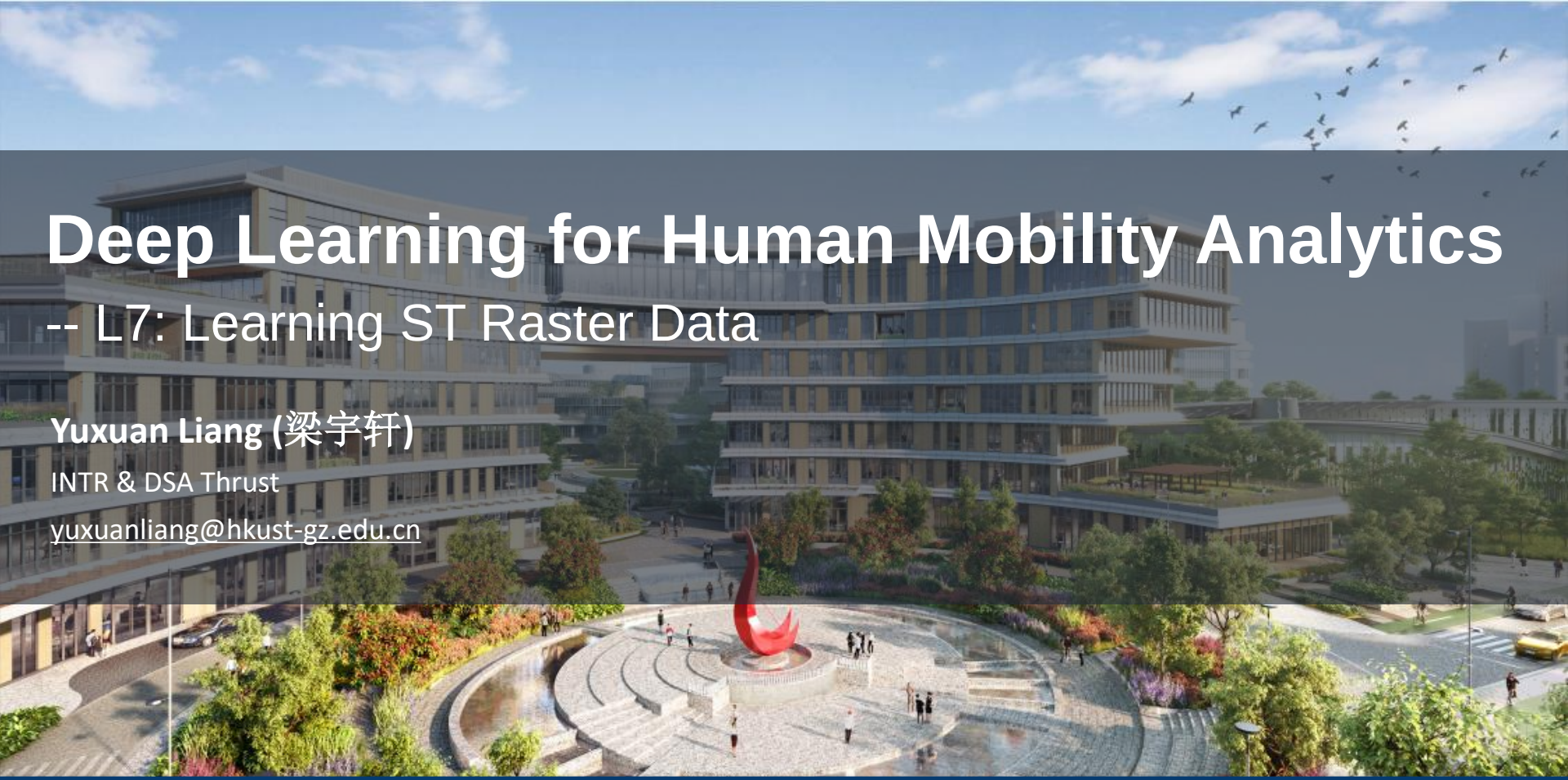


Deep Learning for Human Mobility Analytics

-- L7: Learning ST Raster Data

Yuxuan Liang (梁宇轩)

