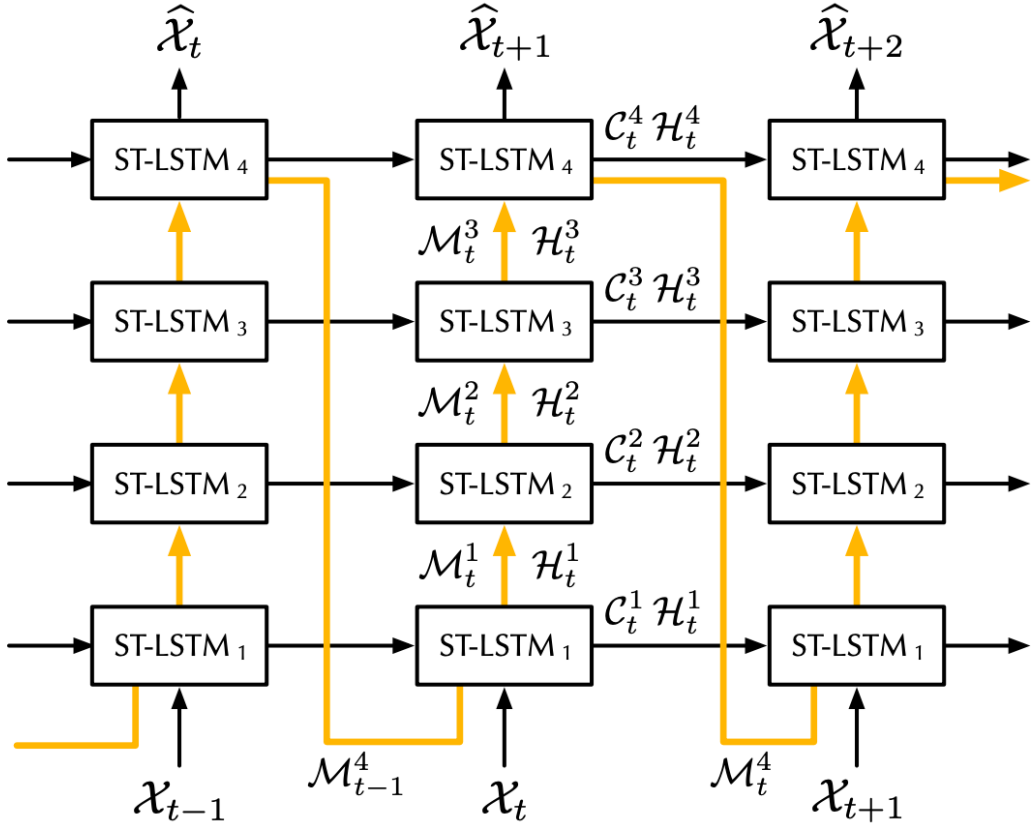


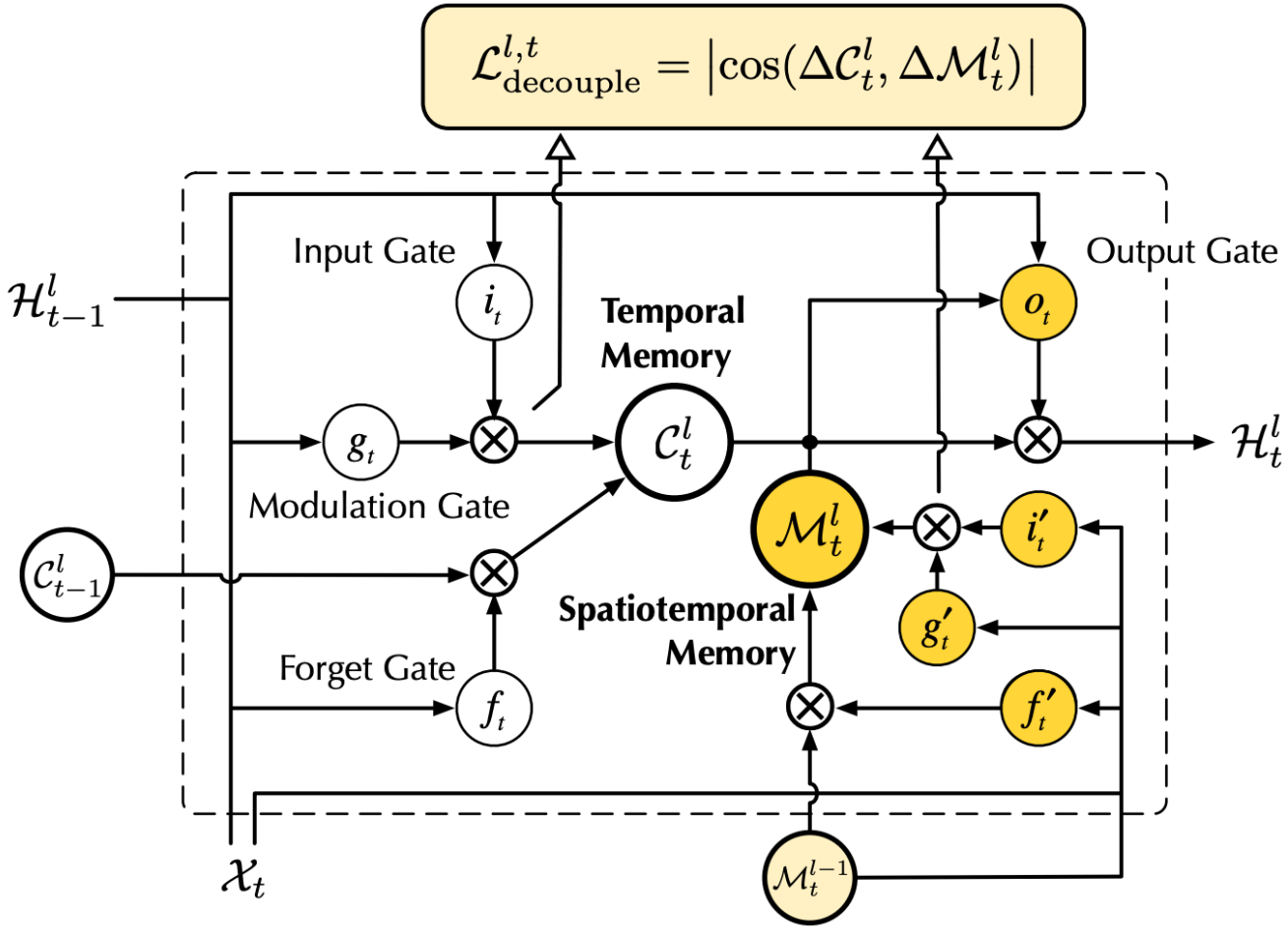
Comparison with ConvLSTM

	ConvLSTM	PredRNN-V2
Training Strategy	The encoder and decoder share the same parameter but their inputs are mismatched ✗	Reverse Scheduled Sampling to bridge the gap between encoder and decoder ✓



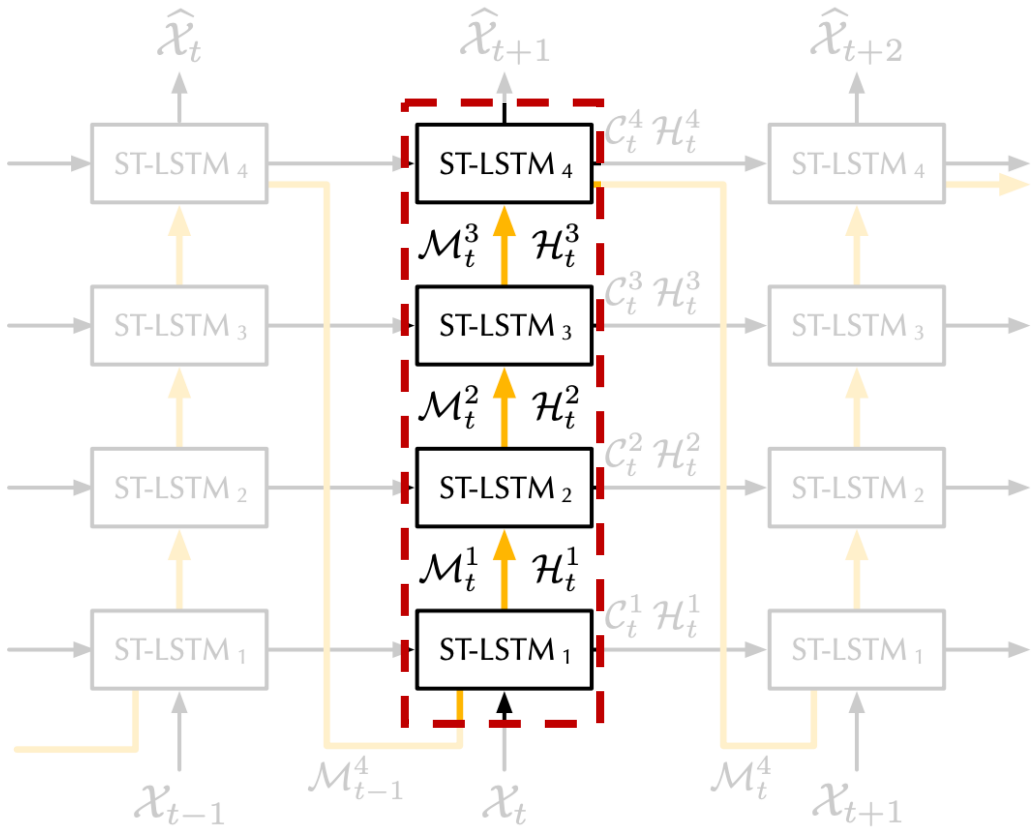
PredRNN-V2





Spatiotemporal Memory M_t: Transit in a (1) vertical and (2) zig-zag way

PredRNN-V2



(1) Vertical Transition

Unify the hierarchical visual features

$$g_t = \tanh(w_{xg} * X_t \mathbb{1}_{\{l=1\}} + w_{hg} * H_t^{l-1} + b_g)$$

$$i_t = \sigma(w_{xi} * X_t \mathbb{1}_{\{l=1\}} + w_{hi} * H_t^{l-1} + w_{ci} * M_t^{l-1} + b_i)$$

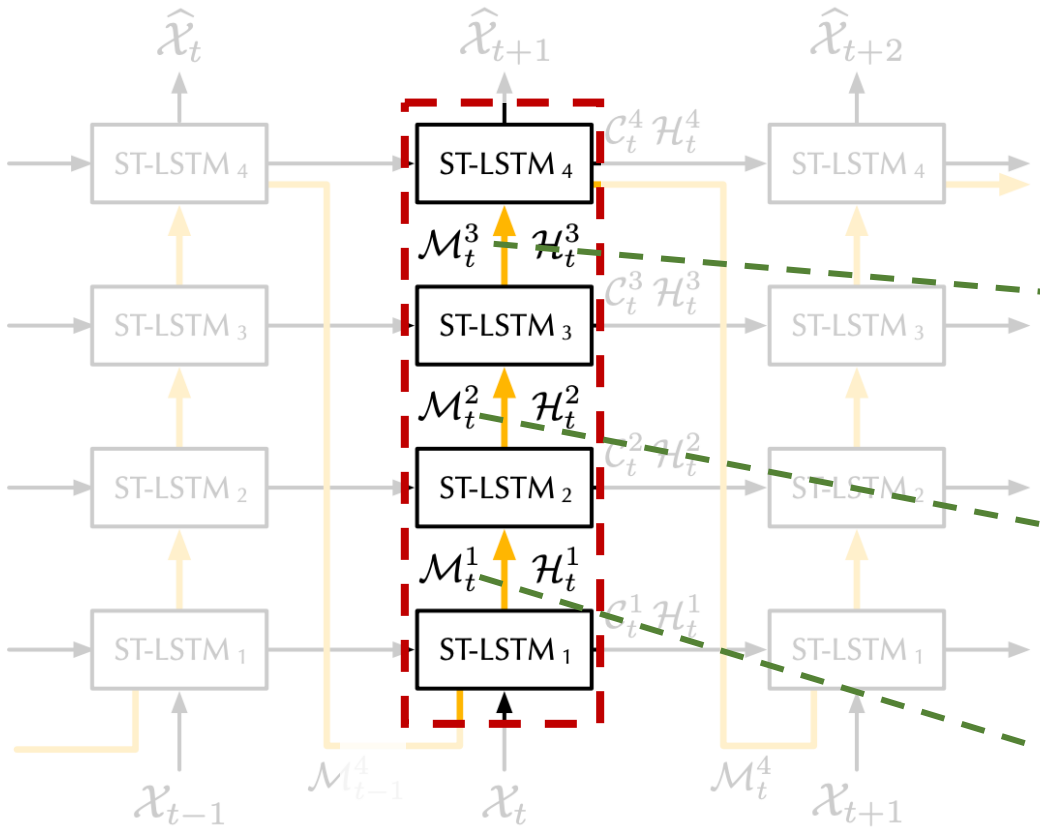
$$f_t = \sigma(w_{xf} * X_t \mathbb{1}_{\{l=1\}} + w_{hf} * H_t^{l-1} + w_{cf} * M_t^{l-1} + b_f)$$

$$M_t^l = f_t \odot M_t^{l-1} + i_t \odot g_t$$

$$o_t = \sigma(w_{xo} * X_t \mathbb{1}_{\{l=1\}} + w_{ho} * H_t^{l-1} + w_{co} * M_t^l + b_o)$$

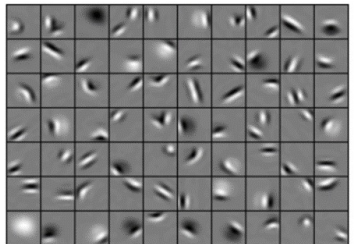
$$H_t^l = o_t \odot \tanh(M_t^l),$$

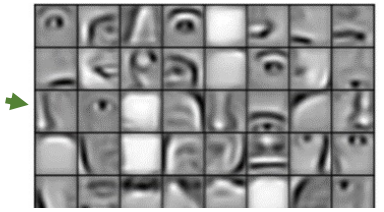
PredRNN-V2

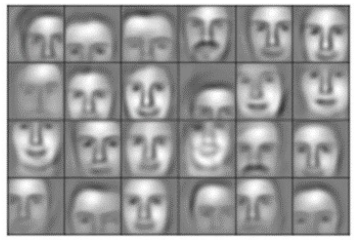


(1) Vertical Transition

Unify the hierarchical visual features





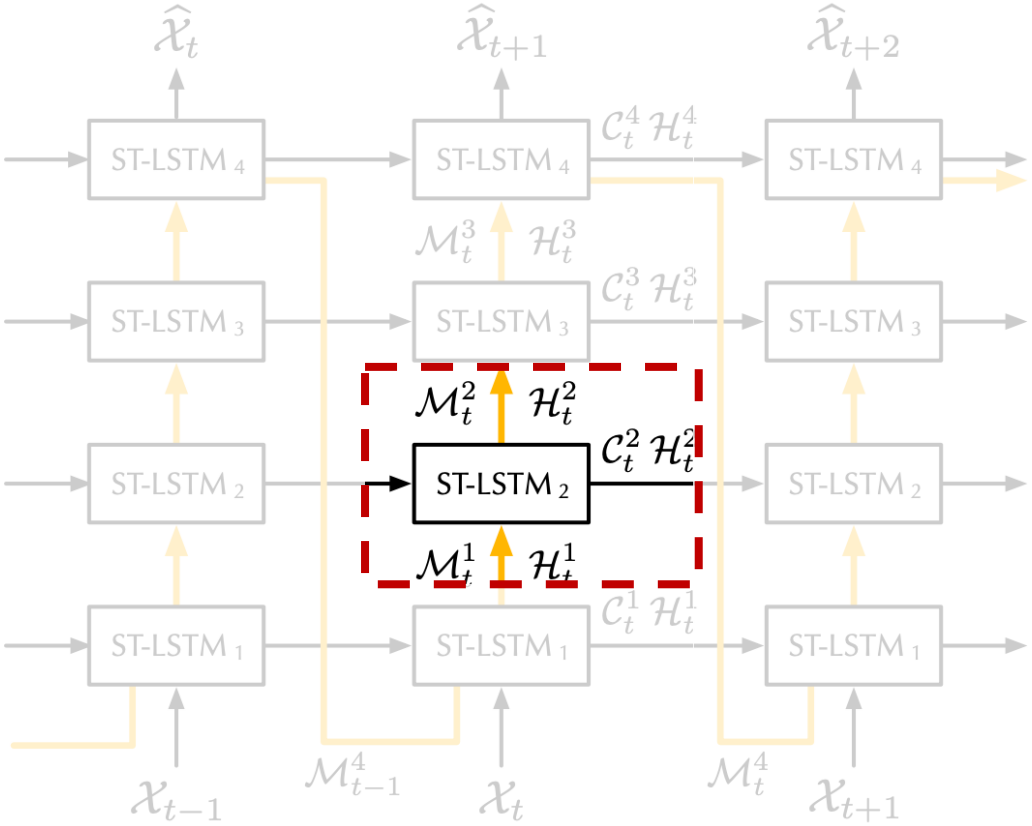


High-level feature

Gated Aggregation

Low-level feature

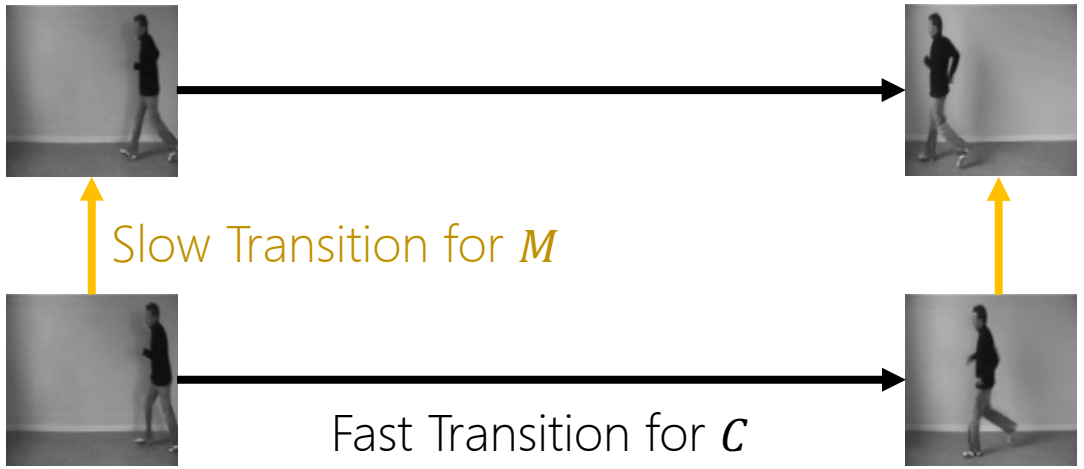
PredRNN-V2



(2) Zig-zag Transition

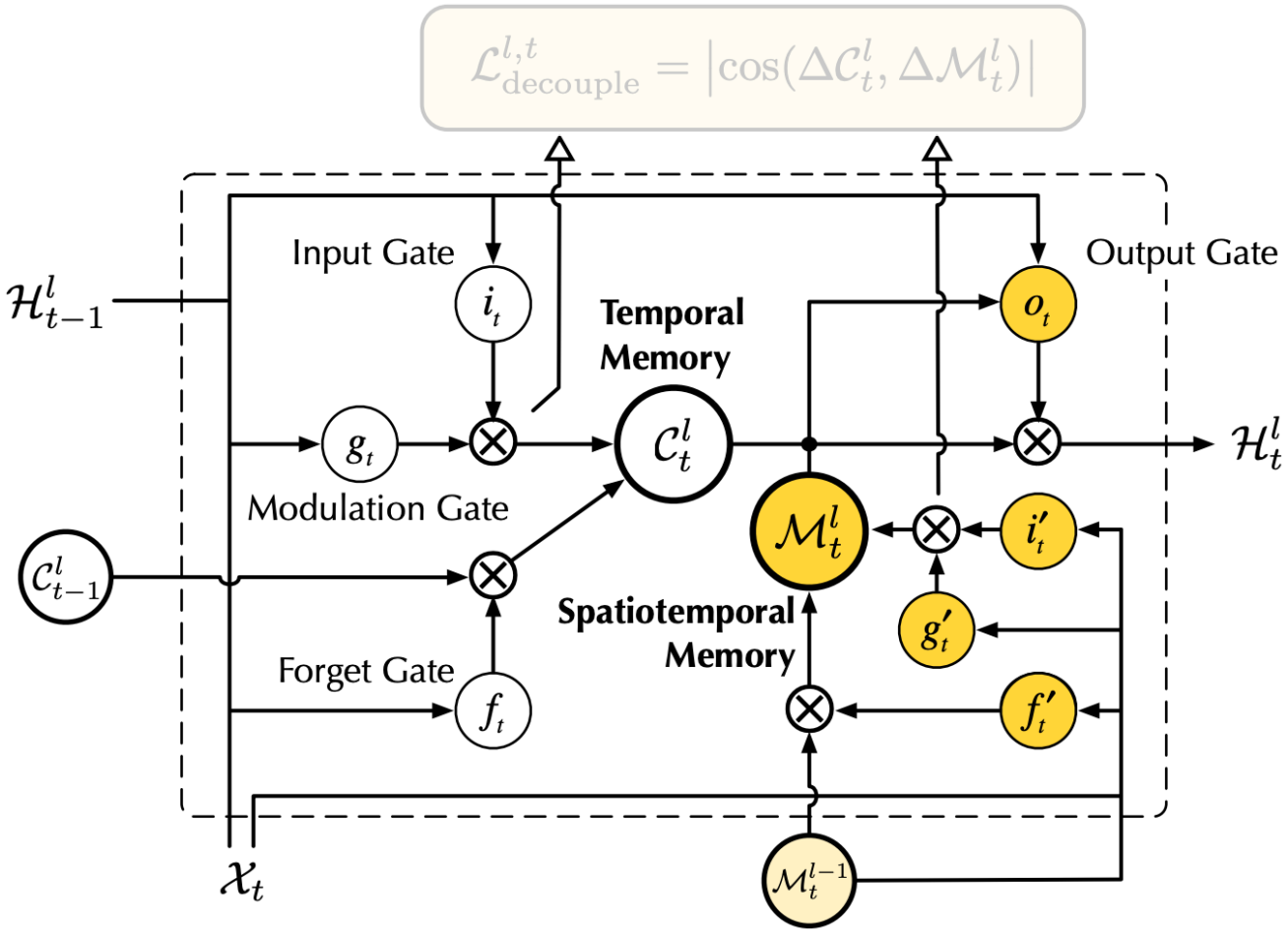
M is in "Slow transition" → short-term dynamics

C is in "Fast transition" → long-term dynamics



Achieve the long- and short-term dynamics decomposition.

PredRNN-V2



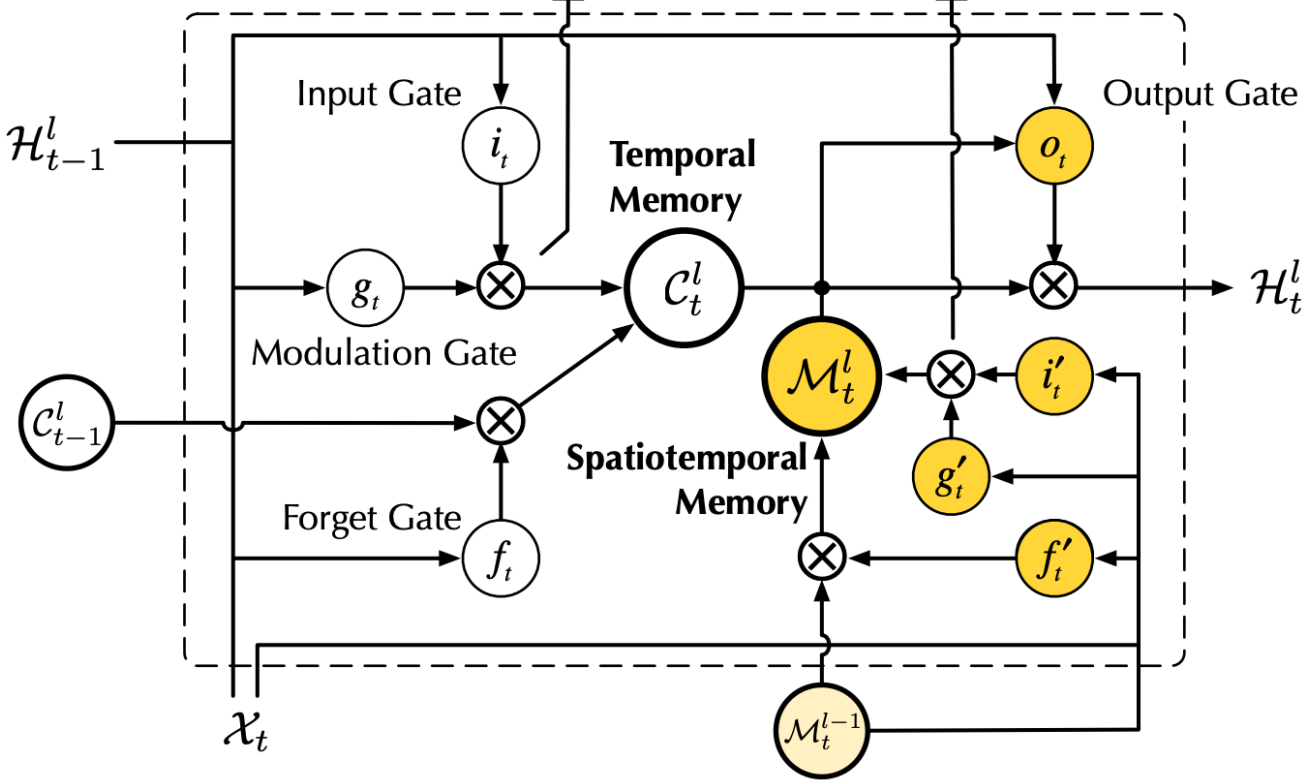
$$g_t = \tanh(W_{xg} * \mathcal{X}_t + W_{hg} * \mathcal{H}^l_{t-1})$$
$$i_t = \sigma(W_{xi} * \mathcal{X}_t + W_{hi} * \mathcal{H}^l_{t-1})$$
$$f_t = \sigma(W_{xf} * \mathcal{X}_t + W_{hf} * \mathcal{H}^l_{t-1})$$
$$\mathcal{C}^l_t = f_t \odot \mathcal{C}^l_{t-1} + i_t \odot g_t$$ long-term dynamics
$$g'_t = \tanh(W'_{xg} * \mathcal{X}_t + W_{mg} * \mathcal{M}^{l-1}_t)$$
$$i'_t = \sigma(W'_{xi} * \mathcal{X}_t + W_{mi} * \mathcal{M}^{l-1}_t)$$
$$f'_t = \sigma(W'_{xf} * \mathcal{X}_t + W_{mf} * \mathcal{M}^{l-1}_t)$$
$$\mathcal{M}^l_t = f'_t \odot \mathcal{M}^{l-1}_t + i'_t \odot g'_t$$ Short-term dynamics
$$o_t = \sigma(W_{xo} * \mathcal{X}_t + W_{ho} * \mathcal{H}^l_{t-1} + W_{co} * \mathcal{C}^l_t + W_{mo} * \mathcal{M}^l_t)$$
$$\mathcal{H}^l_t = o_t \odot \tanh(W_{1\times1} * [\mathcal{C}^l_t, \mathcal{M}^l_t]).$$

Can $\mathcal{C}$ and $\mathcal{M}$ be jointly learned as we expected?