INTR & DSA Thrust

yuxuanliang@hkust-gz.edu.cn

香港科技大学(广州)
THE HONG KONG
UNIVERSITY OF SCIENCE AND
TECHNOLOGY (GUANGZHOU)

Course Reschedule

6	21 Oct	L6: Learning Spatio-Temporal Trajectory Data	✓	
7	28 Oct	L7: Learning Spatio-Temporal Raster Data	✓	Submit Design Report (by 28 Oct @ 23:59)

6:00pm-8:50pm
Nov. 1

Nov. 1

Course Reschedule

- Week 7: Learning Spatio-Temporal Raster Data
  - **6:00pm-8:50pm, Oct. 28 (today)**
  - The first part is as usual
  - The second part is **project discussion** (no presentation today!)
    - Team 1: Jiahui Liang & Gangyang Zhu
    - Team 2: Jiaxi Hu & Yongzi Yu
    - Team 3: Jingtao He & Pei Liu
    - Team 4: Ruiguo Zhong & Yixuan Wang
    - Team 5: Weilin Ruan & Qiongyan Wang
    - Team 6: Yongkai Gao
    - Team 7: Zhixiong Wang & Tianyu Wei

Course Reschedule

- Week 6: Learning Spatio-Temporal Trajectory Data
  - **6:00pm-8:50pm, Nov. 1**
  - The first part (~70 min) is as usual
  - The second part (~100 min) include presentation from both Week 6 and 7

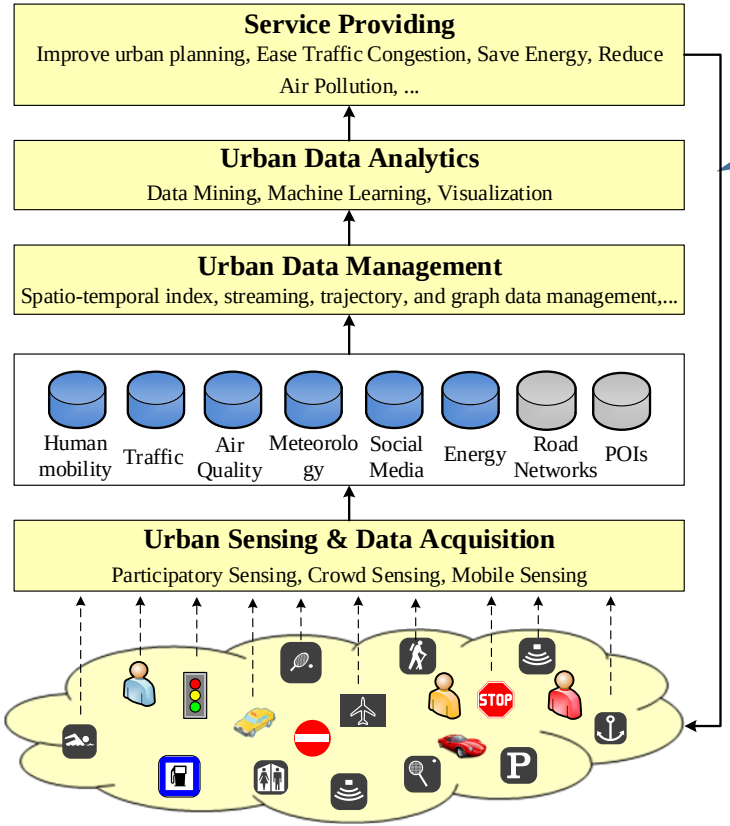
	1	TrajFormer: Efficient Trajectory Classification with Transformers	Link	CIKM	2022	NUS	Jingtao HE
	2	Stochastic Trajectory Prediction via Motion Indeterminacy Diffusion	Link	CVPR	2022	UCLA	Ruiguo ZHONG
Learning Spatio-Temporal Trajectory Data	3	Modeling Trajectories with Recurrent Neural Networks	Link	IJCAI	2017	Singapore Management University	Zhixiong Wang
	1	Urban regional function guided traffic flow prediction	Link	ıation Sci	2023	Sun Yat-sen University	Jiahui LIANG
Learning Spatio-Temporal Raster Data	2	UrbanFM: Inferring Fine-Grained Urban Flows		KDD	2019	NUS	Gangyong Zhu

Objectives of this Course

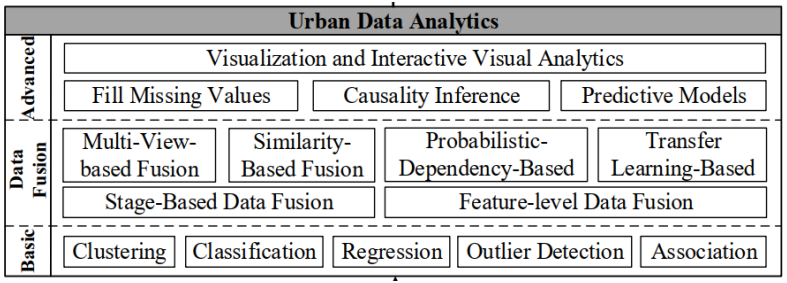
To introduce

- Convolutional neural networks for learning ST raster data
  - Conv2D-based
  - Conv3D-based
  - ConvLSTM-based
- Transformers for learning ST raster data
- Advanced learning framework for ST raster data
- Applications
  - Crowd flow forecasting, traffic prediction
  - Weather forecasting
  - Video recognition

3rd Stage: Urban Data Analytics



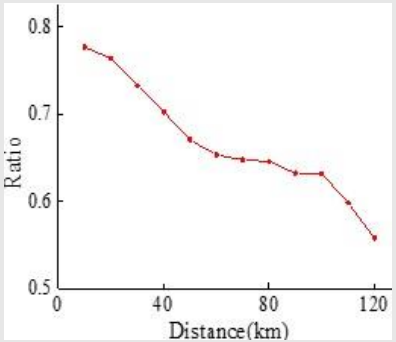
- Texts and images → spatio-temporal data
- A single data source → cross-domain data sources
- Separate data mining algorithms → ML + data management
- Visual and interactive data analytics



Spatio-Temporal Data is Unique

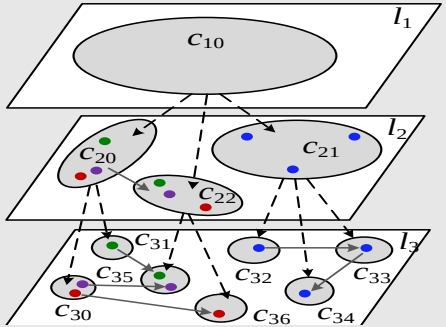
- Spatial property

Spatial closeness



Describing correlations

Spatial hierarchy



Structural constraints between different spatial granularity

Spatial distance

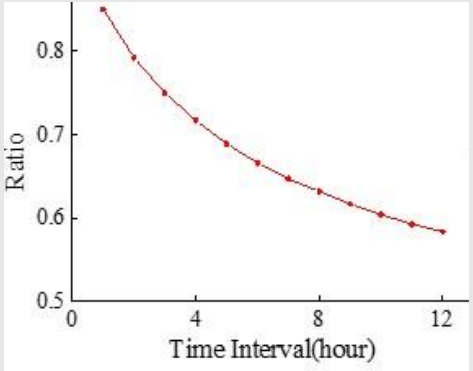
Triangle inequality:

$|d_1 - d_2| \leq d_3 \leq |d_1 + d_2|$

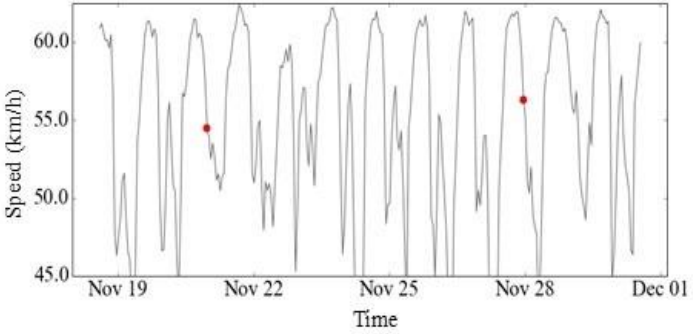
Spatio-Temporal Data is Unique

- Temporal property

Closeness

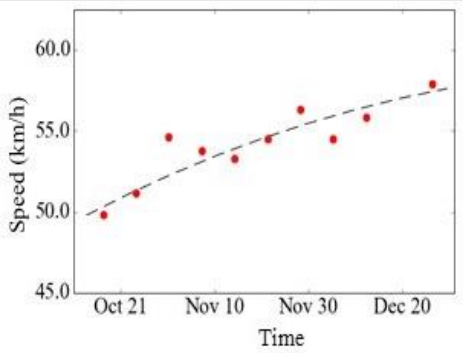


Periodicity



A) Hourly traffic speed on consecutive days

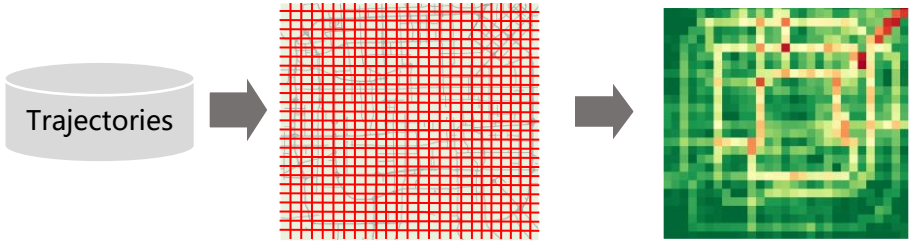
Trend



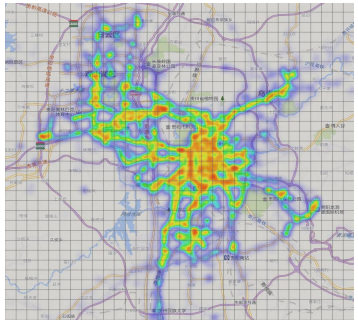
B) Traffic speed at 9-10am on consecutive weekends

Example of ST Raster Data: Transportation Domain

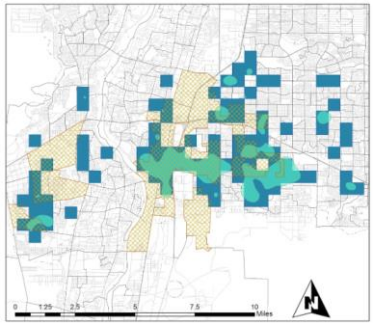
- We partition an area of interest (e.g., a metropolitan) evenly into grid cells, leading to an image-like data format called **ST grid**
  - A pixel → A region
  - RGB → Observations / Attributes