PredRNN-V2

$$\mathcal{L}_{\text{decouple}}^{l,t} = \left|\cos(\Delta\mathcal{C}_t^l, \Delta\mathcal{M}_t^l)\right|$$

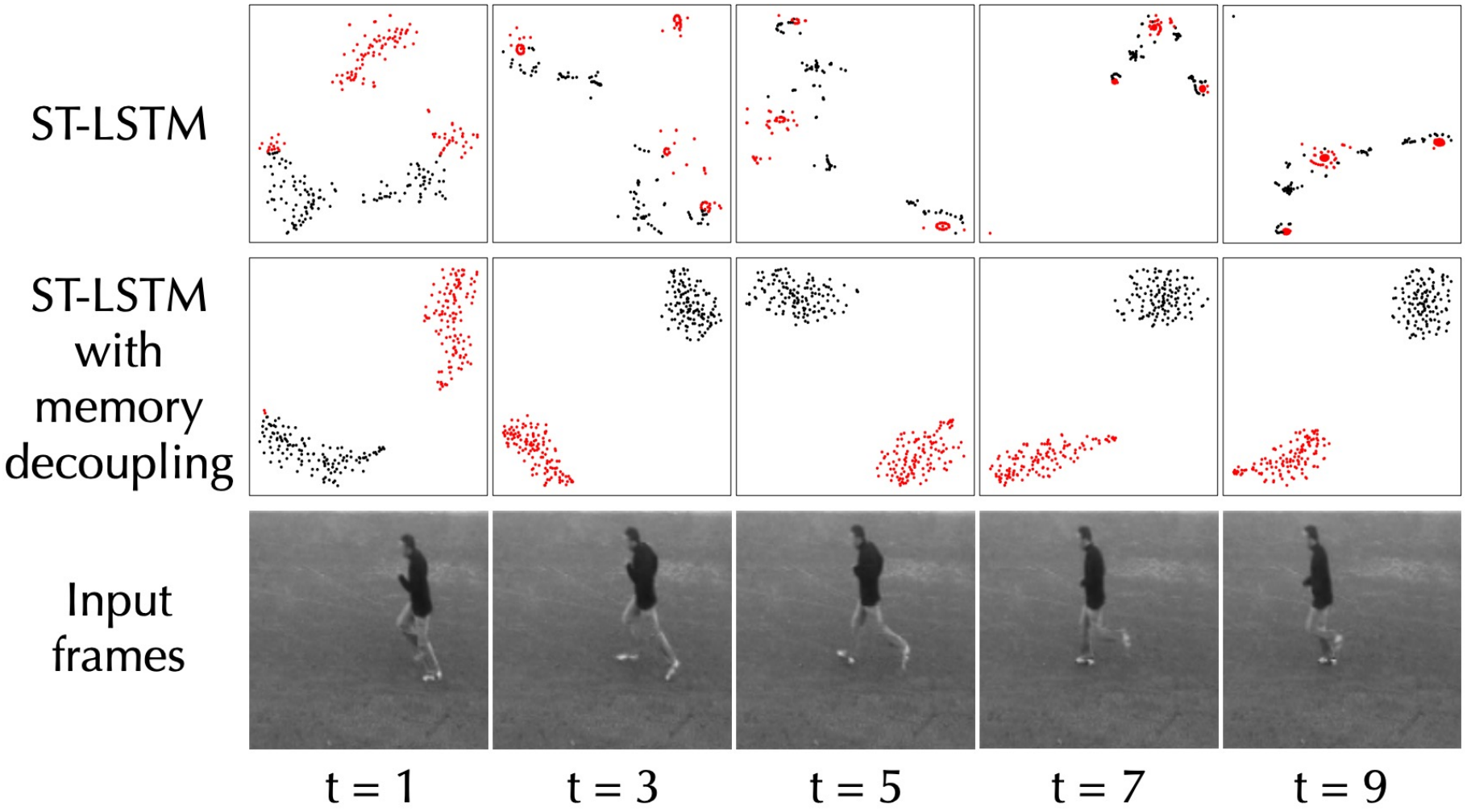
$$\Delta\mathcal{C}_t^l = W_{\text{decouple}} * (i_t \odot g_t)$$
$$\Delta\mathcal{M}_t^l = W_{\text{decouple}} * (i'_t \odot g'_t)$$
$$\mathcal{L}_{\text{decouple}} = \sum_t \sum_l \sum_c \frac{|\langle \Delta\mathcal{C}_t^l, \Delta\mathcal{M}_t^l \rangle_c|}{\|\Delta\mathcal{C}_t^l\|_c \cdot \|\Delta\mathcal{M}_t^l\|_c}$$



$$g_t = \tanh(W_{xg} * \mathcal{X}_t + W_{hg} * \mathcal{H}_{t-1}^l)$$
$$i_t = \sigma(W_{xi} * \mathcal{X}_t + W_{hi} * \mathcal{H}_{t-1}^l)$$
$$f_t = \sigma(W_{xf} * \mathcal{X}_t + W_{hf} * \mathcal{H}_{t-1}^l)$$
$$\mathcal{C}_t^l = f_t \odot \mathcal{C}_{t-1}^l + i_t \odot g_t$$ long-term dynamics
$$g'_t = \tanh(W'_{xg} * \mathcal{X}_t + W_{mg} * \mathcal{M}_t^{l-1})$$
$$i'_t = \sigma(W'_{xi} * \mathcal{X}_t + W_{mi} * \mathcal{M}_t^{l-1})$$
$$f'_t = \sigma(W'_{xf} * \mathcal{X}_t + W_{mf} * \mathcal{M}_t^{l-1})$$
$$\mathcal{M}_t^l = f'_t \odot \mathcal{M}_t^{l-1} + i'_t \odot g'_t$$ Short-term dynamics
$$o_t = \sigma(W_{xo} * \mathcal{X}_t + W_{ho} * \mathcal{H}_{t-1}^l + W_{co} * \mathcal{C}_t^l + W_{mo} * \mathcal{M}_t^l)$$
$$\mathcal{H}_t^l = o_t \odot \tanh(W_{1\times1} * [\mathcal{C}_t^l, \mathcal{M}_t^l]).$$

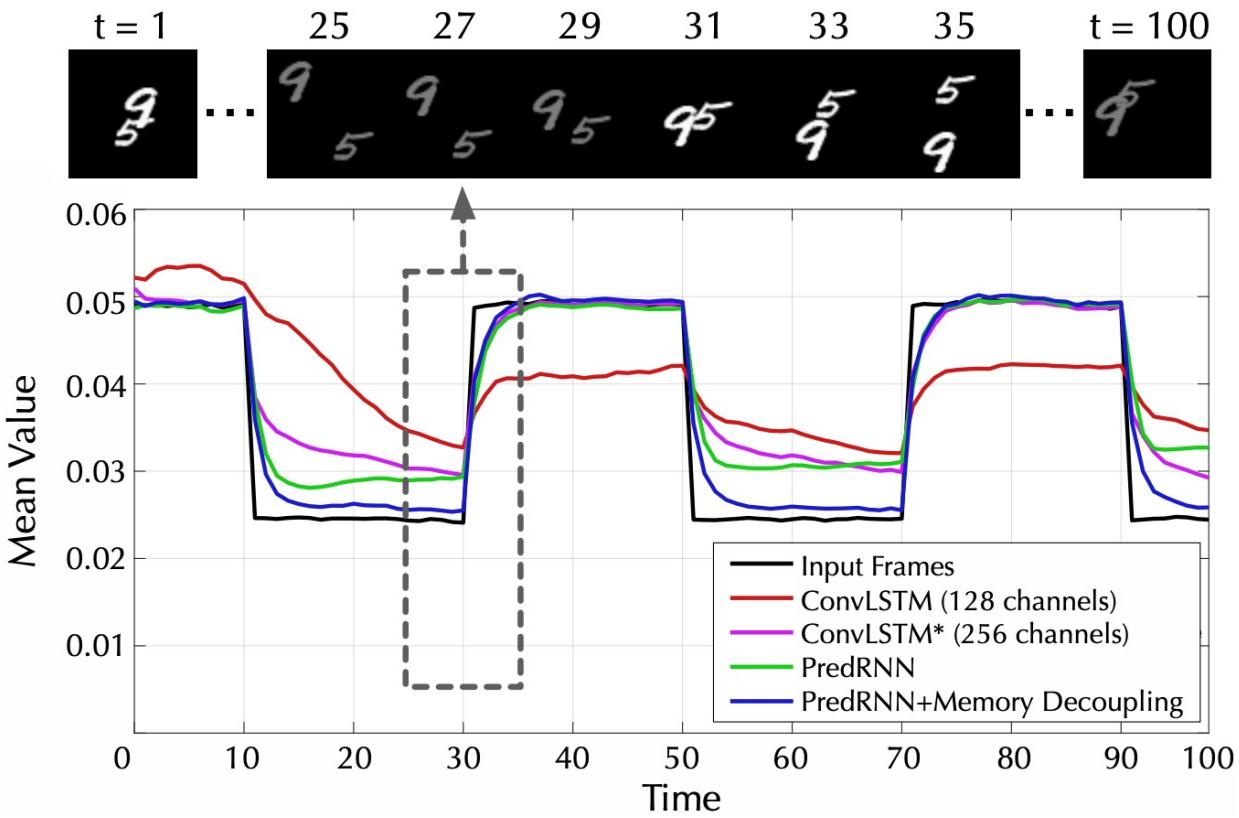
PredRNN-V2

Red points: ΔC; Black Points ΔM

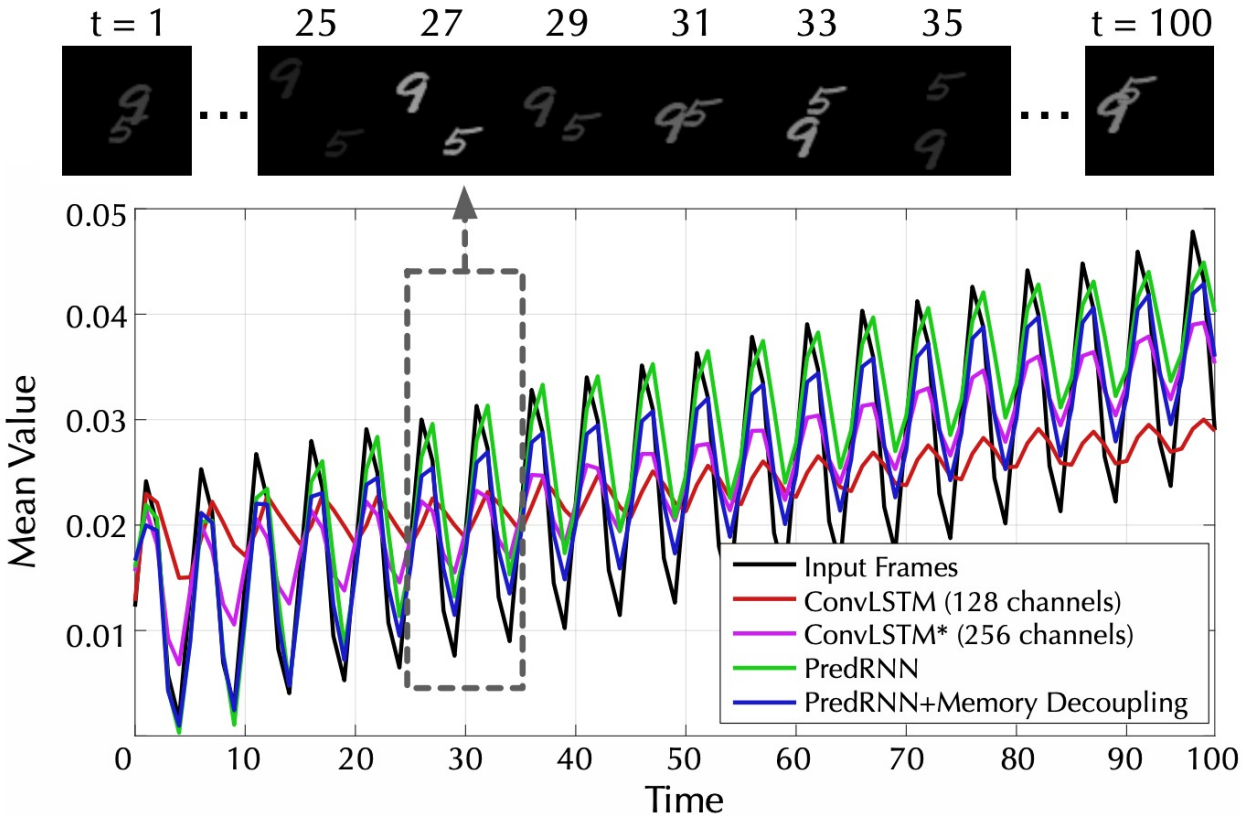


Successfully decouple C and M, release the model prediction capability.

PredRNN-V2



(a) Responses to sudden step changes



(b) Responses to long-term and short-term dynamics

Memory decoupling empowers the model with

stronger modeling capability in long- and short-term dynamics

PredRNN-V2

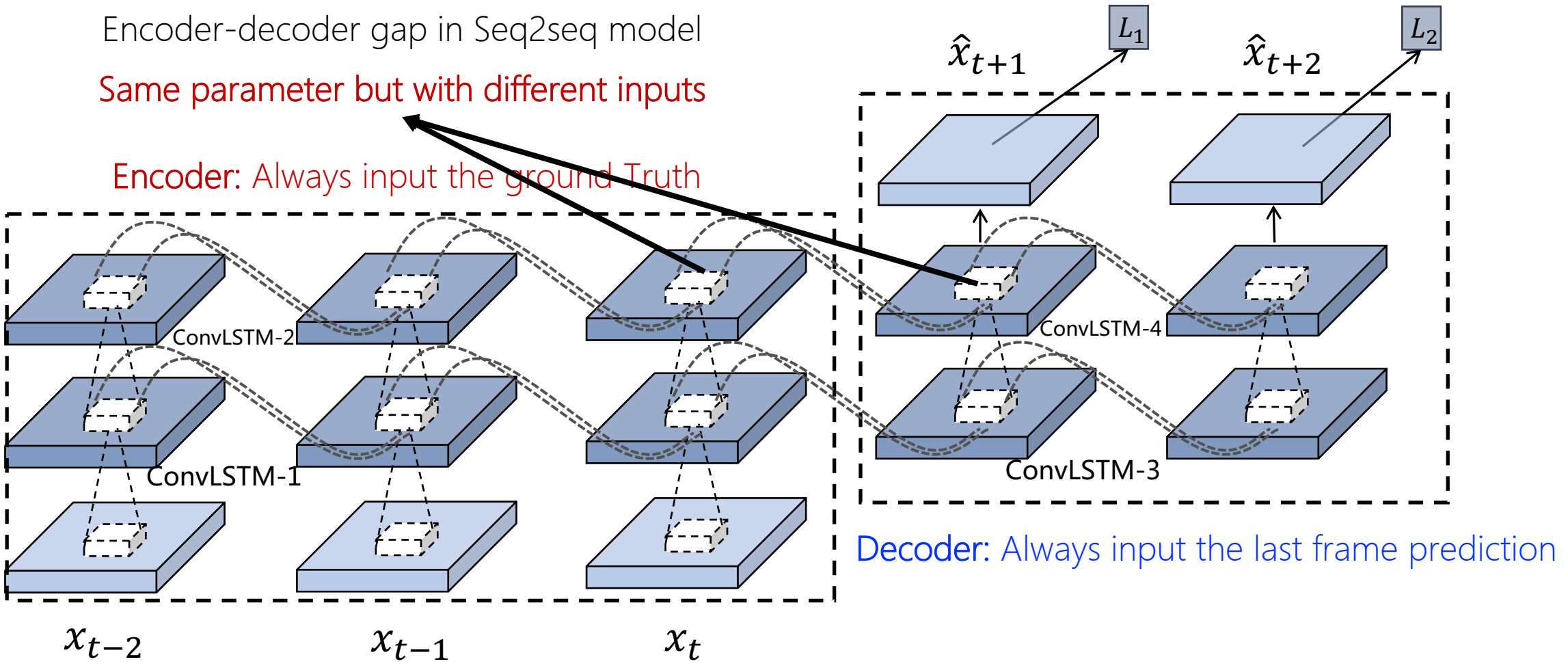
Model	Training (Encoder + Decoder)
ConvLSTM [1]	Standard [23] + Standard [23]
TrajGRU [25]	Standard + Standard
CDNA [26]	Standard + SS [24]
PredRNN [2] (*Conf. version*)	Standard + SS
PredRNN++ [27]	Standard + SS
E3D-LSTM [9]	Standard + SS
CrevNet [28]	Standard + Standard
Conv-TT-LSTM [29]	Standard + SS
PredRNN-V2 (*Ours*)	Reverse scheduled sampling + SS

Training Process of a Seq2Seq model

Encoder-decoder gap in Seq2seq model

Same parameter but with different inputs

Encoder: Always input the ground Truth

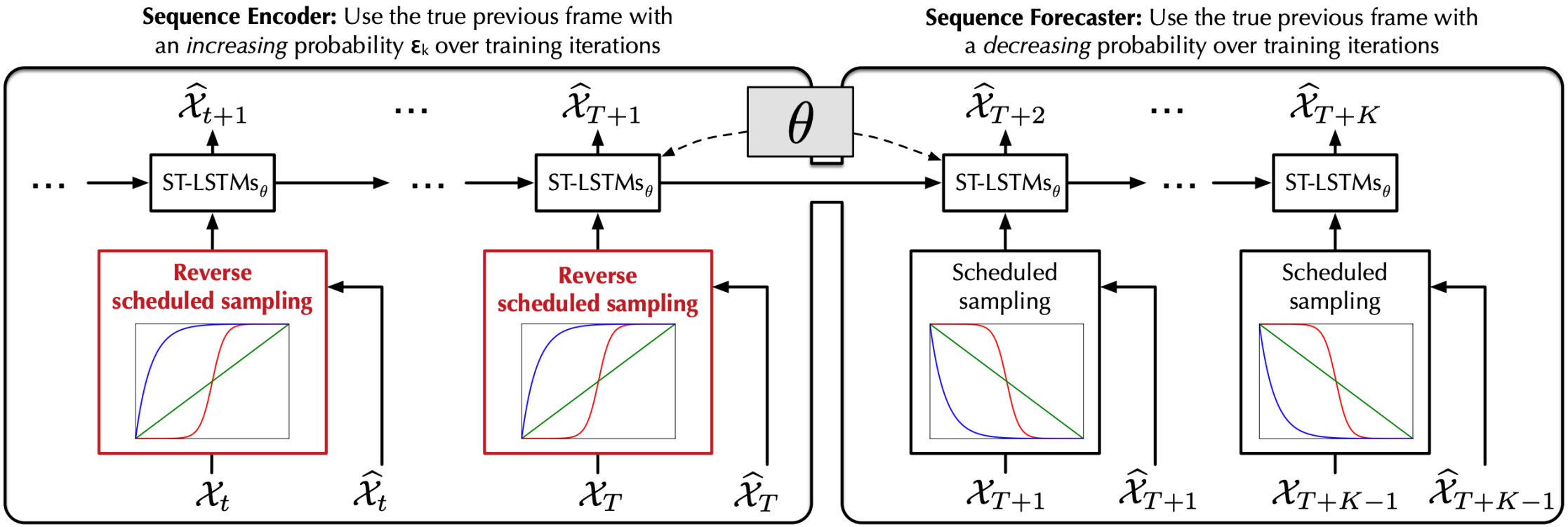


Decoder: Always input the last frame prediction

PredRNN-V2

✔ Encoder inputs: last frame prediction $\hat{x}_t$ → ground truth x_t

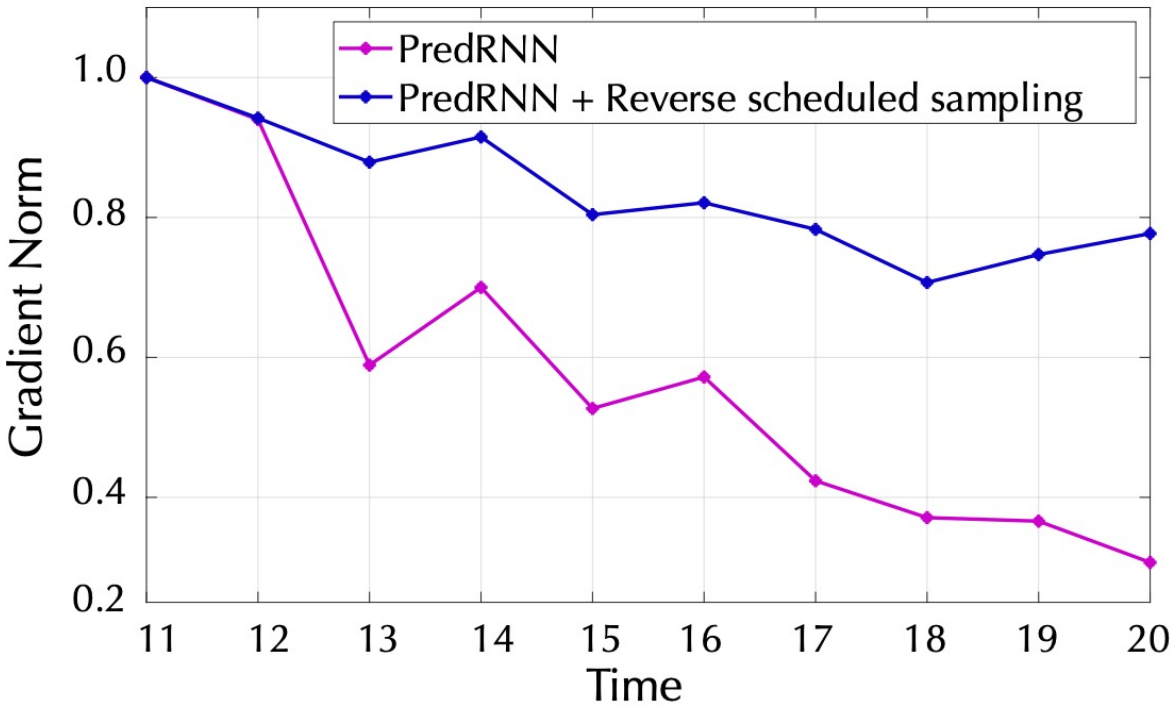
✔ Decoder inputs: ground truth x_t → last frame prediction $\hat{x}_t$



Reverse Scheduled Sampling (RSS)