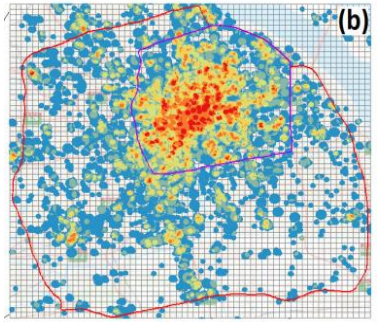
- Real-world examples



Taxi flows

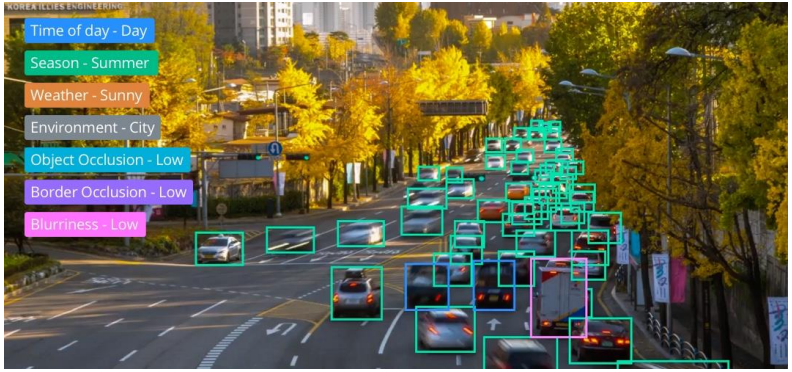


Crime hotspots

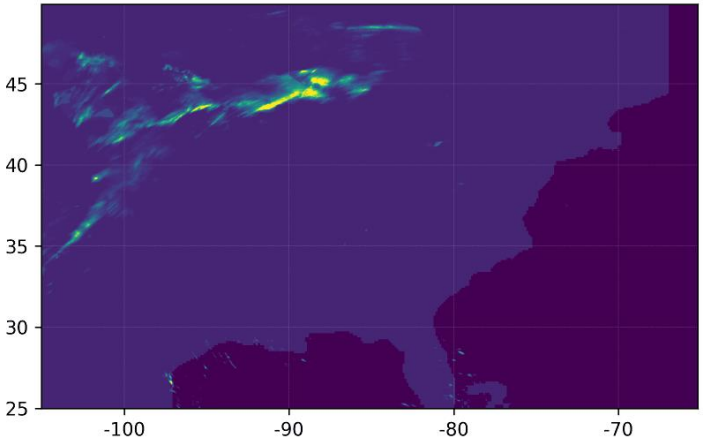


Bike-sharing demands

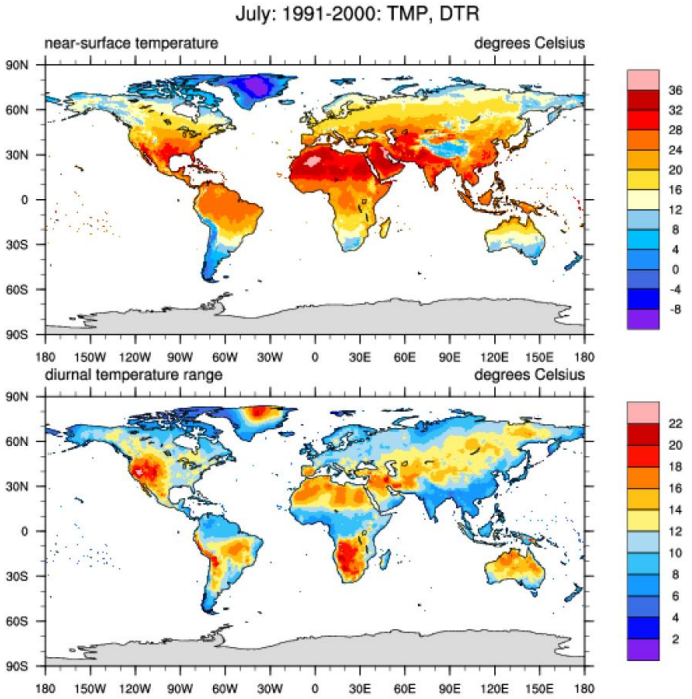
Example of ST Raster Data: Transportation Domain