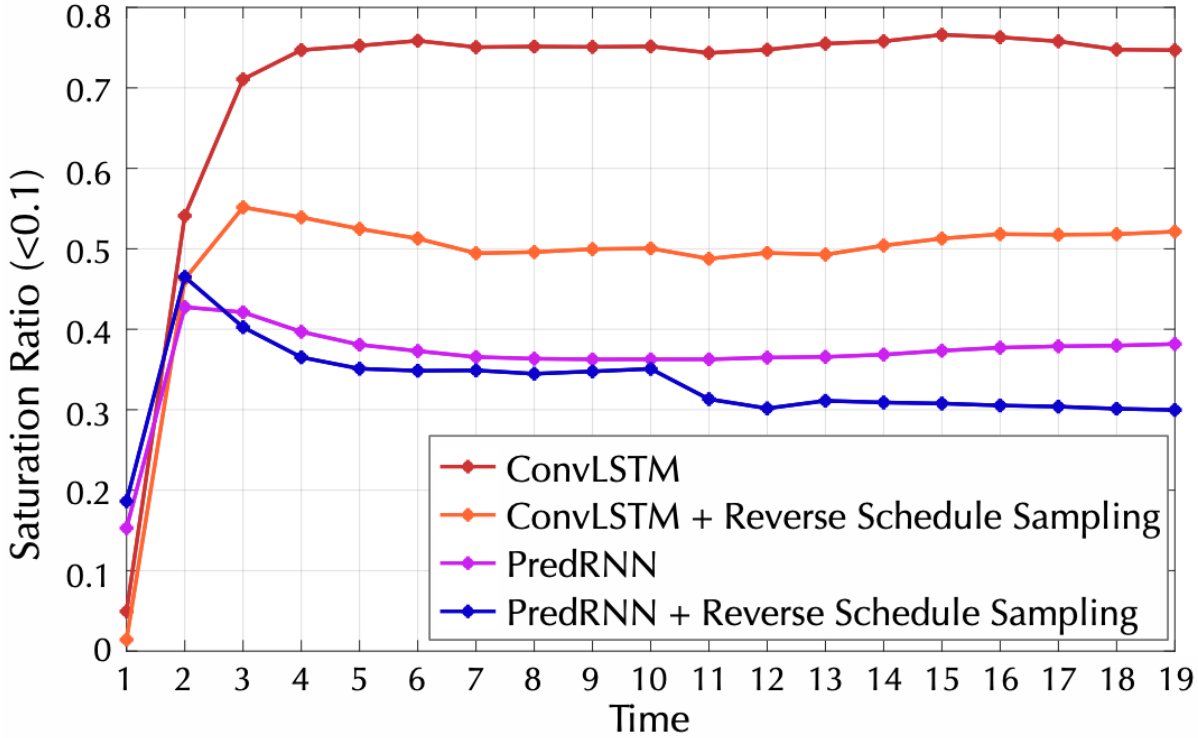
PredRNN-V2

Benefits of Reverse Scheduled Sampling



$$\frac{1}{T-1}\sum_{\tau=2}^{T}\left\|\nabla_{\mathcal{H}_{\tau}^{1}}\mathcal{L}_{t}\right\|,\ t\geq T+1$$

(1) Optimization: Alleviates the gradient optimization problem in RNNs



Forget gate value of C_t

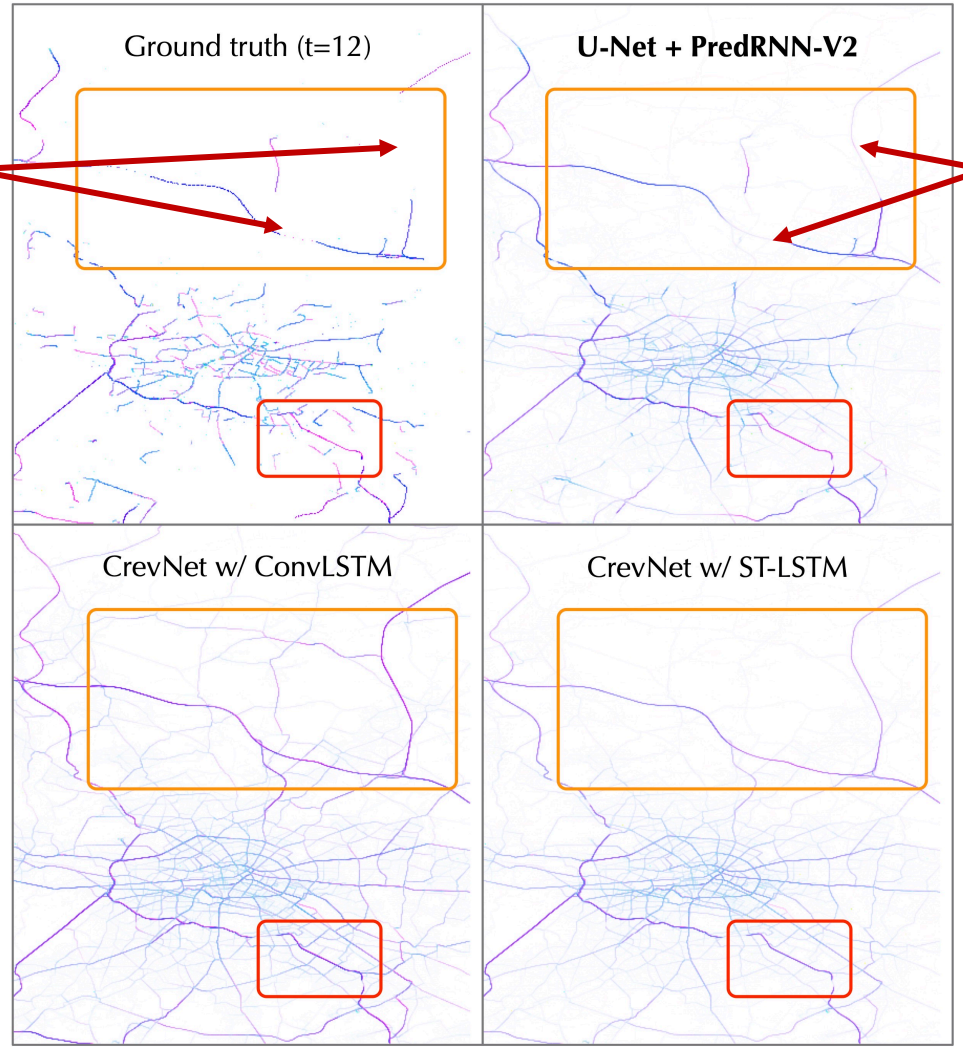
(2) Long-term dependencies: RSS creates a harder task, which can force the model to memorize more information

PredRNN-V2 for Traffic

- ✓ ST-LSTM can be used as a general unit and combined with U-Net
- ✓ Memory decoupling and RSS are general techniques to improve performance

PredRNN-V2

Backbone	Recurrent unit	Decoupl.	RSS	MSE (10^{-3})
U-Net [88]	None	×	×	6.992
U-Net + PredRNN	ST-LSTM	×	×	7.035
	ST-LSTM	○	○	**5.135** ✓
CrevNet [68]	ConvLSTM	×	×	6.789
	ST-LSTM	×	×	6.613
	ST-LSTM	○	○	6.506

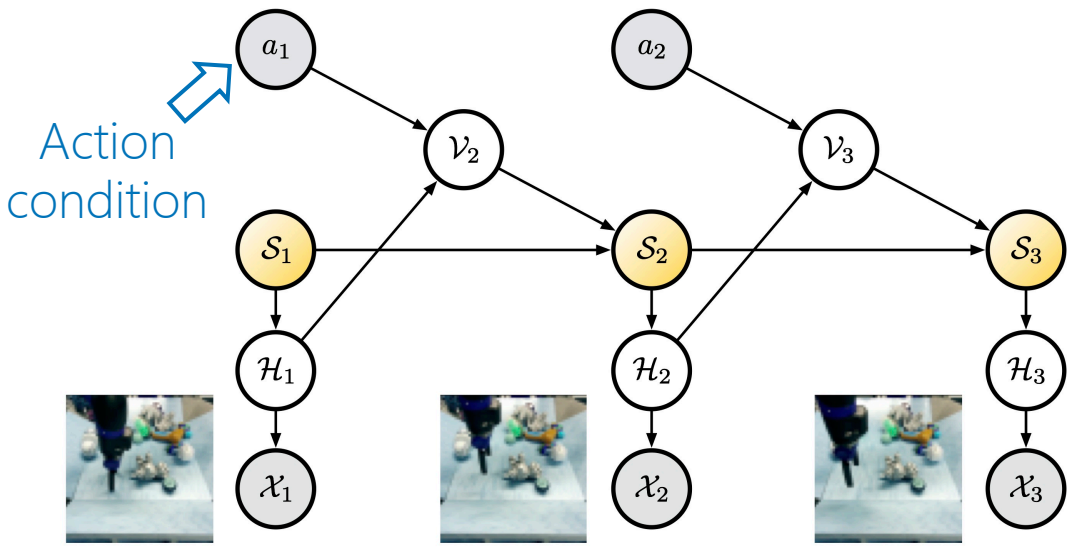


Blank area

Accurate prediction

CrevNet [Yu et al. ICLR20]

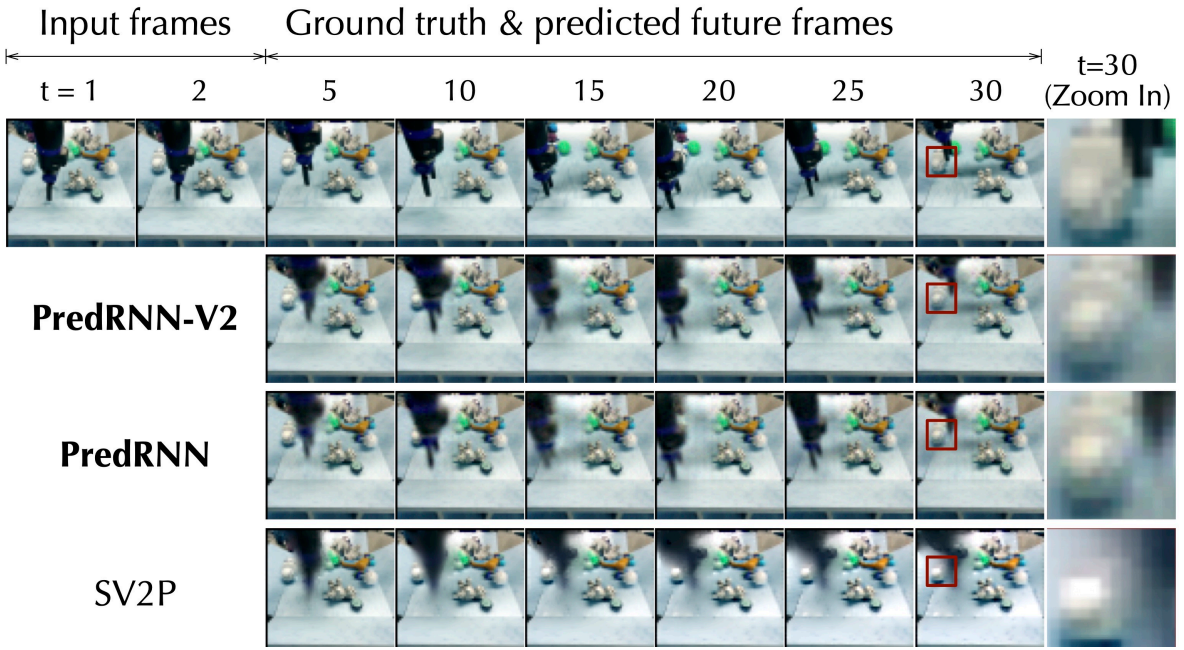
PredRNN-V2 for Robotics

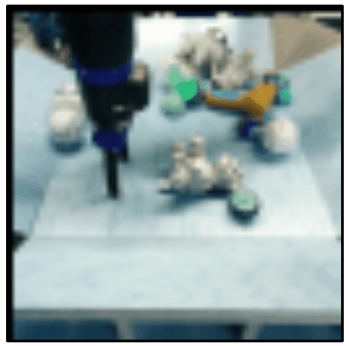


Action fusion: $\mathcal{V}_t^l = (W_{hv} * \mathcal{H}_{t-1}^l) \odot (W_{av} * \mathcal{A}_{t-1})$

$\mathcal{H}_t^l, \mathcal{C}_t^l, \mathcal{M}_t^l = \text{ST-LSTM}(\mathcal{X}_t, \mathcal{V}_t^l, \mathcal{C}_{t-1}^l, \mathcal{M}_t^{l-1}),$

- ✓ Action condition prediction
- ✓ Prediction model can be the world simulator for controlling



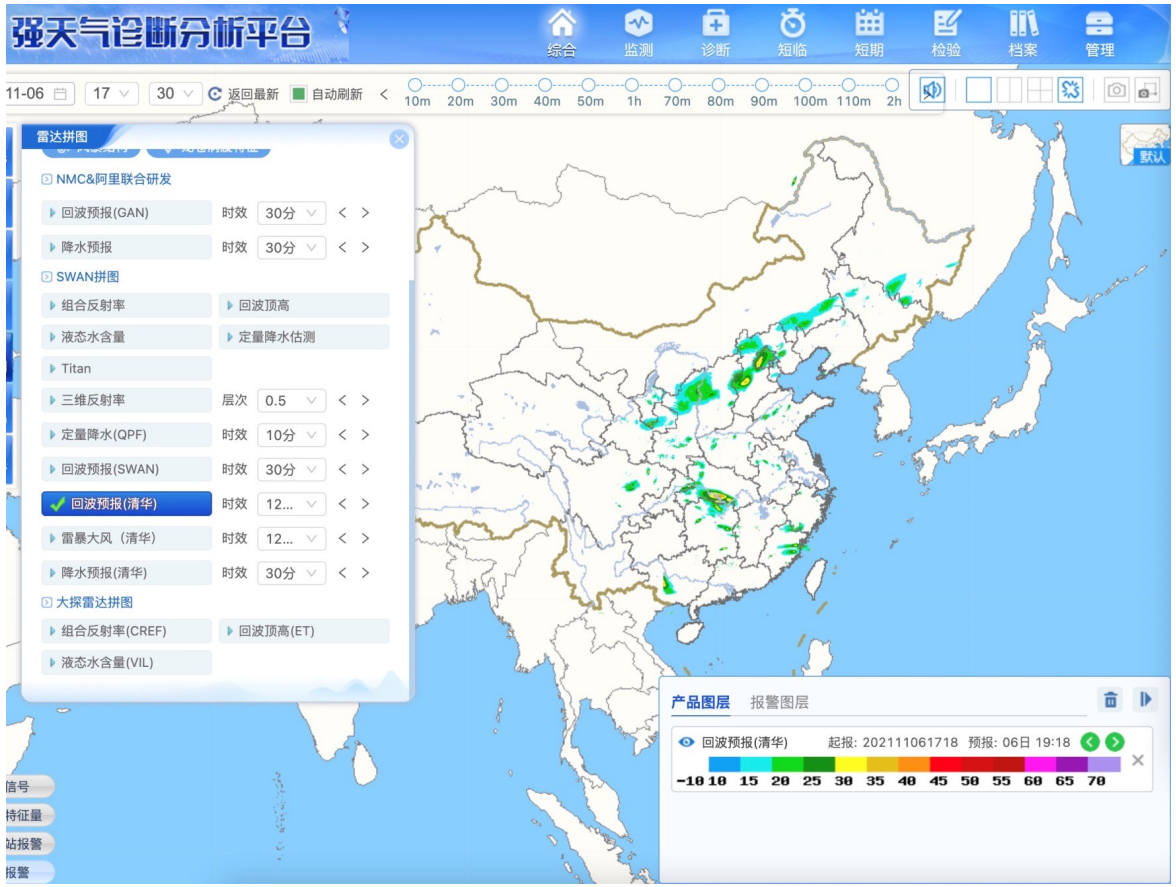

Ground truth

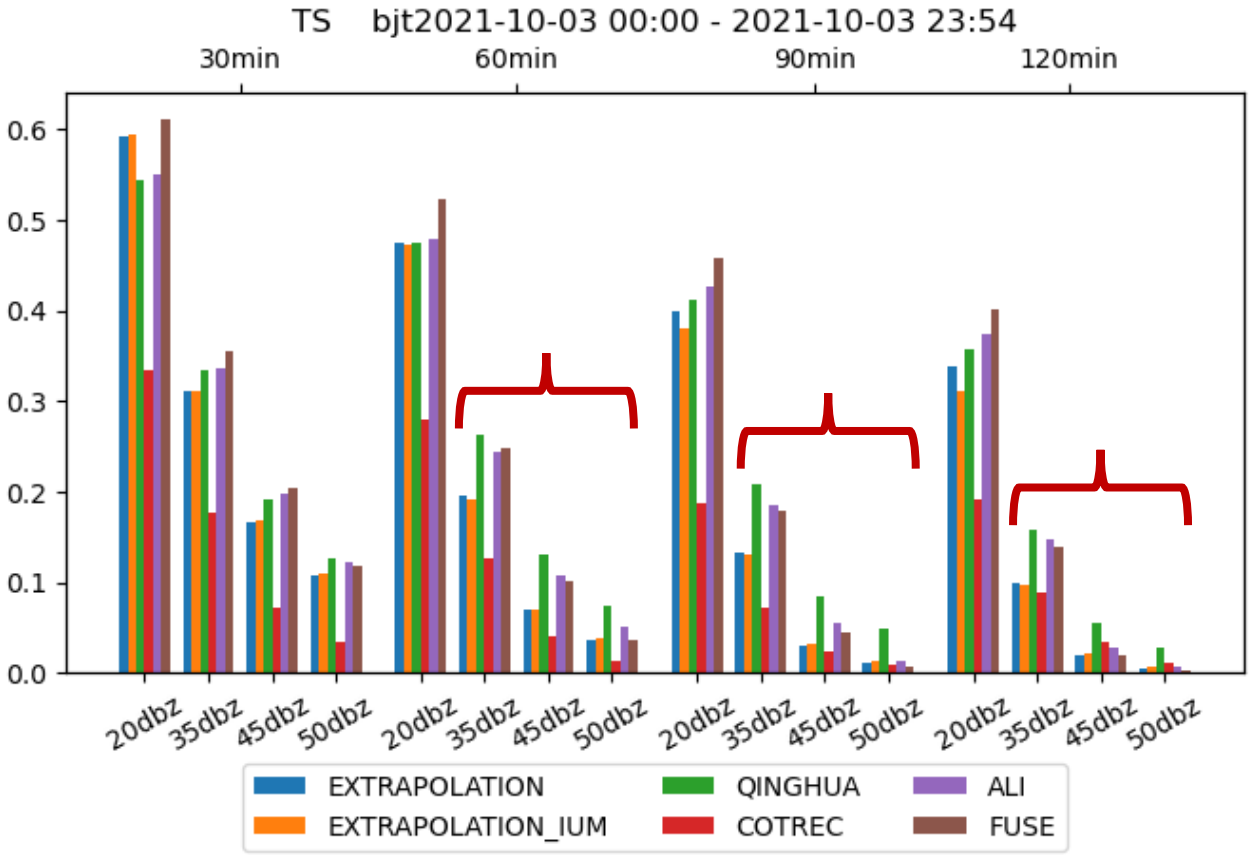

PredRNN-V2


SV2P

SV2P [Babaeizadeh et al. ICLR17]

PredRNN-V2 for Precipitation Nowcasting





PredRNN-V2 surpasses all other five methods and achieves the best performance in

long-term (≥60min) and high density (≥ 35dbz) radar prediction

Time Series Forecasting

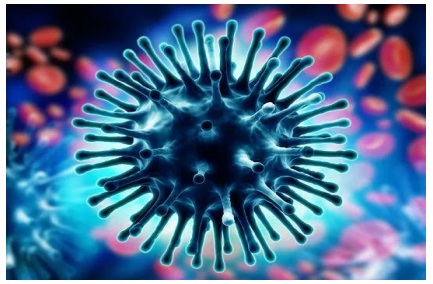
Wide Applications

Energy Consumption

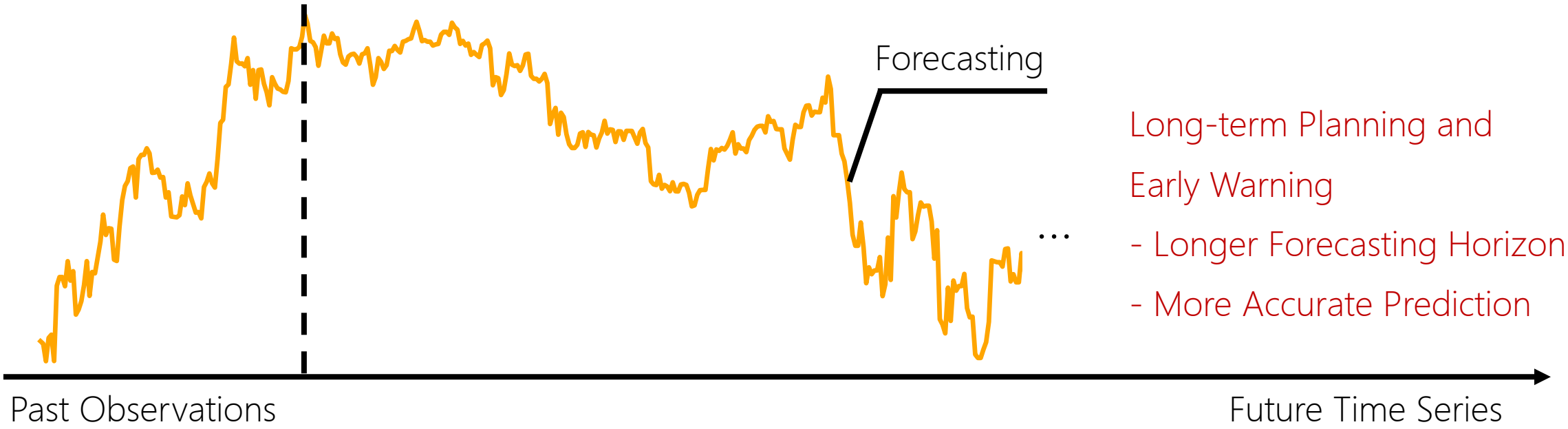
Traffic Flow

Economic Changes

Weather Variations



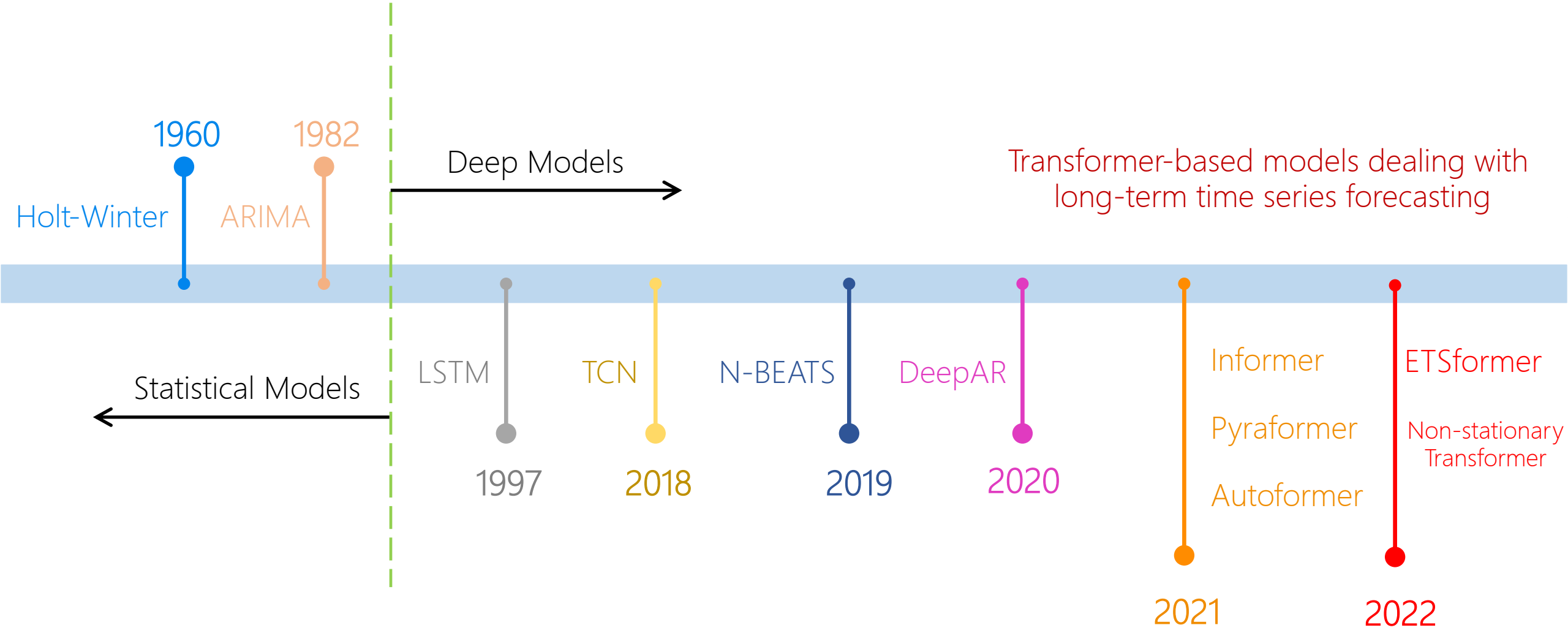
Disease Propagation



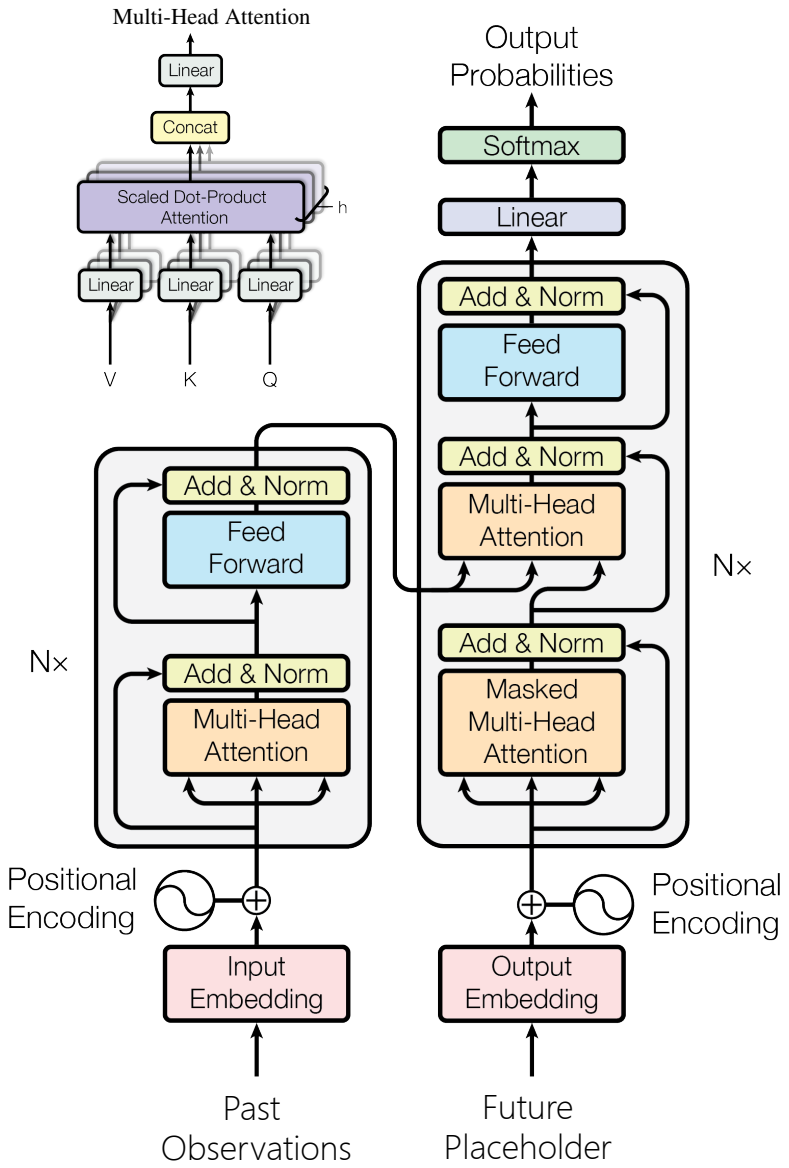
Long-term Planning and Early Warning

- Longer Forecasting Horizon
- More Accurate Prediction

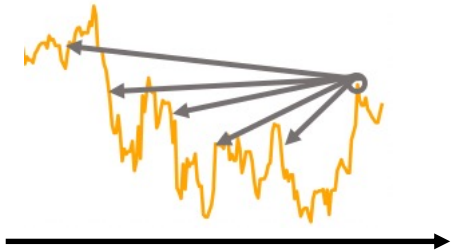
Timeline


1960
1982
Holt-Winter
ARIMA
Deep Models
Transformer-based models dealing with long-term time series forecasting
Statistical Models
LSTM
TCN
N-BEATS
DeepAR
Informer
ETSformer
Pyraformer
Non-stationary Transformer
Autoformer
1997
2018
2019
2020
2021
2022

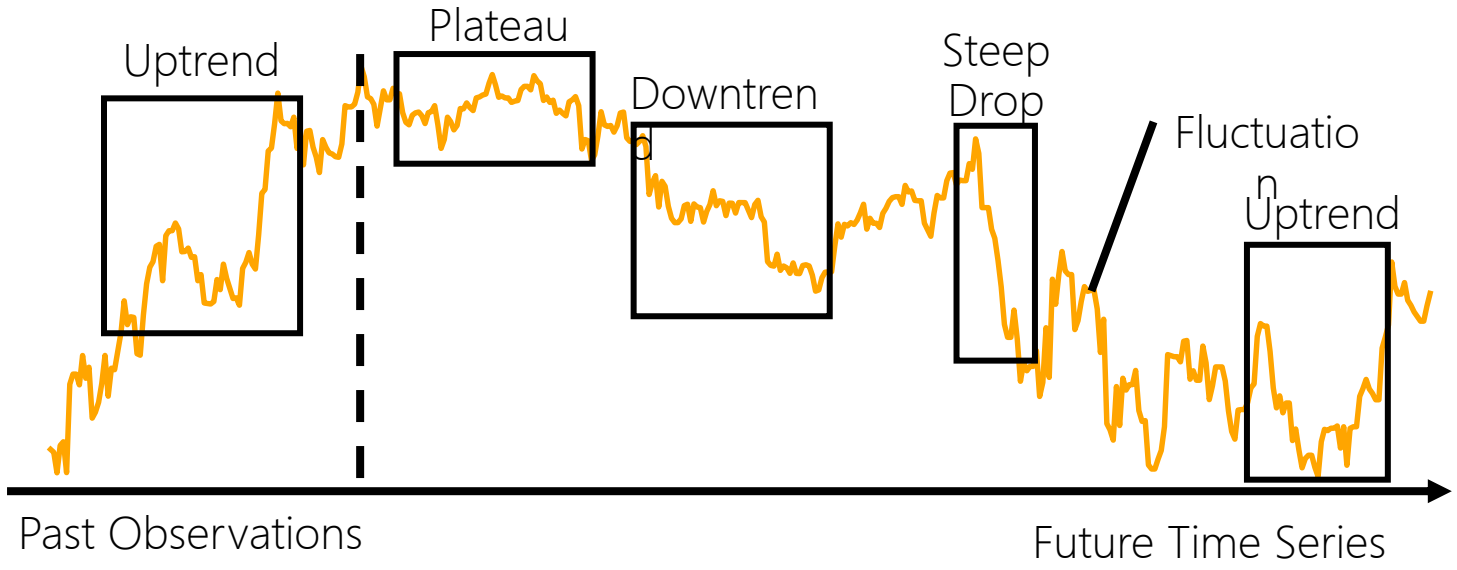
Transformer



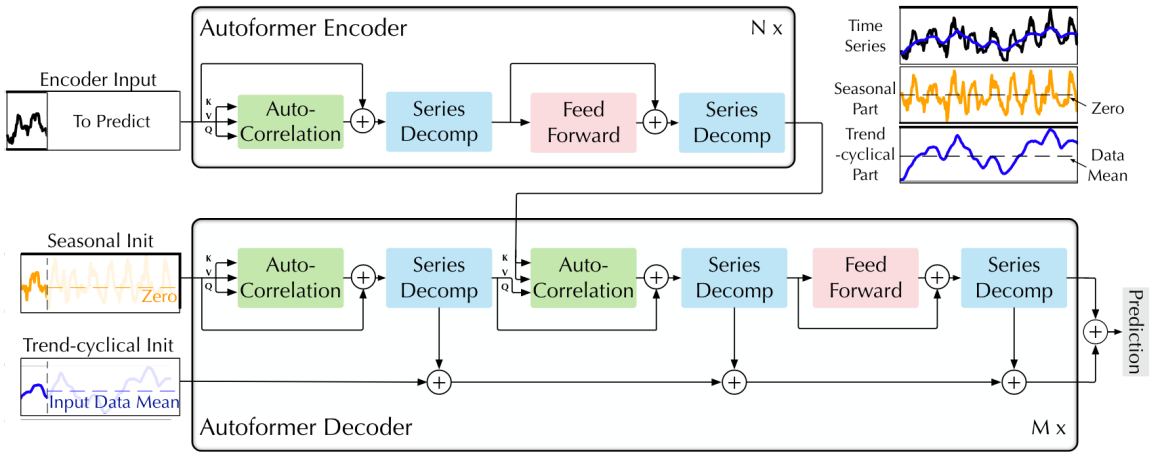
Modeling temporal dependencies globally with point-wise Self-Attention



Longer time series come with high complexity for Self-Attention and complex temporal patterns to discover



Timeline



- Decomposition architecture to split more predictable components
- Auto-Correlation to discover the temporal dependencies among sub-series with $\mathbf{O(L \log L)}$ complexity

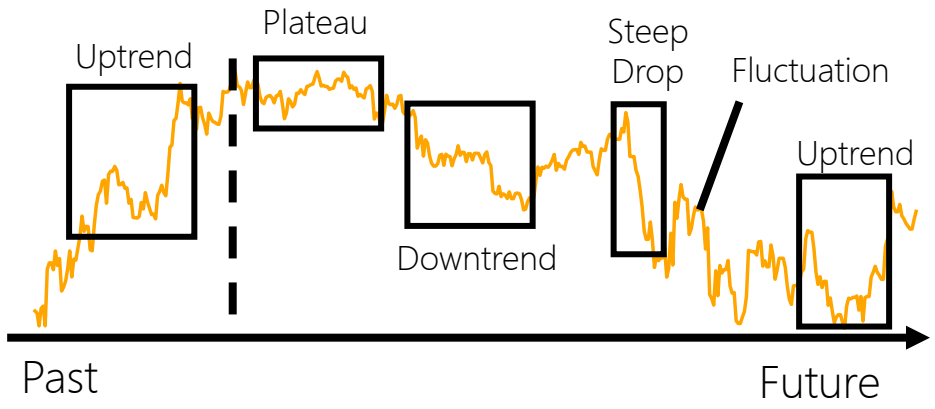
Autoformer

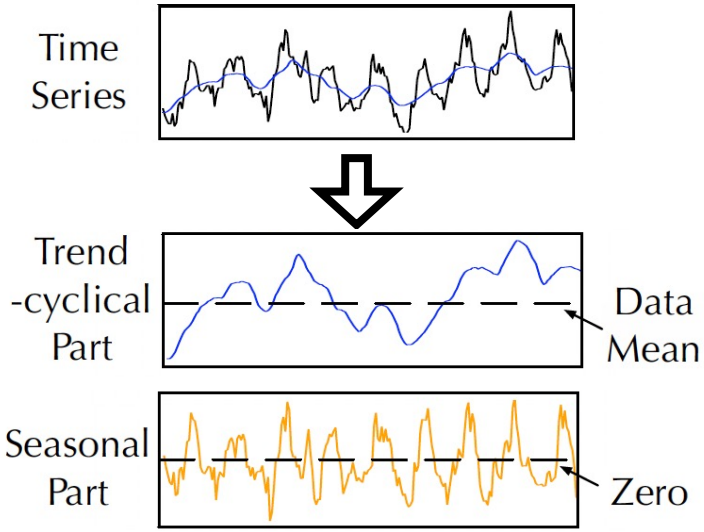
2021

Haixu Wu, Jiehui Xu, Jianmin Wang, Mingsheng Long✉. Autoformer: Decomposition Transformers with Auto-Correlation for Long-Term Series Forecasting. **NeurIPS**, 2021.

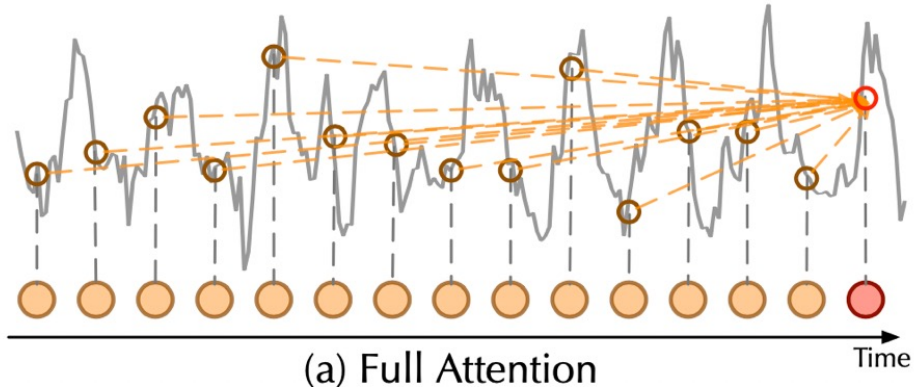
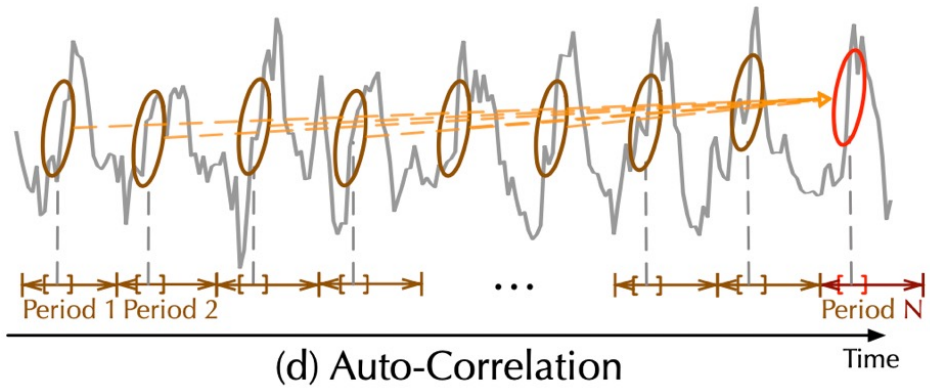
Comparison with Transformer

	Transformer	Autoformer
Intricate Temporal Patterns	Hard to directly find reliable temporal dependencies from raw series ✗	Decomposition architecture to ravel out the entangled temporal patterns ✓



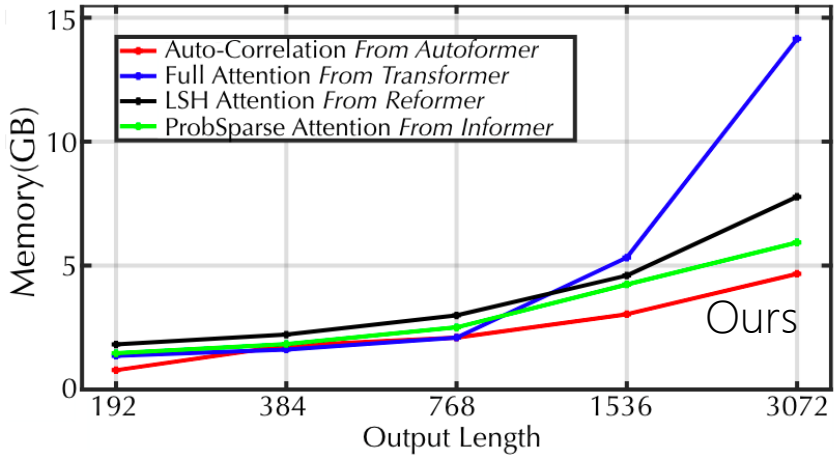


Comparison with Transformer

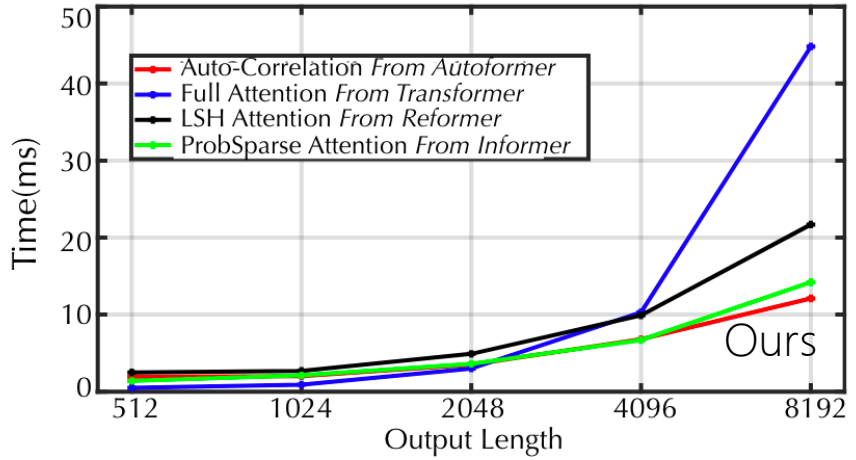
	Transformer	Autoformer
Time Series Continuity	Self-Attention discover the temporal dependencies from scattered points ✗	Auto-Correlation to discover the Series-wise temporal dependencies ✓
		

Comparison with Transformer

	Transformer	Autoformer
Computation Efficiency	• Point-wise Self-Attention is $O(L^2)$ • Adopt sparse version for efficiency resulting in the trade-off dilemma ✗	Auto-Correlation mechanism based on stochastic process theory with inherent $O(L \log L)$ complexity ✓

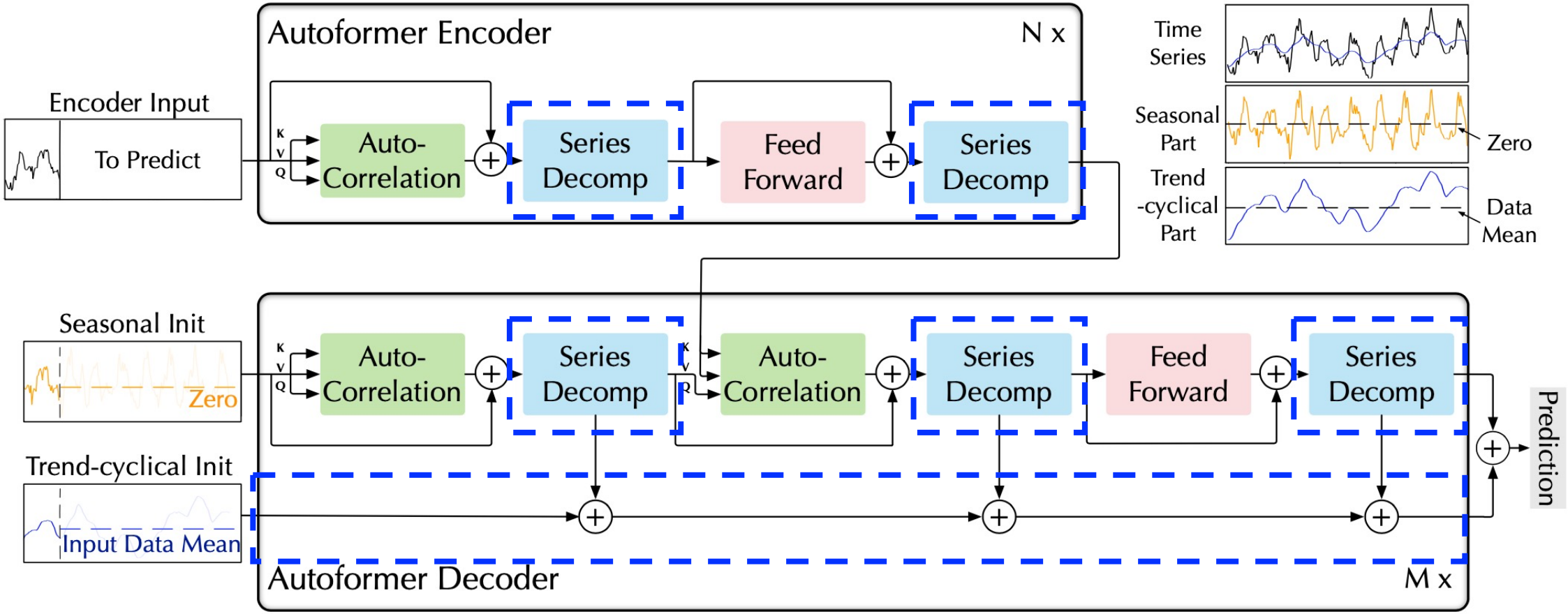


(a) Memory Efficiency Analysis



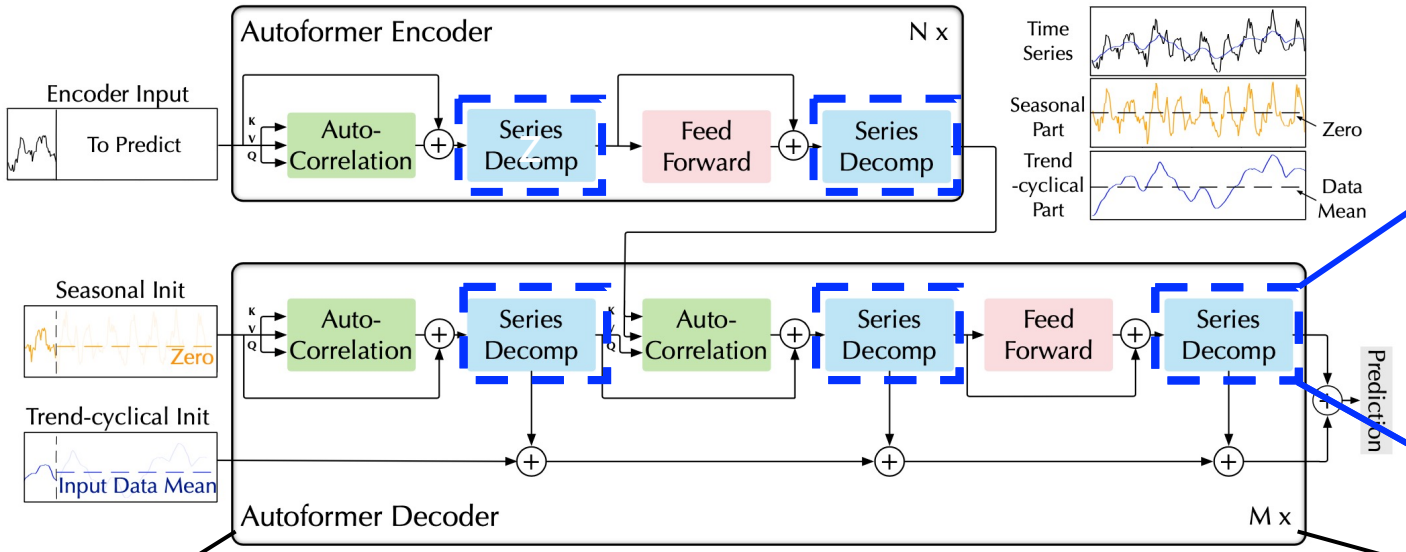
(b) Running Time Efficiency Analysis

Autoformer



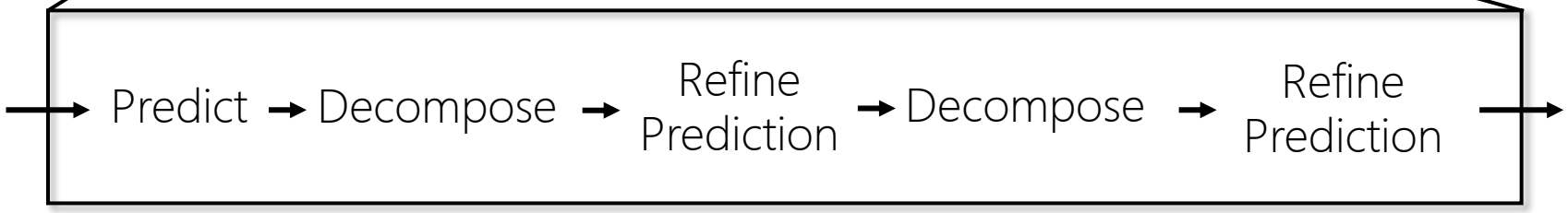
Decomposition architecture for intricate temporal patterns

Autoformer



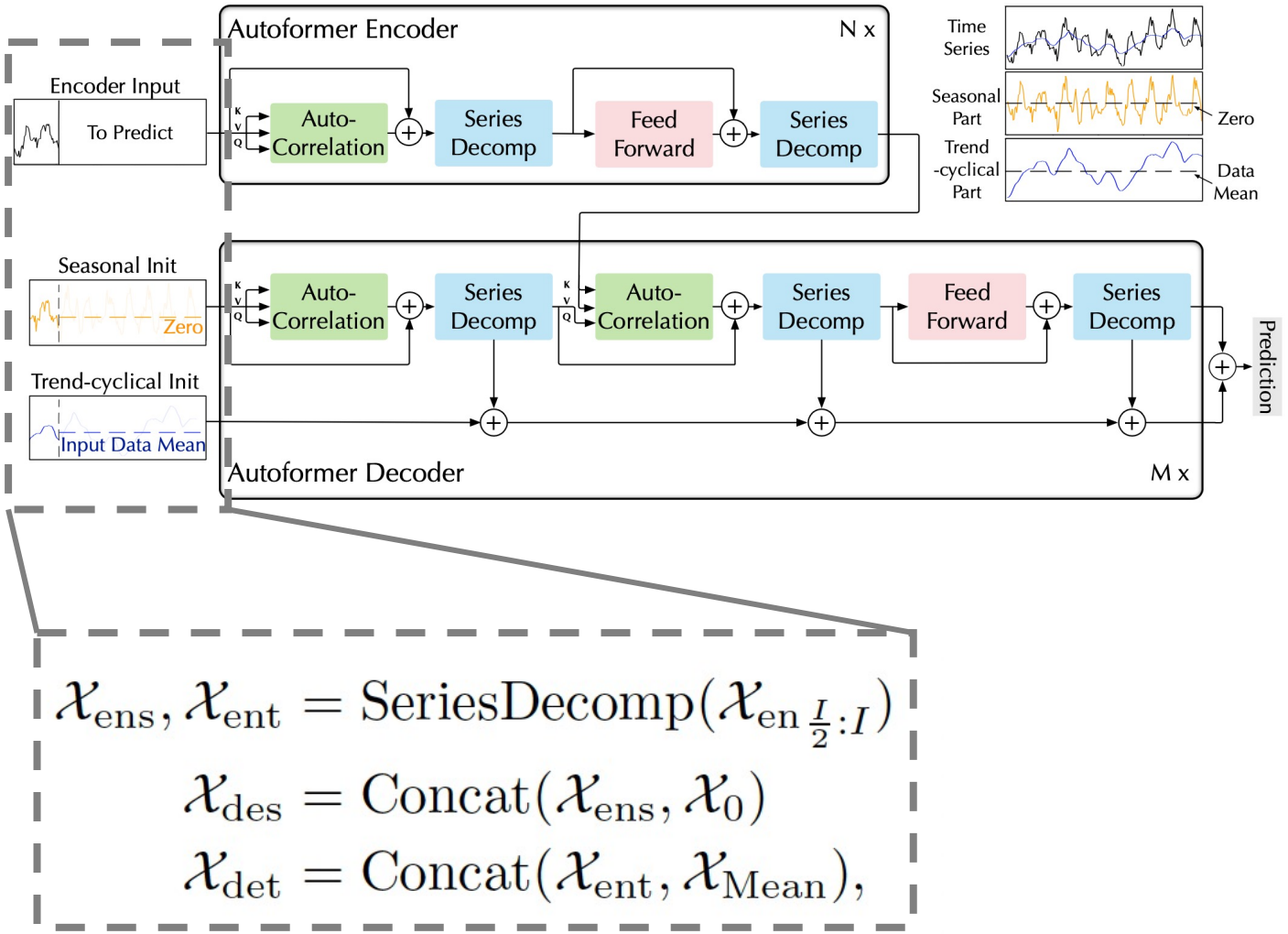
Decomposition with Moving Average

$$\mathcal{X}_{\mathrm{t}} = \mathrm{AvgPool}(\mathrm{Padding}(\mathcal{X}))$$
$$\mathcal{X}_{\mathrm{s}} = \mathcal{X} - \mathcal{X}_{\mathrm{t}},$$