Example of ST Raster Data: Climate Domain



Precipitation data

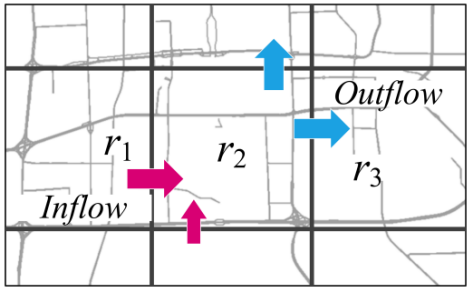


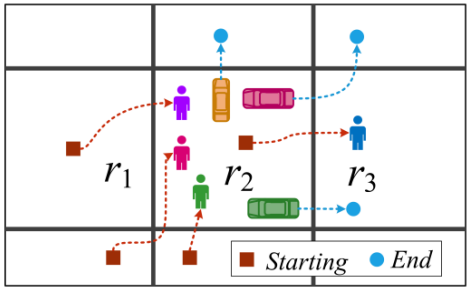
Meteorological data

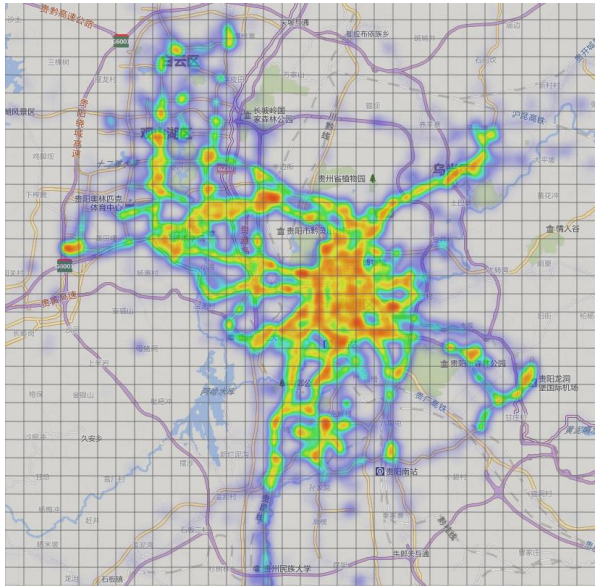
Application: Citywide Crowd Flow Forecasting

Grid-based citywide crowd flow prediction

- Predicting the inflow/outflow of every region in several hours



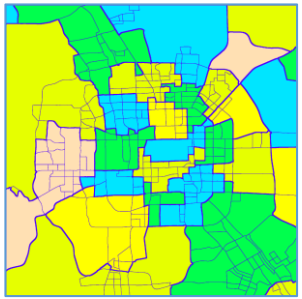


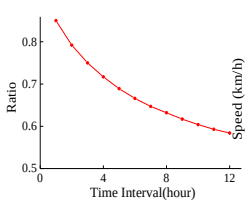


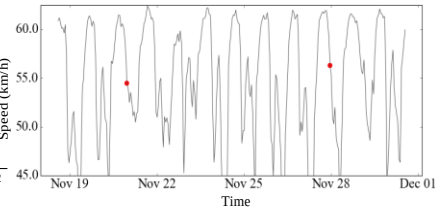
- It can provide insights to
  - Traffic control
  - Risk assessment
  - Public safety

Challenges

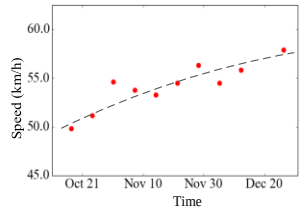
- Urban crowd flow depends on many factors
  - Flows of previous time interval
  - Flows of nearby regions and distant regions
  - Weather, traffic control and events
- Capturing spatial properties
  - Spatial distance and hierarchy
- Capturing temporal properties
  - Temporal closeness
  - Period and trend