Progressive decomposition capacity

Autoformer

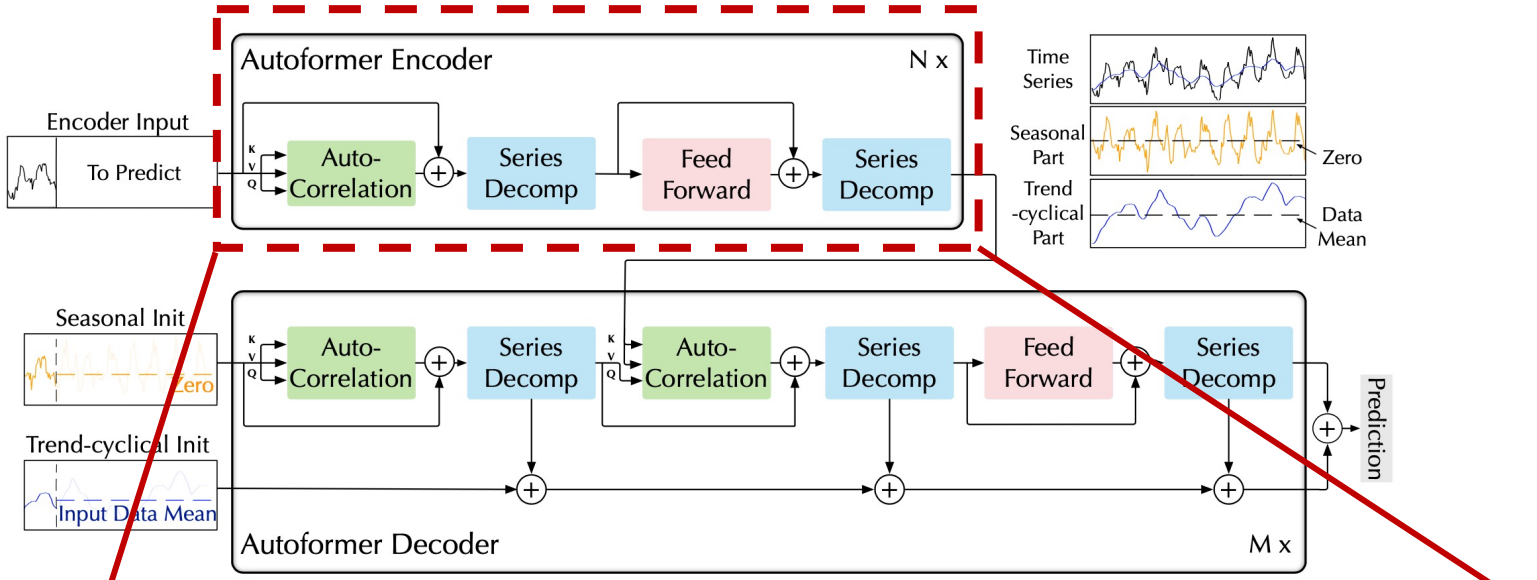


$$\mathcal{X}_{\text{ens}}, \mathcal{X}_{\text{ent}} = \text{SeriesDecomp}(\mathcal{X}_{\text{en}\,\frac{I}{2}:I})$$

$$\mathcal{X}_{\text{des}} = \text{Concat}(\mathcal{X}_{\text{ens}}, \mathcal{X}_0)$$

$$\mathcal{X}_{\text{det}} = \text{Concat}(\mathcal{X}_{\text{ent}}, \mathcal{X}_{\text{Mean}}),$$

Autoformer



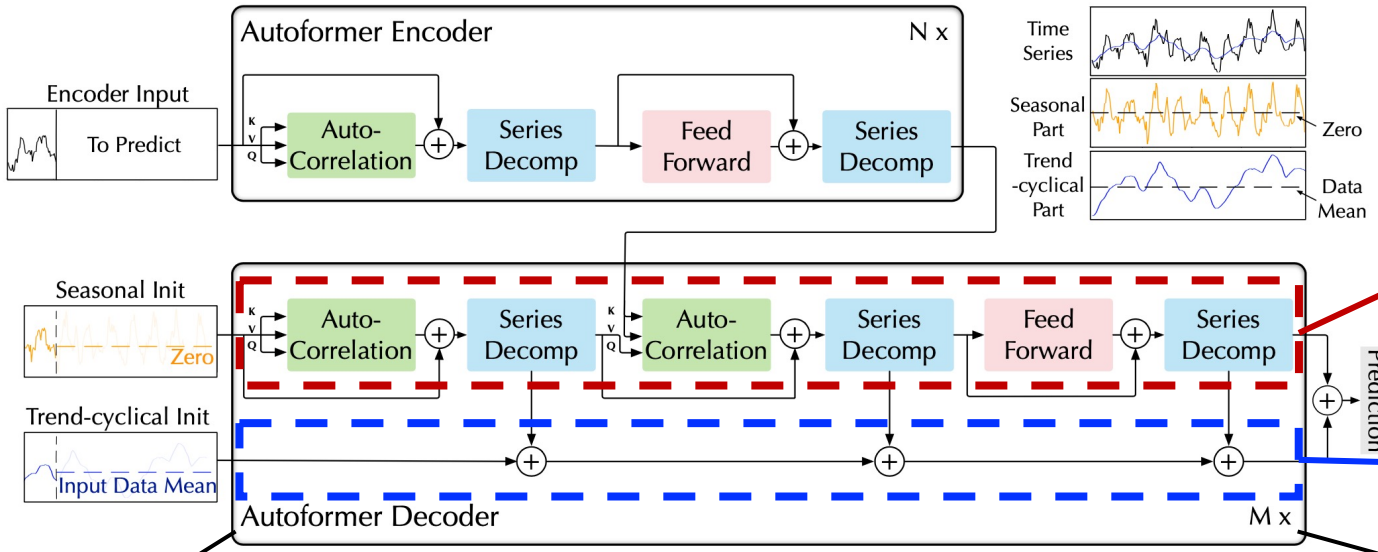
$$\mathcal{S}_{\text{en}}^{l,1}, \_ = \text{SeriesDecomp}\left(\text{AutoCorrelation}(\mathcal{X}_{\text{en}}^{l-1}) + \mathcal{X}_{\text{en}}^{l-1}\right)$$

$$\mathcal{S}_{\text{en}}^{l,2}, \_ = \text{SeriesDecomp}\left(\text{FeedForward}(\mathcal{S}_{\text{en}}^{l,1}) + \mathcal{S}_{\text{en}}^{l,1}\right),$$

Focus on seasonal part modeling,

Provide cross information for decoder

Autoformer



Adopt the Auto-Correlation for the seasonal prediction

Accumulation for the trend-cyclical prediction

$$\mathcal{S}_{\text{de}}^{l,1}, \mathcal{T}_{\text{de}}^{l,1} = \text{SeriesDecomp}\left(\text{AutoCorrelation}(\mathcal{X}_{\text{de}}^{l-1}) + \mathcal{X}_{\text{de}}^{l-1}\right)$$

$$\mathcal{S}_{\text{de}}^{l,2}, \mathcal{T}_{\text{de}}^{l,2} = \text{SeriesDecomp}\left(\text{AutoCorrelation}(\mathcal{S}_{\text{de}}^{l,1}, \mathcal{X}_{\text{en}}^{N}) + \mathcal{S}_{\text{de}}^{l,1}\right)$$

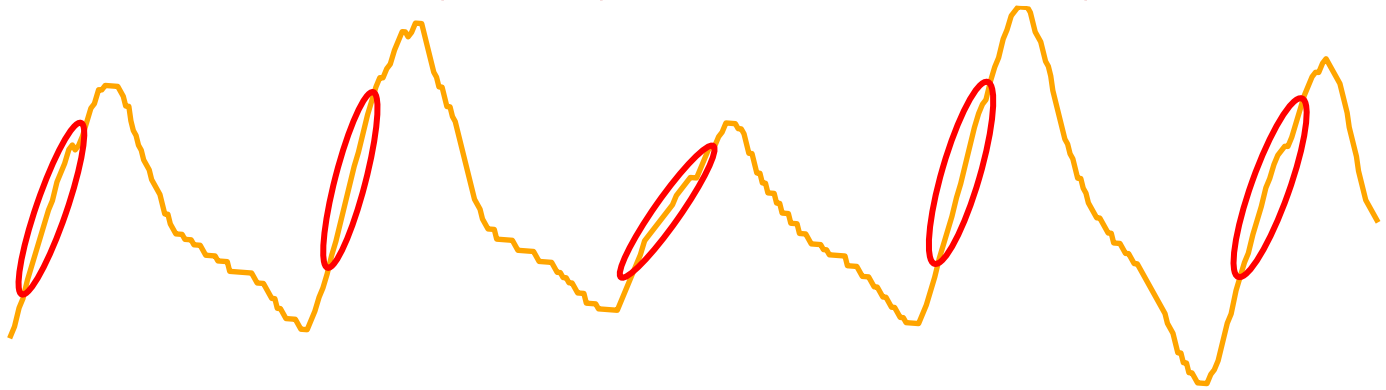
$$\mathcal{S}_{\text{de}}^{l,3}, \mathcal{T}_{\text{de}}^{l,3} = \text{SeriesDecomp}\left(\text{FeedForward}(\mathcal{S}_{\text{de}}^{l,2}) + \mathcal{S}_{\text{de}}^{l,2}\right)$$

$$\mathcal{T}_{\text{de}}^{l} = \mathcal{T}_{\text{de}}^{l-1} + \mathcal{W}_{l,1} * \mathcal{T}_{\text{de}}^{l,1} + \mathcal{W}_{l,2} * \mathcal{T}_{\text{de}}^{l,2} + \mathcal{W}_{l,3} * \mathcal{T}_{\text{de}}^{l,3},$$

Autoformer

Period-based dependencies

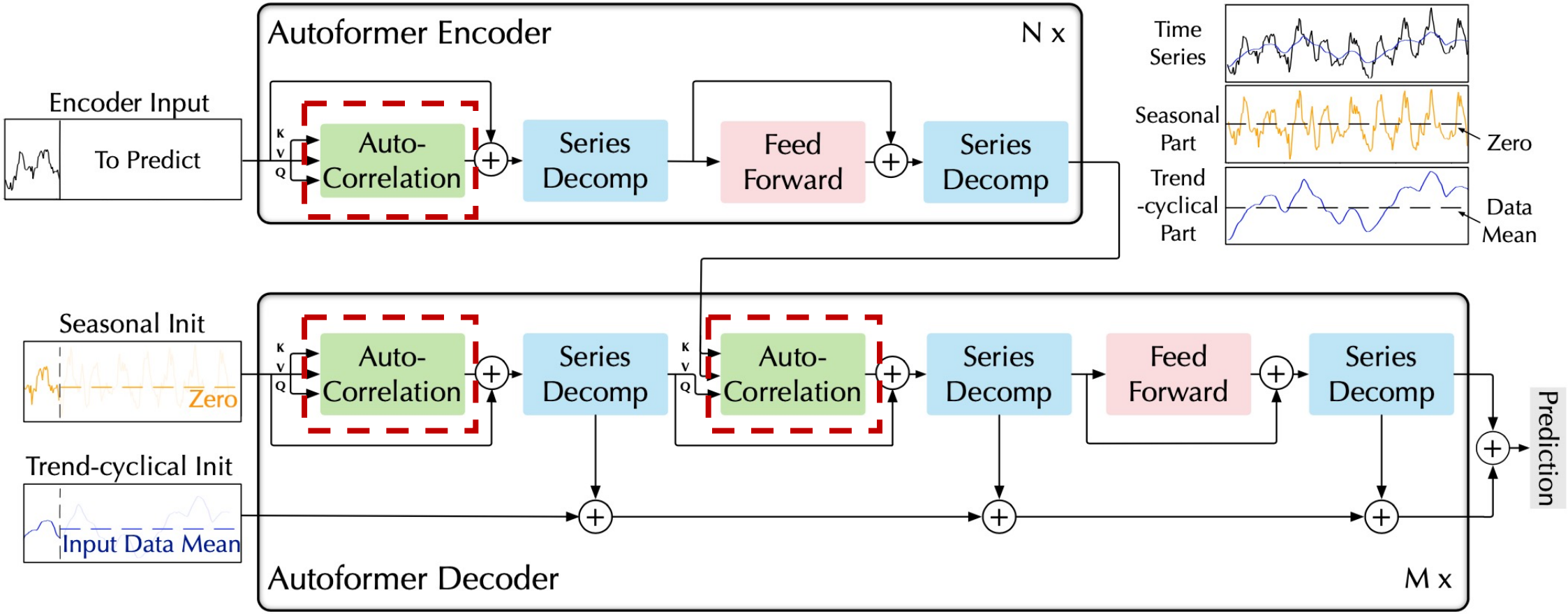
The same phase position of different periods



Benefit from the deep decomposition,
the seasonal part is highlighted with periodicity

Conduct the dependencies discovery and representation aggregation at the series level

Autoformer



Series-wise Auto-Correlation towards information utilization bottleneck

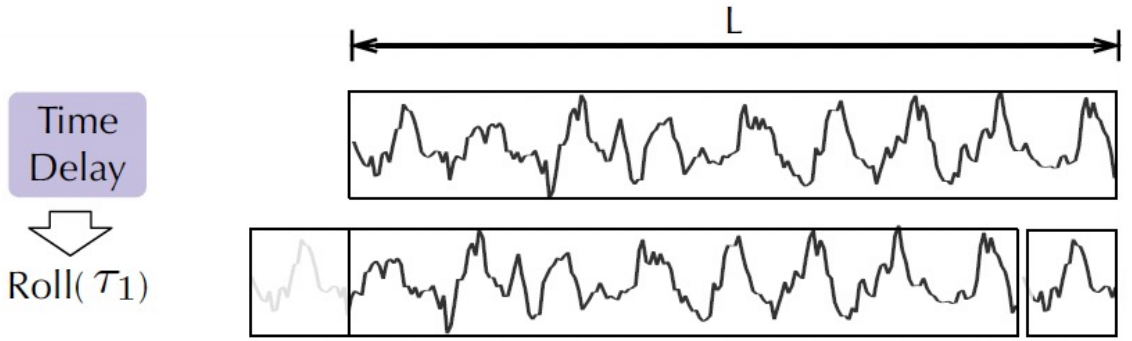
Autoformer

Discover period-based dependencies with autocorrelation in stochastic process:

$$\mathcal{R}_{\mathcal{X}\mathcal{X}}(\tau) = \lim_{L\to\infty} \frac{1}{L} \sum_{t=0}^{L-1} \mathcal{X}_t \mathcal{X}_{t-\tau}.$$

Autocorrelation reflects the time delay similarity,

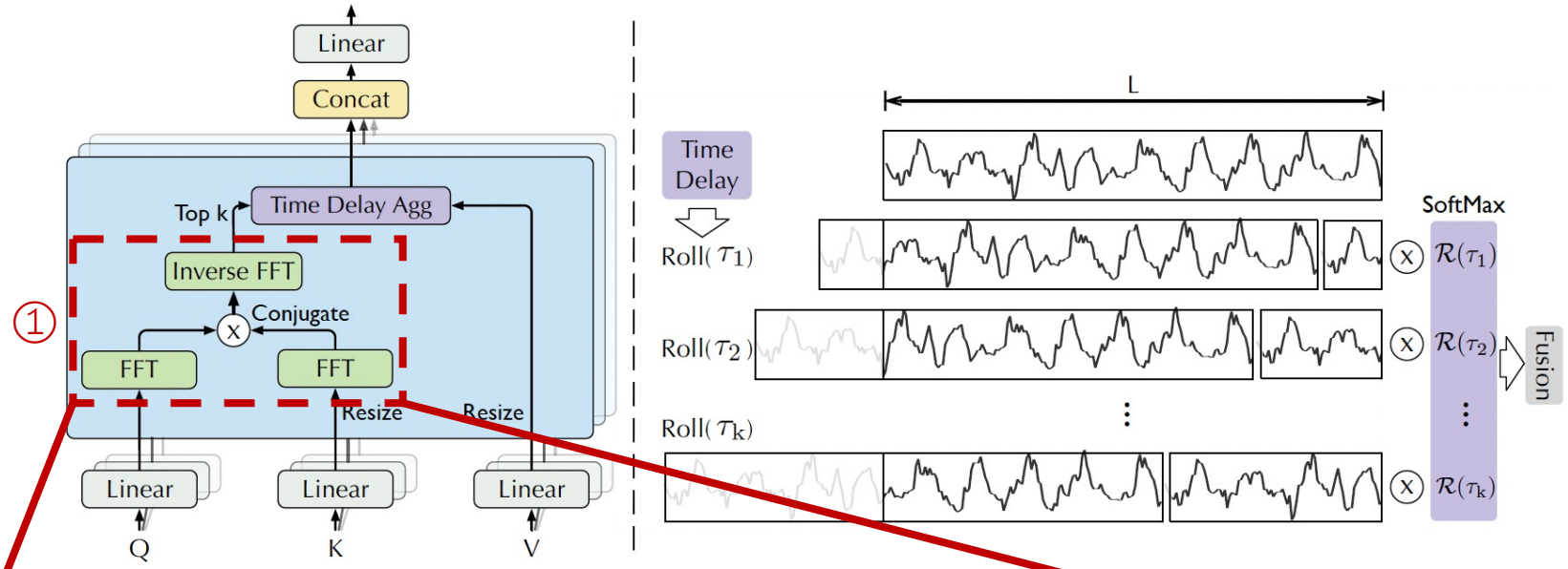
and corresponds to the confidence of period estimation



Larger autocorrelation $\mathcal{R}(\tau)$ means

- stronger time delay similarity w.r.t. τ
- more confidence of period length as τ

Autoformer



$$\mathcal{S}_{\mathcal{X}\mathcal{X}}(f) = \mathcal{F}(\mathcal{X}_t)\mathcal{F}^*(\mathcal{X}_t) = \int_{-\infty}^{\infty} \mathcal{X}_t e^{-i2\pi tf}\mathrm{d}t \overline{\int_{-\infty}^{\infty} \mathcal{X}_t e^{-i2\pi tf}\mathrm{d}t}$$

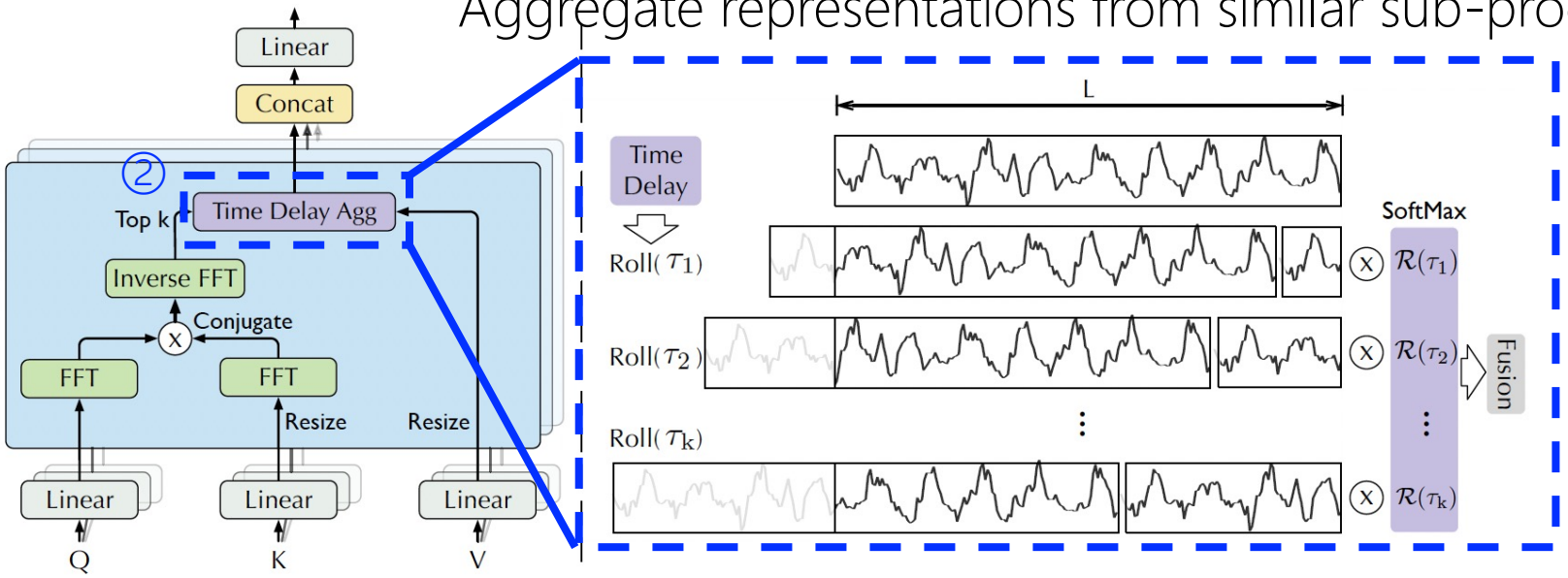
$$\mathcal{R}_{\mathcal{X}\mathcal{X}}(\tau) = \mathcal{F}^{-1}(\mathcal{S}_{\mathcal{X}\mathcal{X}}(f)) = \int_{-\infty}^{\infty} \mathcal{S}_{\mathcal{X}\mathcal{X}}(f) e^{i2\pi f\tau}\mathrm{d}f,$$

① Discover period-based dependencies with $\mathbf{O(L \log L)}$ complexity

Efficient computation of autocorrelation with Wiener–Khinchin theorem by FFT

Autoformer

Aggregate representations from similar sub-processes



② Aggregate similar sub-processes from different periods

1. Select the Top-k period lengths

2. Softmax-normalization

3. Align delayed series and aggregate sub-series representations

$$\tau_1, \cdots, \tau_k = \underset{\tau \in \{1, \cdots, L\}}{\arg \operatorname{Topk}} \left(\mathcal{R}_{\mathcal{Q},\mathcal{K}}(\tau) \right)$$

$$\widehat{\mathcal{R}}_{\mathcal{Q},\mathcal{K}}(\tau_1), \cdots, \widehat{\mathcal{R}}_{\mathcal{Q},\mathcal{K}}(\tau_k) = \operatorname{SoftMax}\left(\mathcal{R}_{\mathcal{Q},\mathcal{K}}(\tau_1), \cdots, \mathcal{R}_{\mathcal{Q},\mathcal{K}}(\tau_k) \right)$$

$$\operatorname{AutoCorrelation}(\mathcal{Q}, \mathcal{K}, \mathcal{V}) = \sum_{i=1}^{k} \operatorname{Roll}(\mathcal{V}, \tau_k) \widehat{\mathcal{R}}_{\mathcal{Q},\mathcal{K}}(\tau_k)$$

Autoformer

Benchmark

				Transformers						LSTMs				TCN		Prediction Accuracy Relative Promotion (In MSE)
Models		**Autoformer**		Informer[41]		LogTrans[20]		Reformer[17]		LSTNet[19]		LSTM[13]		TCN[3]		
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	
Energy	ETT*	96	**0.255**	**0.339**	0.365	0.453	0.768	0.642	0.658	0.619	3.142	1.365	2.041	1.073	3.041	1.330	Input-96-predict-336 ↑ 74%
		192	**0.281**	**0.340**	0.533	0.563	0.989	0.757	1.078	0.827	3.154	1.369	2.249	1.112	3.072	1.339	
		336	**0.339**	**0.372**	1.363	0.887	1.334	0.872	1.549	0.972	3.160	1.369	2.568	1.238	3.105	1.348	
		720	**0.422**	**0.419**	3.379	1.388	3.048	1.328	2.631	1.242	3.171	1.368	2.720	1.287	3.135	1.354	
	Electricity	96	**0.201**	**0.317**	0.274	0.368	0.258	0.357	0.312	0.402	0.680	0.645	0.375	0.437	0.985	0.813	Input-96-predict-336 ↑ 18%
		192	**0.222**	**0.334**	0.296	0.386	0.266	0.368	0.348	0.433	0.725	0.676	0.442	0.473	0.996	0.821	
		336	**0.231**	**0.338**	0.300	0.394	0.280	0.380	0.350	0.433	0.828	0.727	0.439	0.473	1.000	0.824	
		720	**0.254**	**0.361**	0.373	0.439	0.283	0.376	0.340	0.420	0.957	0.811	0.980	0.814	1.438	0.784	
Economics	Exchange	96	**0.197**	**0.323**	0.847	0.752	0.968	0.812	1.065	0.829	1.551	1.058	1.453	1.049	3.004	1.432	Input-96-predict-336 ↑ 61%
		192	**0.300**	**0.369**	1.204	0.895	1.040	0.851	1.188	0.906	1.477	1.028	1.846	1.179	3.048	1.444	
		336	**0.509**	**0.524**	1.672	1.036	1.659	1.081	1.357	0.976	1.507	1.031	2.136	1.231	3.113	1.459	
		720	**1.447**	**0.941**	2.478	1.310	1.941	1.127	1.510	1.016	2.285	1.243	2.984	1.427	3.150	1.458	
Traffic	Traffic	96	**0.613**	**0.388**	0.719	0.391	0.684	0.384	0.732	0.423	1.107	0.685	0.843	0.453	1.438	0.784	Input-96-predict-336 ↑ 15%
		192	**0.616**	**0.382**	0.696	0.379	0.685	0.390	0.733	0.420	1.157	0.706	0.847	0.453	1.463	0.794	
		336	**0.622**	**0.337**	0.777	0.420	0.733	0.408	0.742	0.420	1.216	0.730	0.853	0.455	1.479	0.799	
		720	**0.660**	**0.408**	0.864	0.472	0.717	0.396	0.755	0.423	1.481	0.805	1.500	0.805	1.499	0.804	
Weather	Weather	96	**0.266**	**0.336**	0.300	0.384	0.458	0.490	0.689	0.596	0.594	0.587	0.369	0.406	0.615	0.589	Input-96-predict-336 ↑ 21%
		192	**0.307**	**0.367**	0.598	0.544	0.658	0.589	0.752	0.638	0.560	0.565	0.416	0.435	0.629	0.600	
		336	**0.359**	**0.395**	0.578	0.523	0.797	0.652	0.639	0.596	0.597	0.587	0.455	0.454	0.639	0.608	
		720	**0.419**	**0.428**	1.059	0.741	0.869	0.675	1.130	0.792	0.618	0.599	0.535	0.520	0.639	0.610	
Disease	ILI	24	**3.483**	**1.287**	5.764	1.677	4.480	1.444	4.400	1.382	6.026	1.770	5.914	1.734	6.624	1.830	Input-24-predict-48 ↑ 43%
		36	**3.103**	**1.148**	4.755	1.467	4.799	1.467	4.783	1.448	5.340	1.668	6.631	1.845	6.858	1.879	
		48	**2.669**	**1.085**	4.763	1.469	4.800	1.468	4.832	1.465	6.080	1.787	6.736	1.857	6.968	1.892	
		60	**2.770**	**1.125**	5.264	1.564	5.278	1.560	4.882	1.483	5.548	1.720	6.870	1.879	7.127	1.918	

* *ETT* means the ETTm2. See supplementary materials for the **full benchmark** of ETTh1, ETTh2, ETTm1.