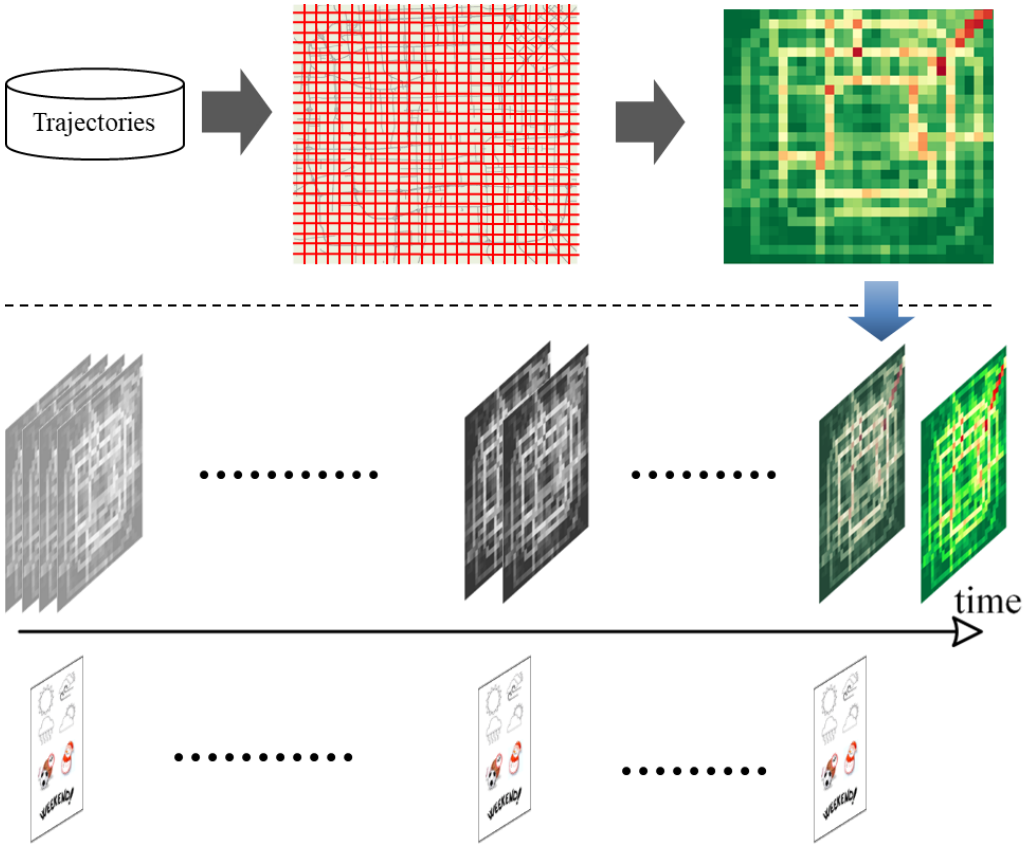


A) Hourly traffic speed on consecutive days

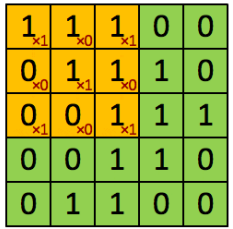


B) Traffic speed at 9-10am on consecutive weekends

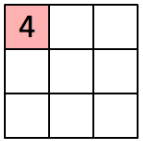
Converting Trajectories into Video-like Data



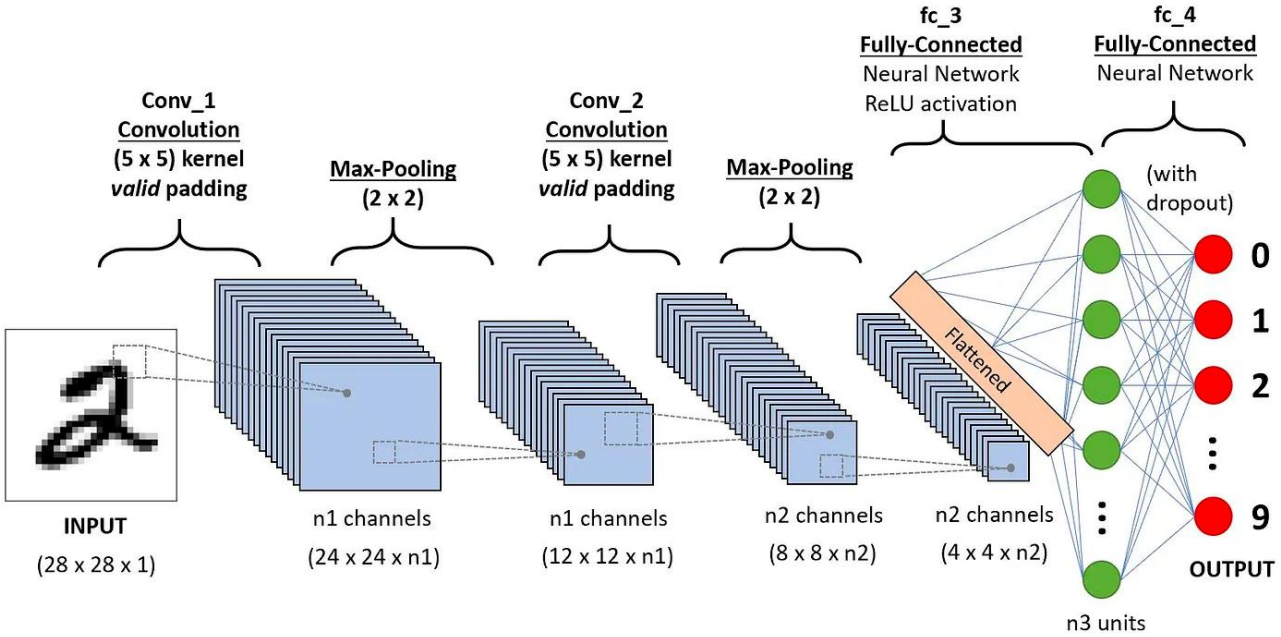
Preliminary: Convolutional Neural Network (CNN)