Autoformer

Ablation of Decomposition

Input-96	Transformer[35]			Informer[41]			LogTrans[17]			Reformer[20]			Promotion	
Predict-O	Origin	Sep	Ours	Origin	Sep	Ours	Origin	Sep	Ours	Origin	Sep	Ours	Sep	Ours
96	0.604	0.311	**0.204**	0.365	0.490	**0.354**	0.768	0.862	**0.231**	0.658	0.445	**0.218**	0.069	0.347
192	1.060	0.760	**0.266**	0.533	0.658	**0.432**	0.989	0.533	**0.378**	1.078	0.510	**0.336**	0.300	0.562
336	1.413	0.665	**0.375**	1.363	1.469	**0.481**	1.334	0.762	**0.362**	1.549	1.028	**0.366**	0.434	1.019
720	2.672	3.200	**0.537**	3.379	2.766	**0.822**	3.048	2.601	**0.539**	2.631	2.845	**0.502**	0.079	2.332

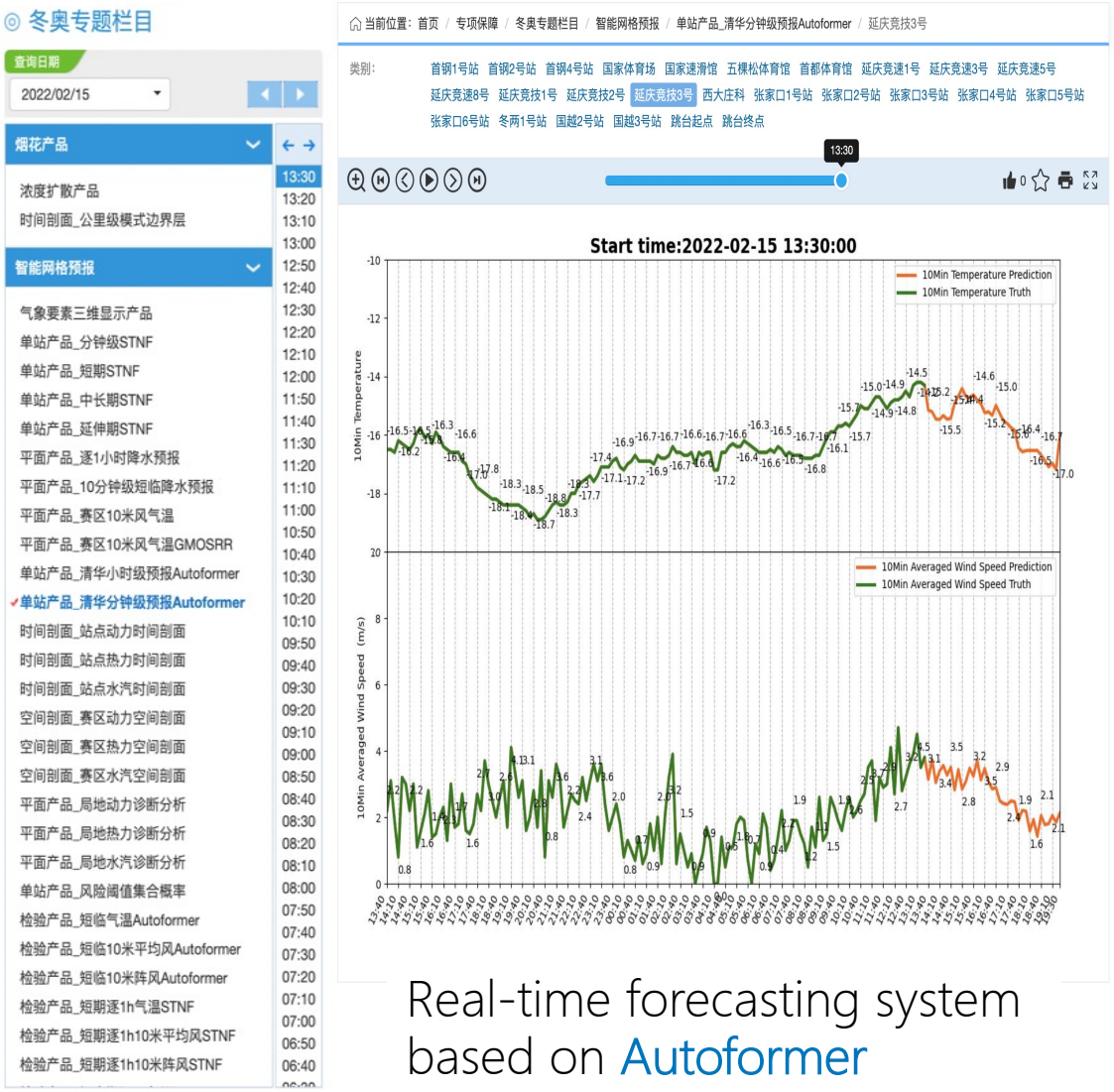
- Decomposition outperforms separately forecasting convention especially in the long-term setting
- Decomposition architecture can be generalized to other Transformers

Ablation of Auto-Correlation

Input Length I		96			192			336		
Prediction Length O		336	720	1440	336	720	1440	336	720	1440
Auto-Correlation	MSE	**0.339**	**0.422**	**0.555**	**0.355**	**0.429**	**0.503**	**0.361**	**0.425**	**0.574**
	MAE	**0.372**	**0.419**	**0.496**	**0.392**	**0.430**	**0.484**	**0.406**	**0.440**	**0.534**
Full Attention[35]	MSE	0.375	0.537	0.667	0.450	0.554	-	0.501	0.647	-
	MAE	0.425	0.502	0.589	0.470	0.533	-	0.485	0.491	-
LogSparse Attention[20]	MSE	0.362	0.539	0.582	0.420	0.552	0.958	0.474	0.601	-
	MAE	0.413	0.522	0.529	0.450	0.513	0.736	0.474	0.524	-
LSH Attention[17]	MSE	0.366	0.502	0.663	0.407	0.636	1.069	0.442	0.615	-
	MAE	0.404	0.475	0.567	0.421	0.571	0.756	0.476	0.532	-
ProbSparse Attention[41]	MSE	0.481	0.822	0.715	0.404	1.148	0.732	0.417	0.631	1.133
	MAE	0.472	0.559	0.586	0.425	0.654	0.602	0.434	0.528	0.691

- Under various input-predict settings, Auto-Correlation outperforms Self-Attention and its variants

Autoformer for 2022 Beijing Olympics



Real-time forecasting system based on Autoformer

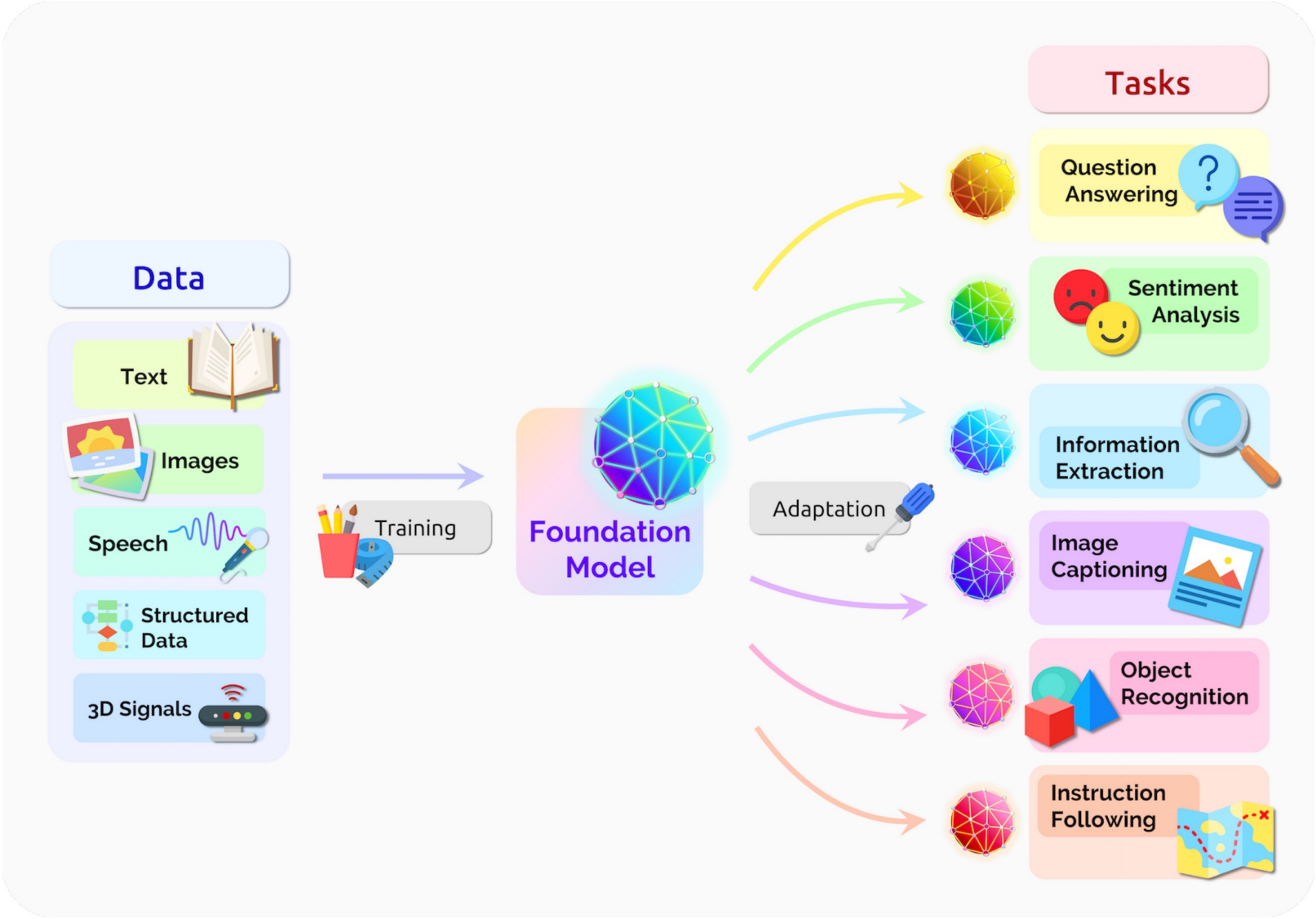
Indoors: Temperature

Outdoors: Wind speed

Provides online forecast service of temperature and wind speed for the 2022 Beijing Winter Olympics, assists athletes preparation and schedule planning, works as a solid support for the competition.

Achieves 10-minute real-time temperature and wind speed forecast based on meteorological observation, and achieves 23% lower forecast error than the mainstream numerical prediction methods.

Foundation Models



[Data Universal]

Learn from various modalities

[Task Universal]

Adapt to a wide range of downstream tasks

Bommasani et al. On the Opportunities and Risks of Foundation Models. Arxiv 2021.

General Relation Modeling of Transformers

Image

Relation among Image Patches

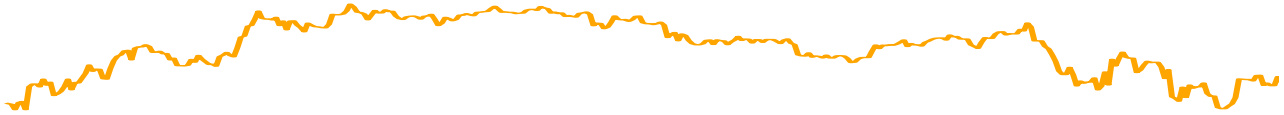
Language

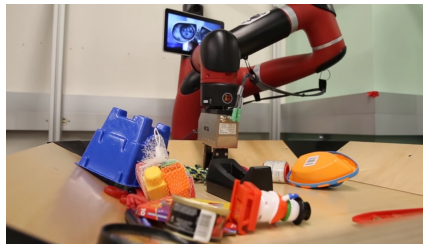
Relation among Words

[SOS] Flowformer is a task-universal linear Transformer. [EOS]

Time Series

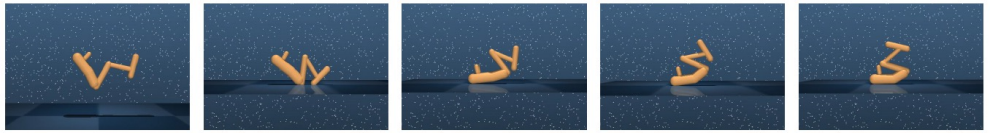
Relation among Time Points





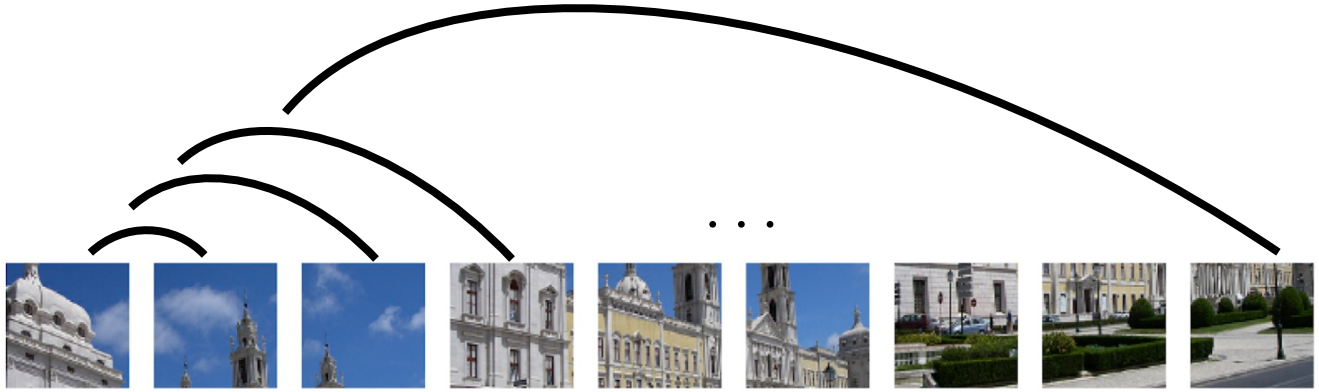
Agent Trajectory

Relation among Agent-Environment Interactions



... [S] [A] [R] ...

Quadratic Complexity in Self-Attention



Pair-wise Relation Modeling: $\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V$

General Relation Modeling

Quadratic Complexity

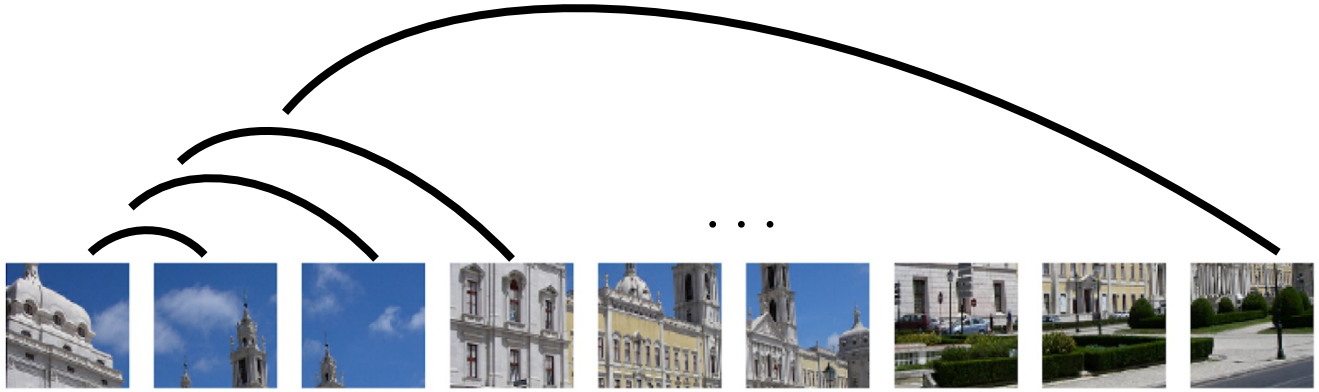
Long Sequence
Model Efficiency
Big Model ☹

Task & Data Universal

?

Foundation Model

Quadratic Complexity in Self-Attention



Pair-wise Relation Modeling: $\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V$

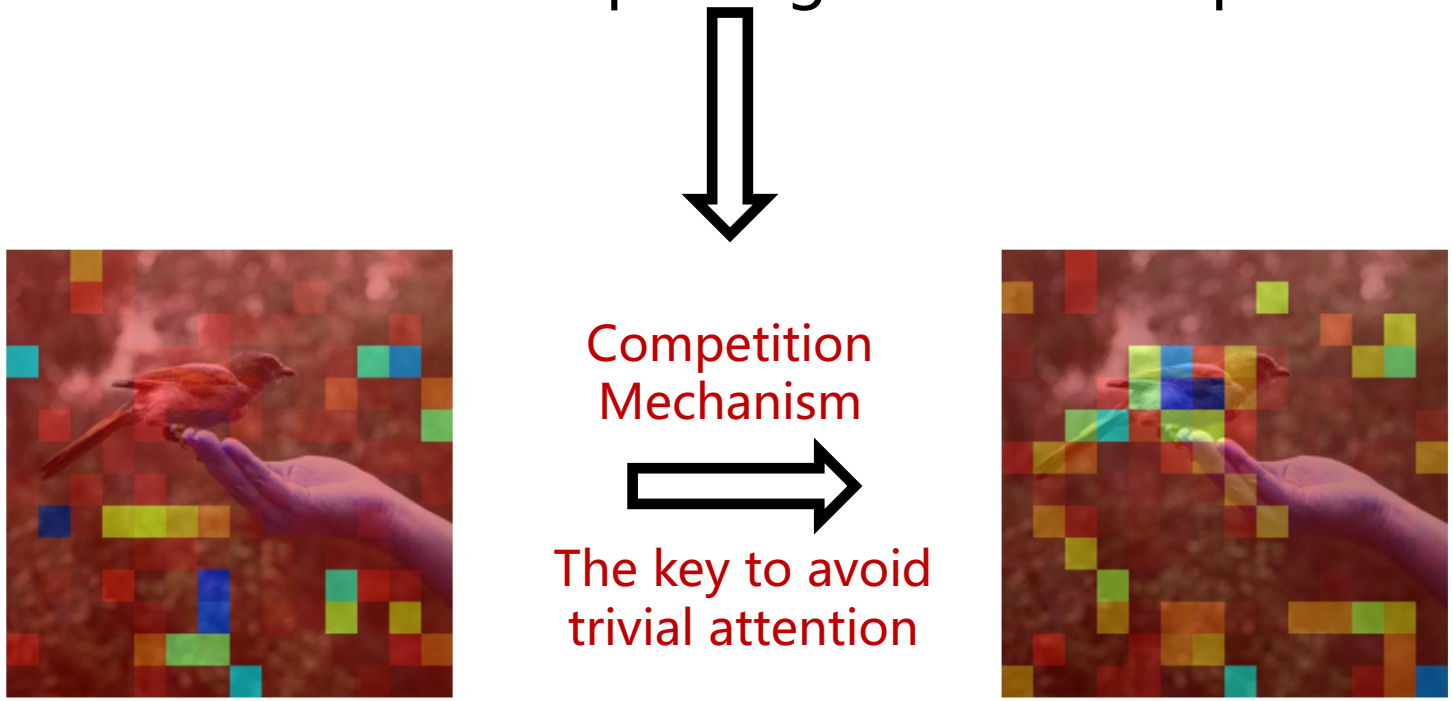
$\mathcal{O}(n^2 d)$

Can we remove Softmax function?

$(\boldsymbol{QK^T})\boldsymbol{V} = \boldsymbol{Q}(\boldsymbol{K^T V})$ ⇨ $\mathcal{O}(n^2 d) \rightarrow \mathcal{O}(nd^2)$

Recap: Softmax function

Softmax function is proposed as a differentiable generalization of the
"winner-take-all" picking maximum operation.



Bridle et al. Training stochastic model recognition algorithms as networks can lead to maximum mutual information estimation of parameters. NeurIPS 1989.

Recap: Softmax function

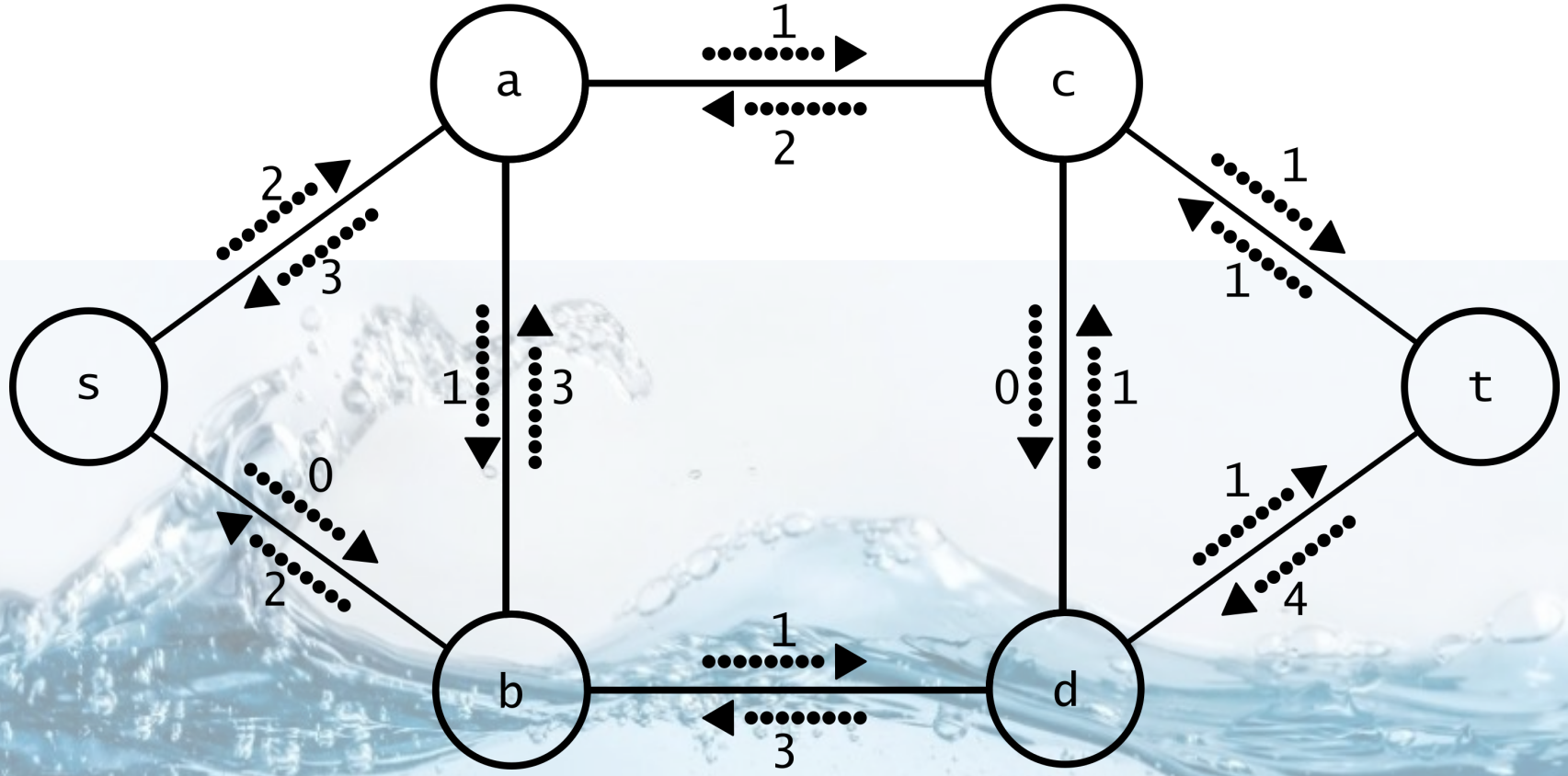
Softmax function is proposed as a differentiable generalization of the
"winner-take-all" picking maximum operation.

$$\boldsymbol{\phi}(\boldsymbol{Q})(\boldsymbol{\phi}(\boldsymbol{K})^{\boldsymbol{T}}\boldsymbol{V}) \quad + \quad \text{Competition Mechanism} \iff \textbf{Softmax}\left(\frac{\boldsymbol{QK^T}}{\sqrt{\boldsymbol{d}}}\right)\boldsymbol{V}$$

"fixed resource will cause competition"

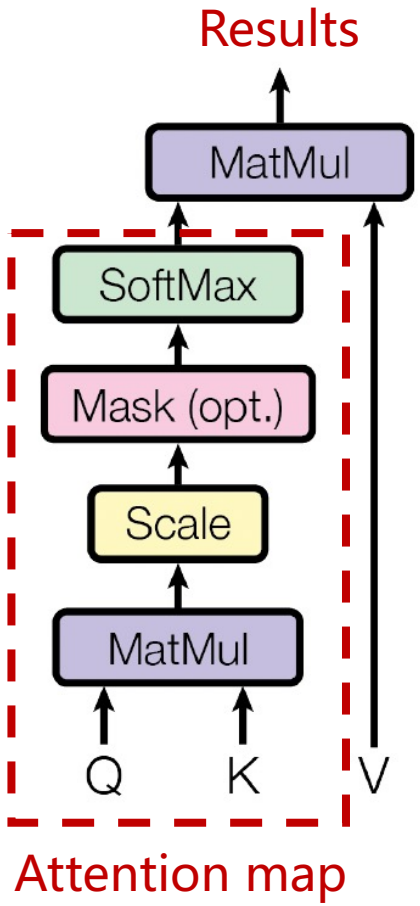
Bridle et al. Training stochastic model recognition algorithms as networks can lead to maximum mutual information estimation of parameters. NeurIPS 1989.

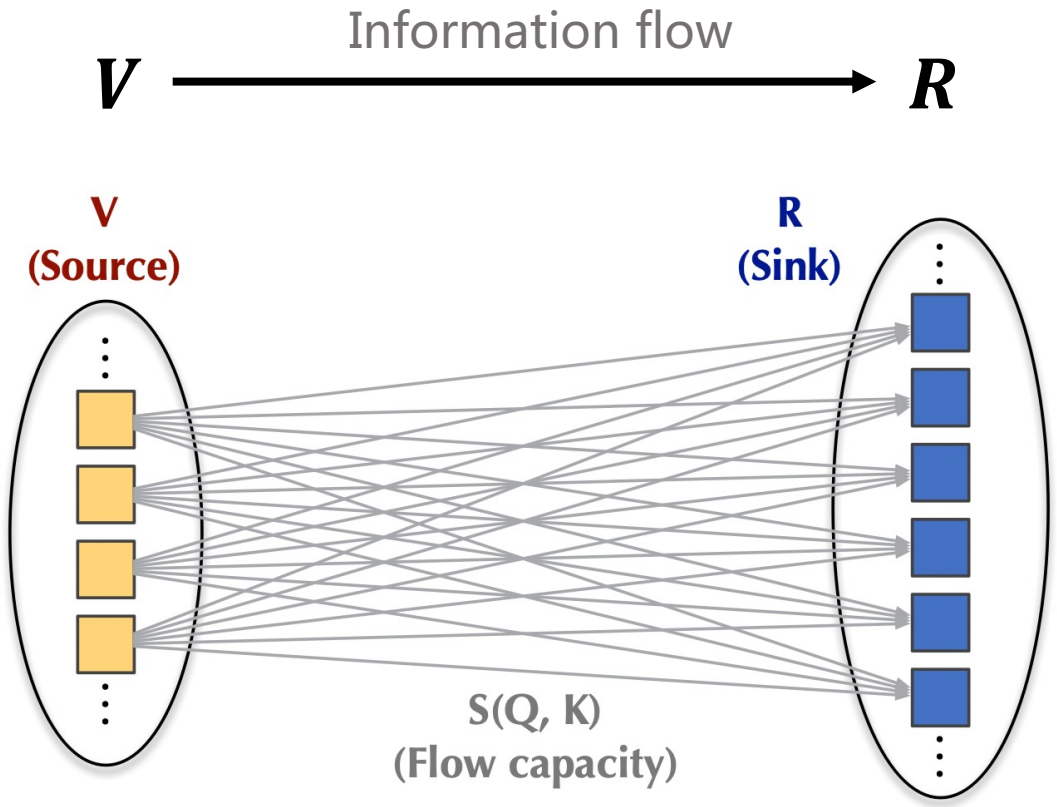
Flow Network Theory



[Conservation Property]: The incoming flow capacity of each node is equal to the outgoing flow.

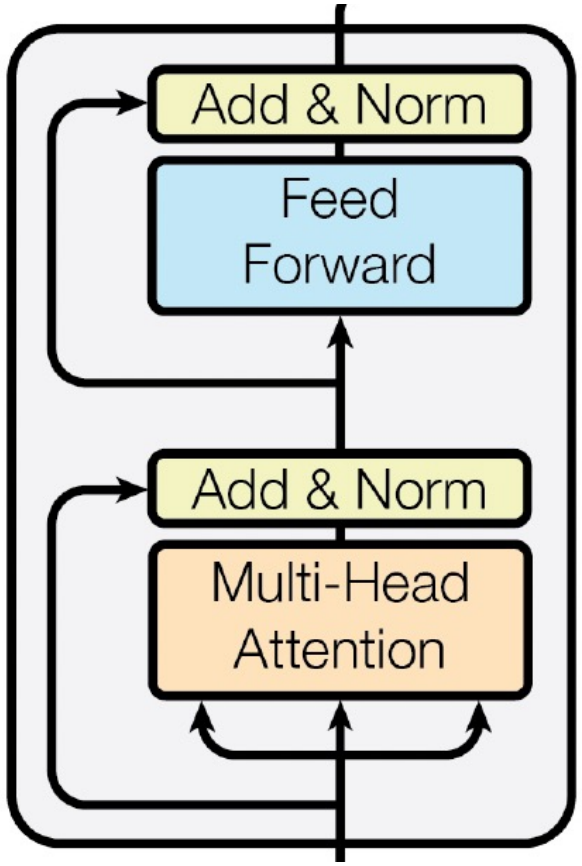
Attention: A Flow Network View

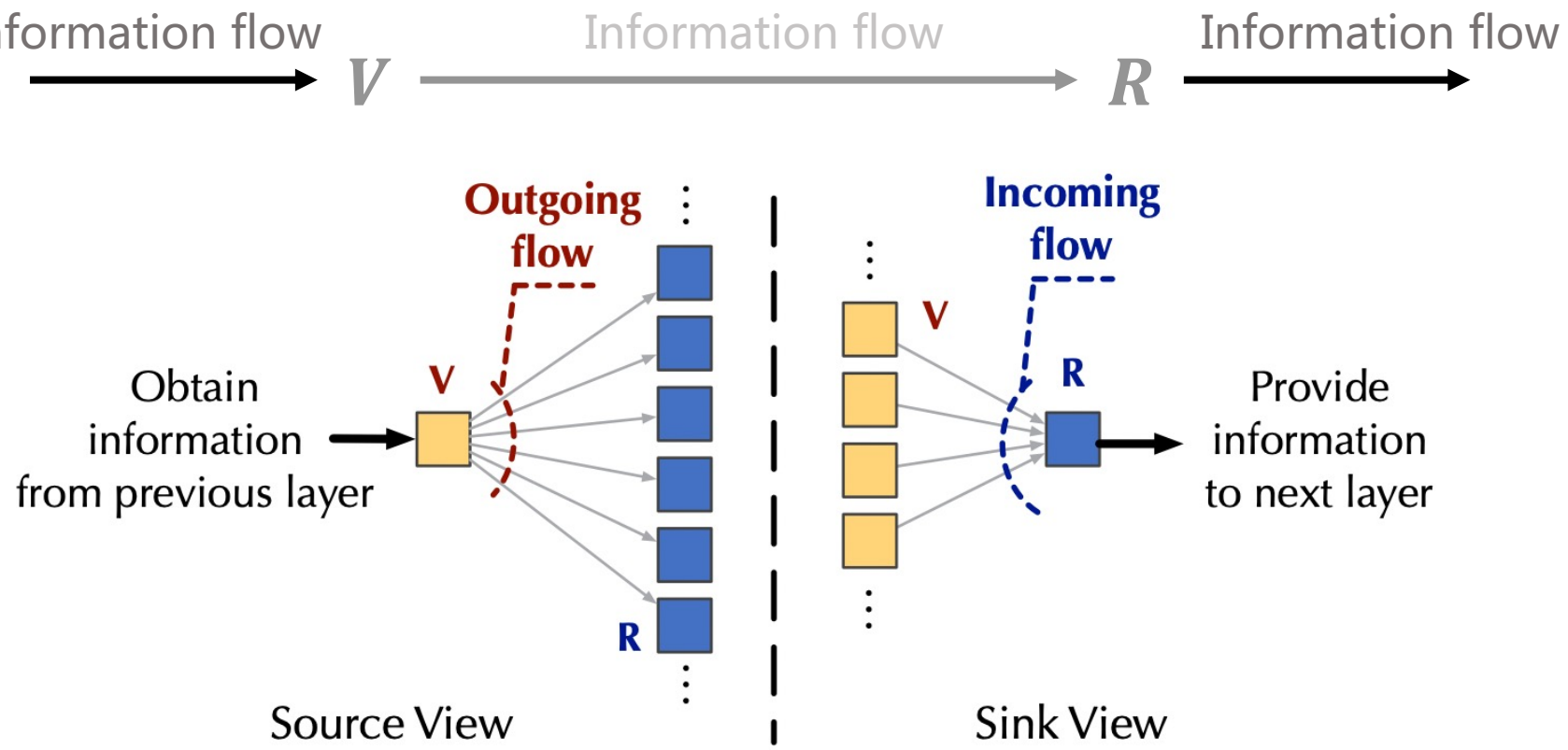




(a) Inner View

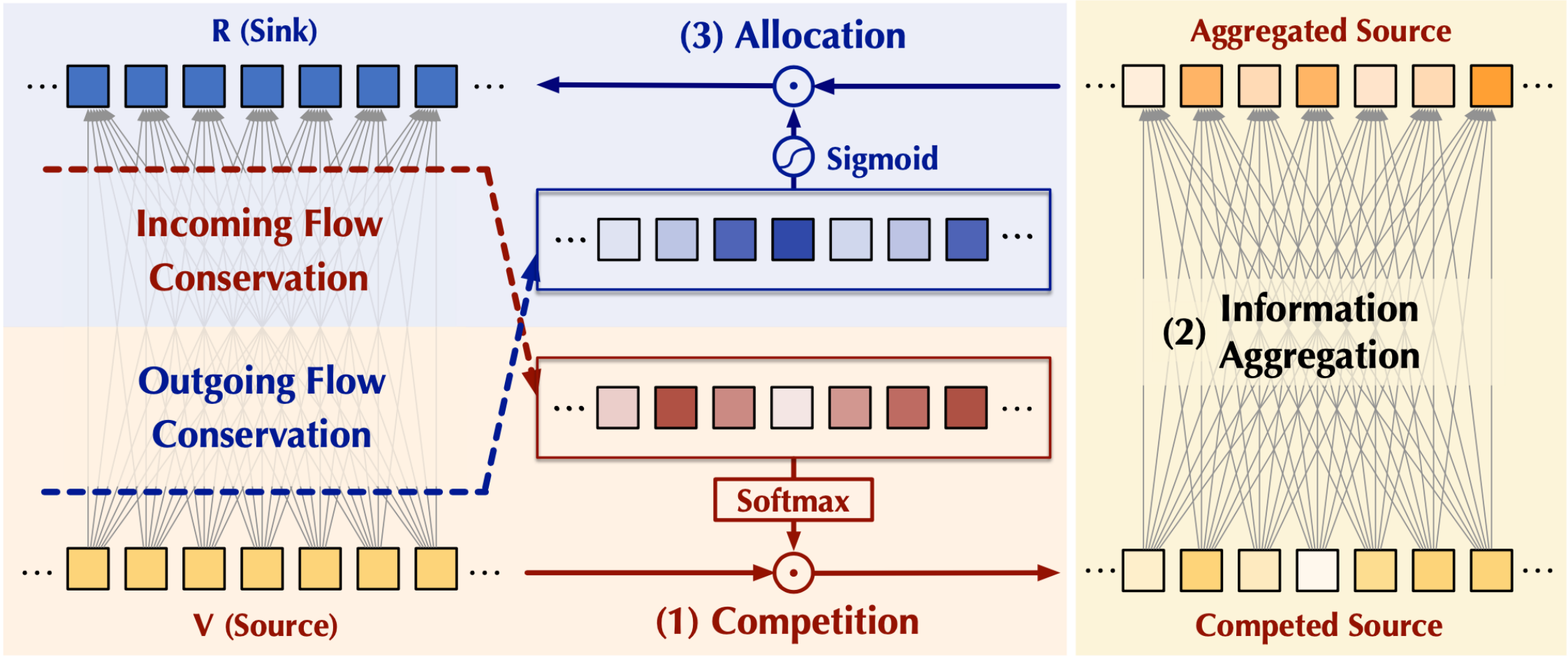
Attention: A Flow Network View





(b) Outer View

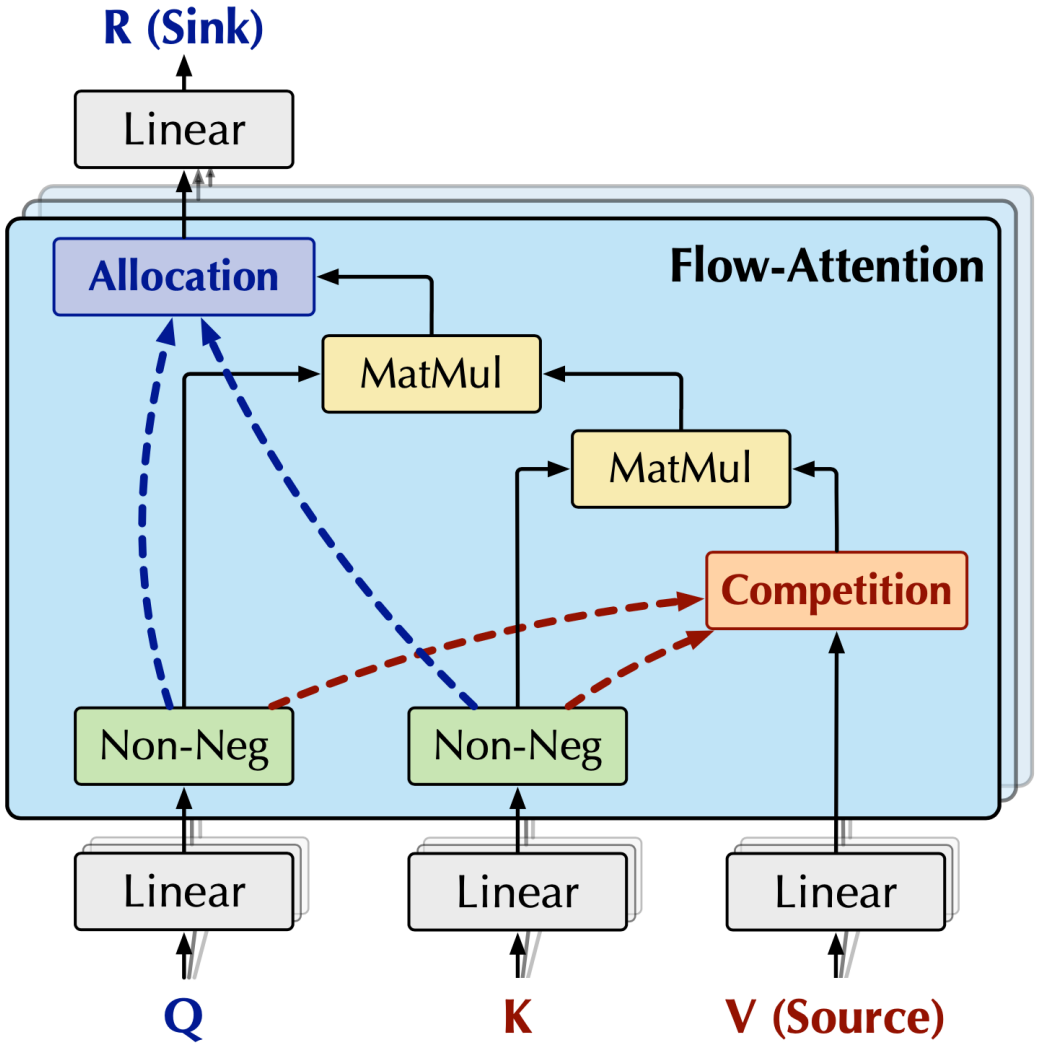
Flowformer: Conservation in Attention



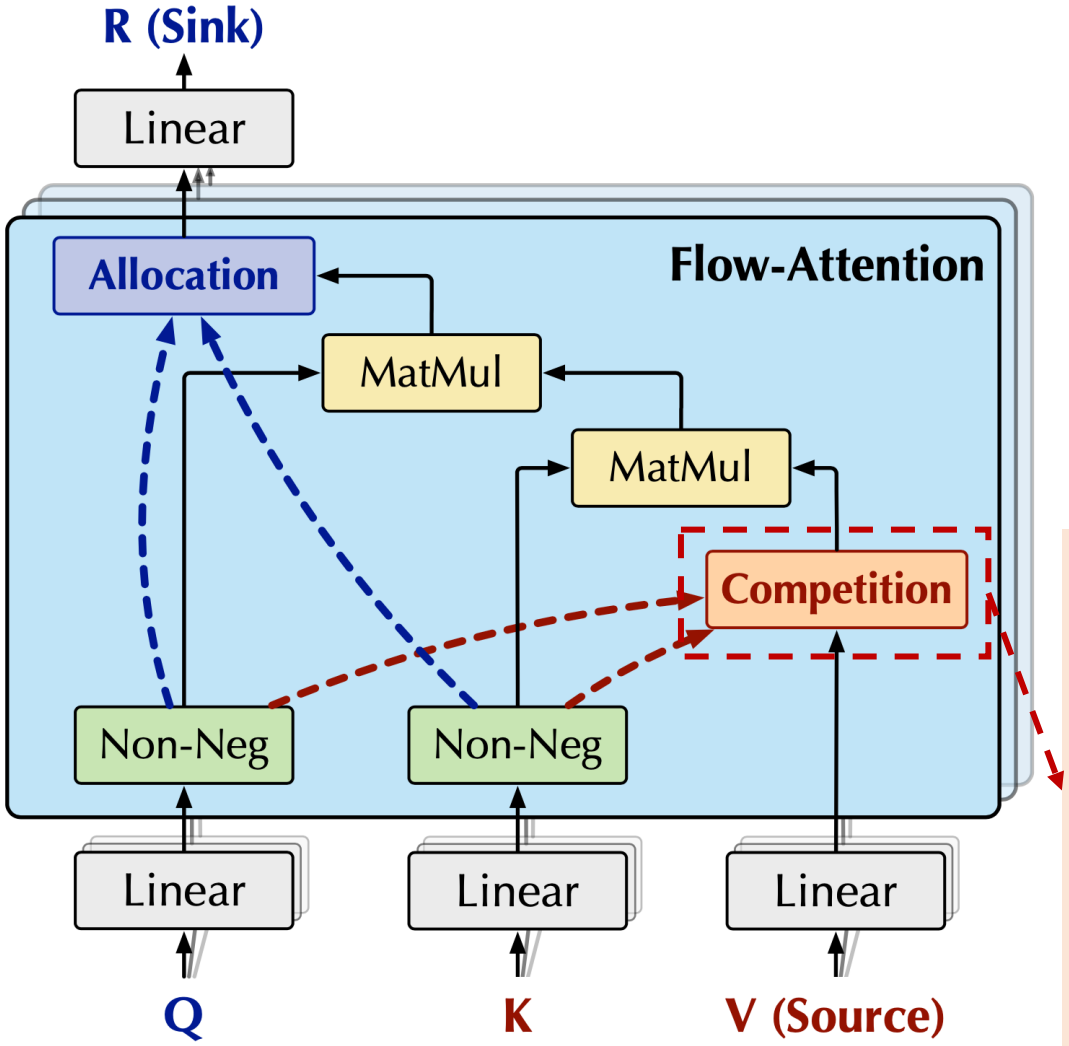
[Incoming Flow Conservation]: Competition among Source tokens

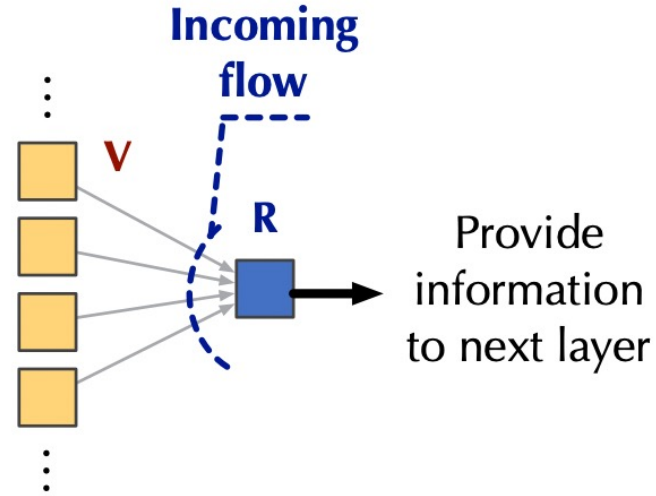
[Outgoing Flow Conservation]: Competition among Sink tokens

Flow-Attention



Flow-Attention



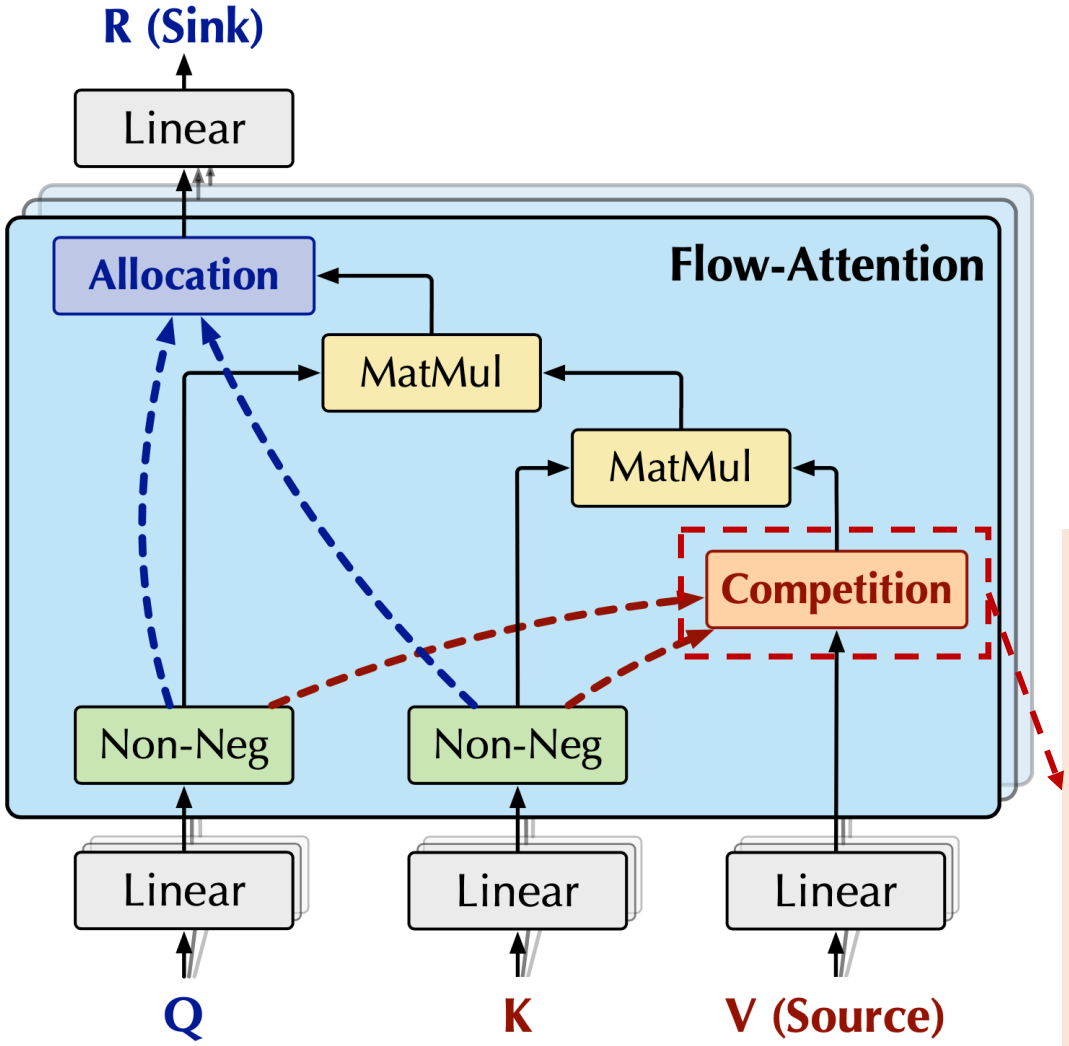


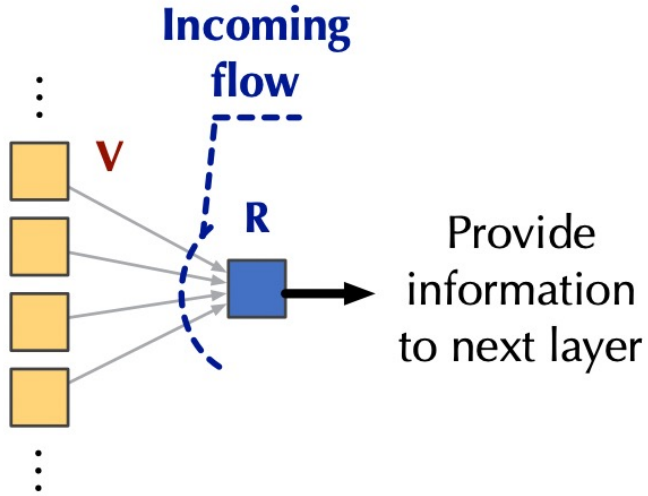
Incoming flow: $I_i = \phi(Q_i) \sum_j \phi(K_j)^T$

Incoming flow conservation: $\frac{\phi(\boldsymbol{Q})}{\boldsymbol{I}}$

Incoming flow: $\frac{\phi(Q_i)}{I_i} \sum_j \phi(K_j)^T = \frac{I_i}{I_i} = 1$

Flow-Attention



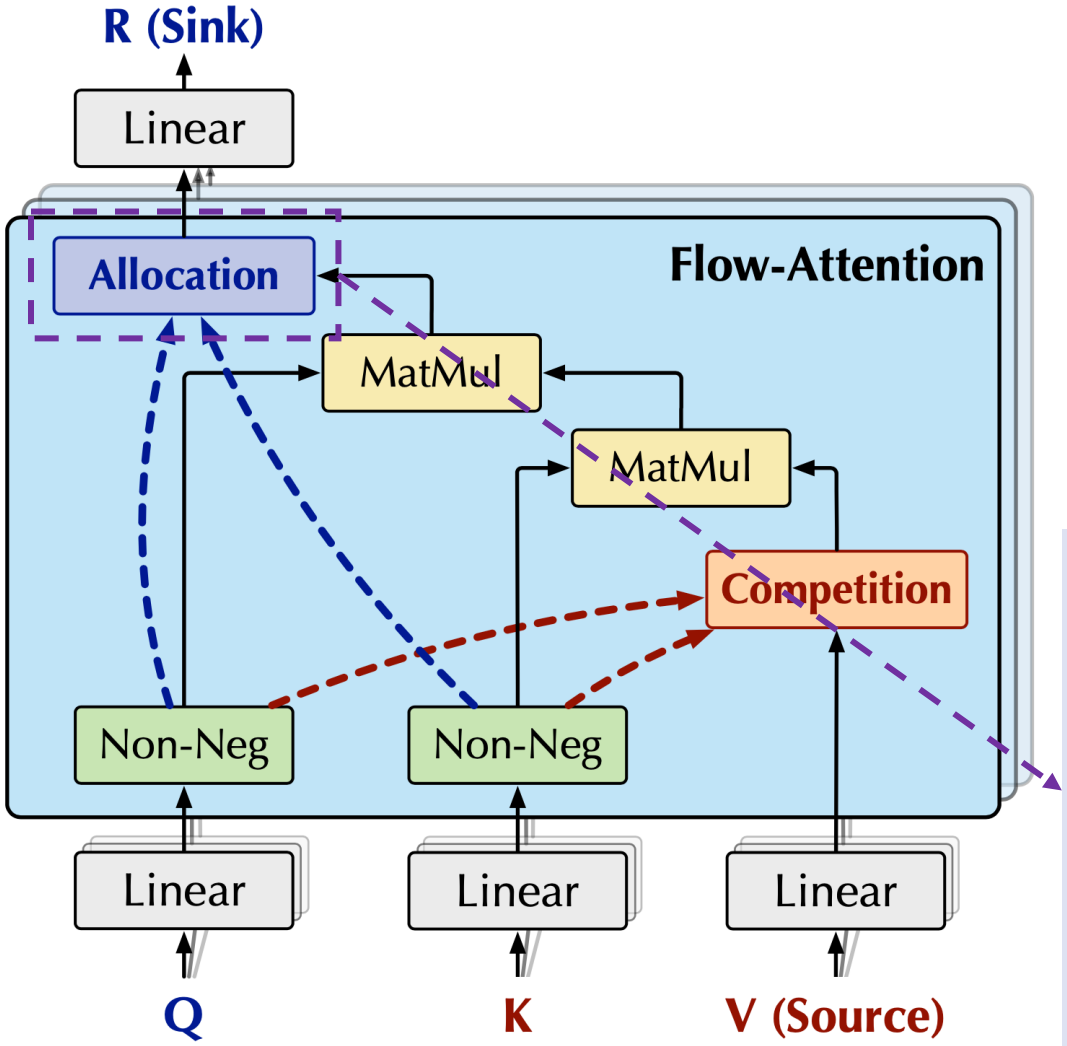


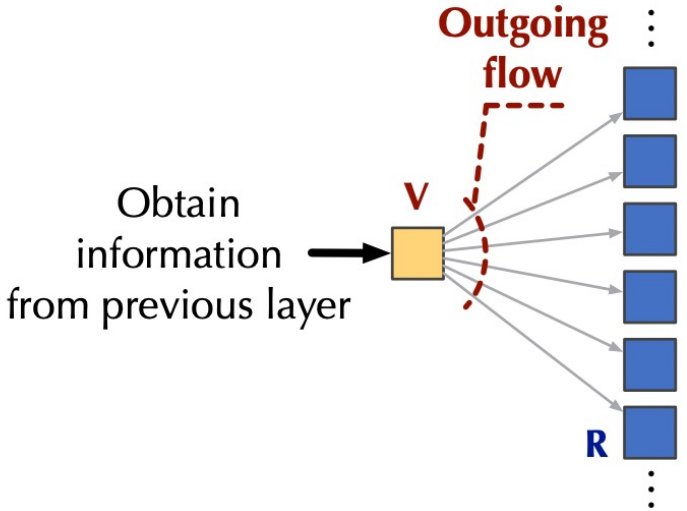
Incoming flow: $I_i = \phi(Q_i) \sum_j \phi(K_j)^T$

Incoming flow conservation: $\frac{\phi(\boldsymbol{Q})}{\boldsymbol{I}}$

Conserved outgoing flow: $\widehat{\boldsymbol{O}} = \phi(\boldsymbol{K}) \sum_i \frac{\phi(Q_i)^T}{I_i}$

Flow-Attention



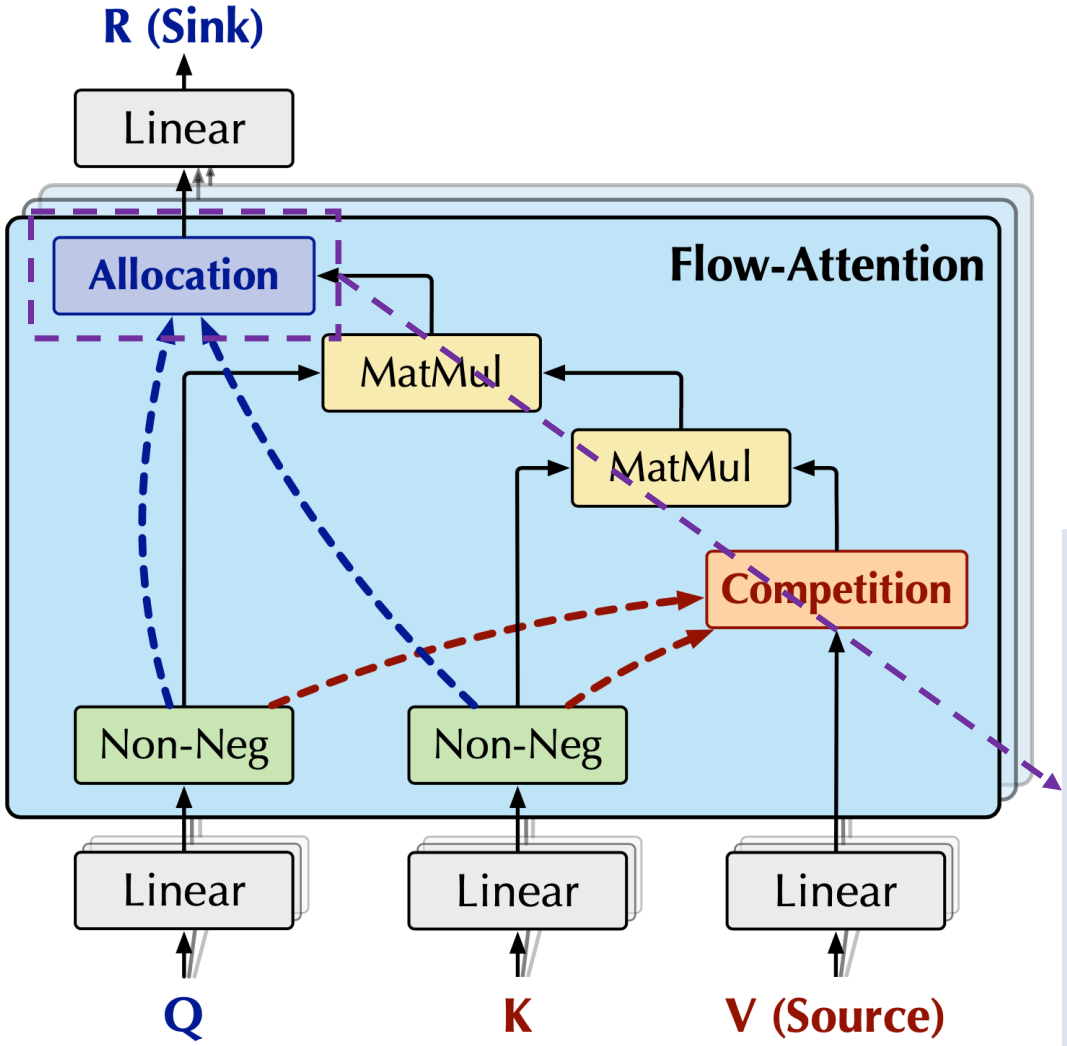


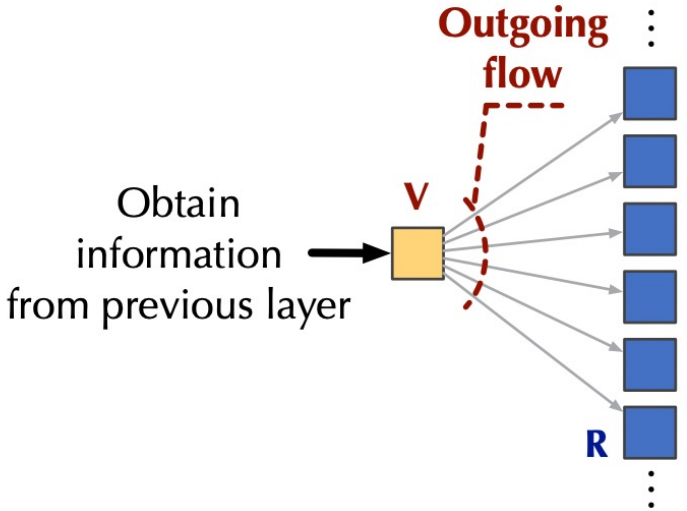
Outgoing flow: $O_i = \phi(K_i)\sum_j \phi(Q_j)^T$

Outgoing flow conservation: $\frac{\phi(\boldsymbol{K})}{\boldsymbol{O}}$

Outgoing flow: $\frac{\phi(K_i)}{O_i}\sum_j \phi(Q_j)^T = \frac{O_i}{O_i} = 1$

Flow-Attention



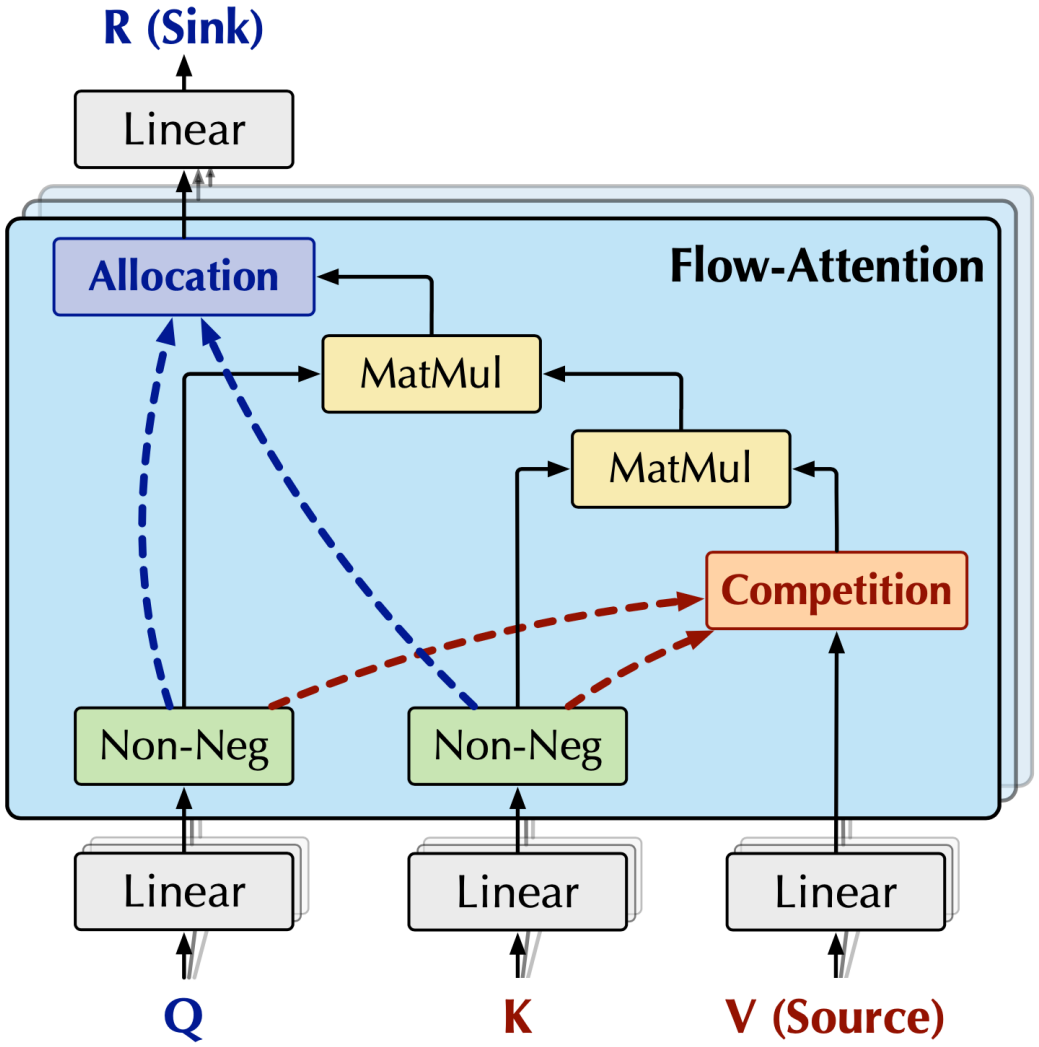


Outgoing flow: $O_i = \phi(K_i) \sum_j \phi(Q_j)^T$

Outgoing flow conservation: $\frac{\phi(\boldsymbol{K})}{\boldsymbol{O}}$

Conserved incoming flow: $\hat{\boldsymbol{I}} = \phi(\boldsymbol{Q}) \sum_j \frac{\phi(K_j)^T}{O_j}$

Flow-Attention



$$\text{Competition: } \hat{\mathbf{V}} = \text{Softmax}(\hat{\mathbf{O}}) \odot \mathbf{V}$$

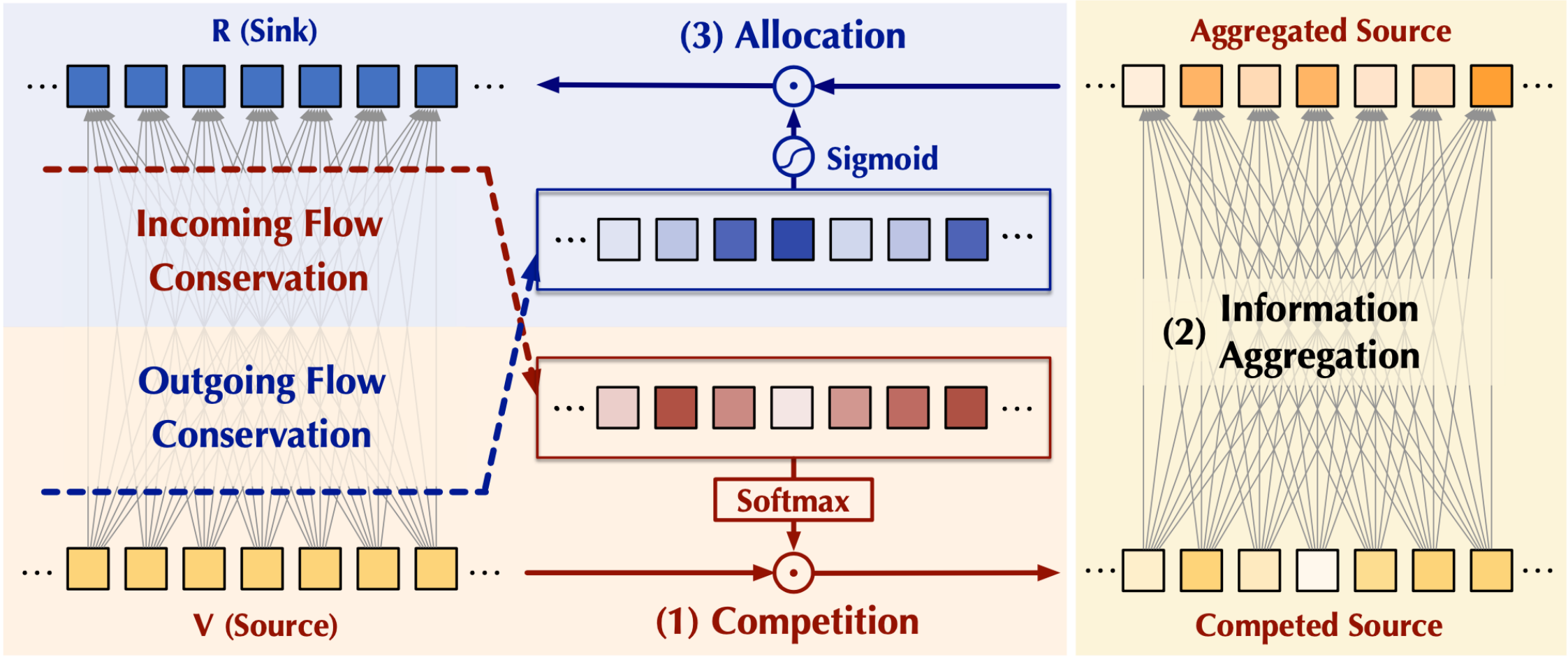
$$\text{Aggregation: } \mathbf{A} = \frac{\phi(\mathbf{Q})}{\mathbf{I}} \left(\phi(\mathbf{K})^{\top} \hat{\mathbf{V}}\right)$$

$$\text{Allocation: } \mathbf{R} = \text{Sigmoid}(\hat{\mathbf{I}}) \odot \mathbf{A},$$

Successfully bring the Competition Mechanism

Into Attention design to avoid trivial attention

Flowformer: Efficiency and Universality



[Efficiency]: All the calculations are in linear complexity.

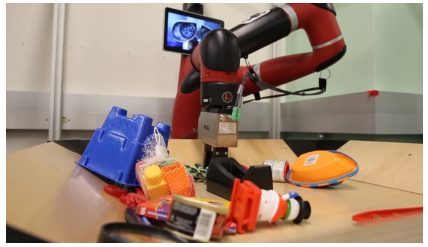
[Universality]: The whole design is based on flow network without specific inductive biases.

Flowformer Experiments

Image

Language

Time Series



Agent Trajectory

BENCHMARKS	TASK	VERSION	LENGTH
LRA (2020C)	SEQUENCE	NORMAL	1000∼4000
WIKITEXT (2017)	LANGUAGE	CAUSAL	512
IMAGENET (2009)	VISION	NORMAL	49∼3136
UEA (2018)	TIME SERIES	NORMAL	29∼1751
D4RL (2020)	OFFLINE RL	CAUSAL	60

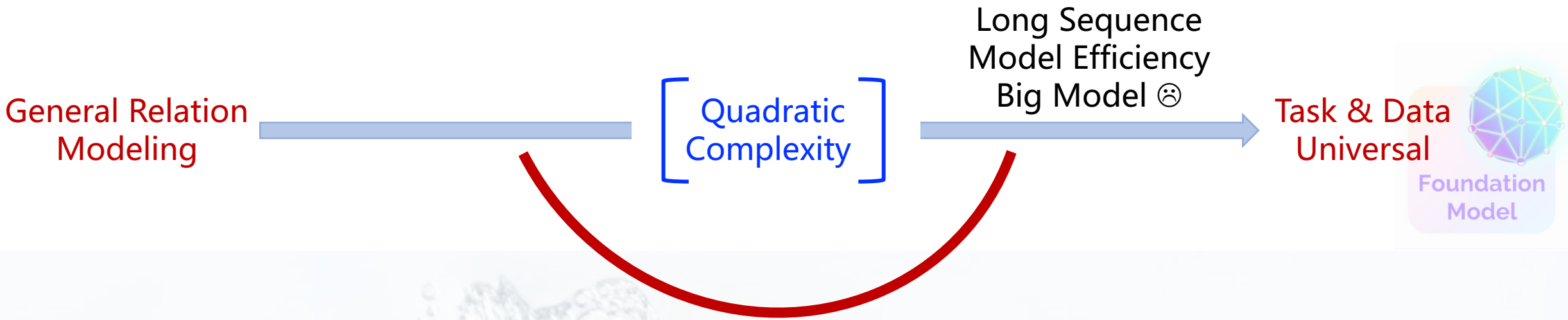
- Extensive tasks (covering 5 mainstream tasks)
- Normal and causal versions
- Various sequence lengths (29-4000)
- Extensive baselines (20+)

Flowformer Experiments

Task	Metrics	Flowformer	Performer	Reformer	Vanilla Transformer
Long Sequence Modeling (LRA)	Avg Acc (%) ↑	**56.48**	51.41	50.67	OOM
Vision Recognization (ImageNet-1K)	Top-1 Acc (%) ↑	**80.6**	78.1	79.6	78.7
Language Modeling (WikiText-103)	Perplexity ↓	**30.8**	37.5	33.6	33.0
Time series classification (UEA)	Avg Acc (%) ↑	**73.0**	71.5	71.9	71.9
Offline RL (D4RL)	Avg Reward ↑ Avg Deviation ↓	**73.5** ± 2.9	63.8 ± 7.6	63.9 ± 2.9	72.2 ± **2.6**

Strong performance on all five mainstream tasks within the linear complexity

Flowformer



Flowformer

Linear complexity w.r.t. sequence length

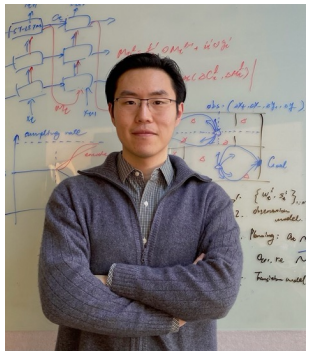
Based on flow network & without specific inductive biases

Strong performance in Long Sequence, CV, NLP, Time Series, RL

Haixu Wu, Jialong Wu, Jiehui Xu, Jianmin Wang, Mingsheng Long✉. Flowformer: Linearizing Transformers with Conservation Flows. ICML, 2022.

Thank You!

吴海旭



王韫博

龙明盛

王建民

大数据系统软件国家工程研究中心

清華大學
Tsinghua University