Image

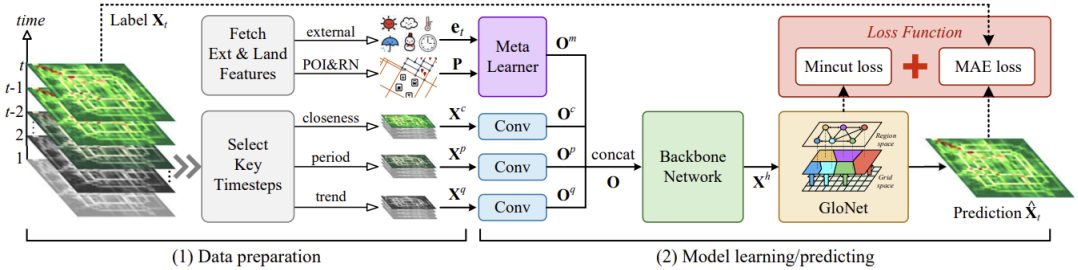


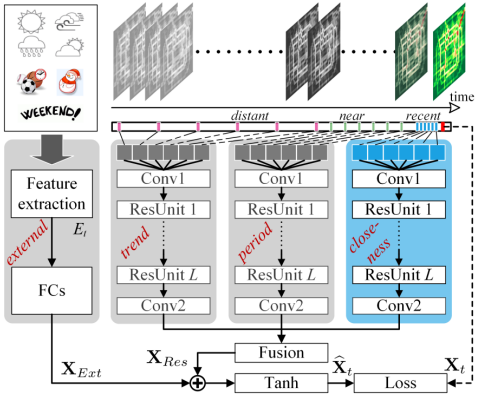
Convolved Feature

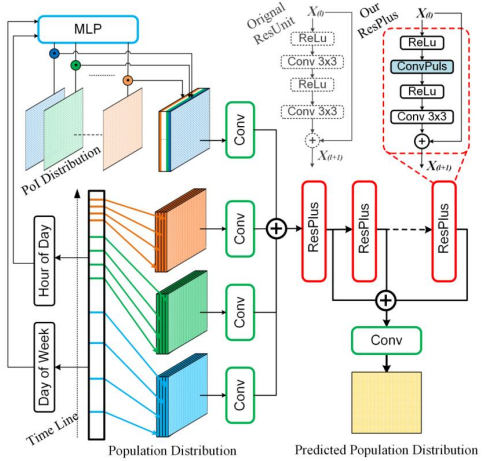


2D Convolution-based Solutions

- ST-ResNet [AAAI'17]
- DeepSTN+ [AAAI'18]
- STRN [WWW'21]

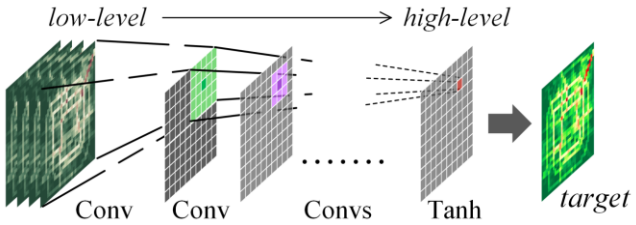


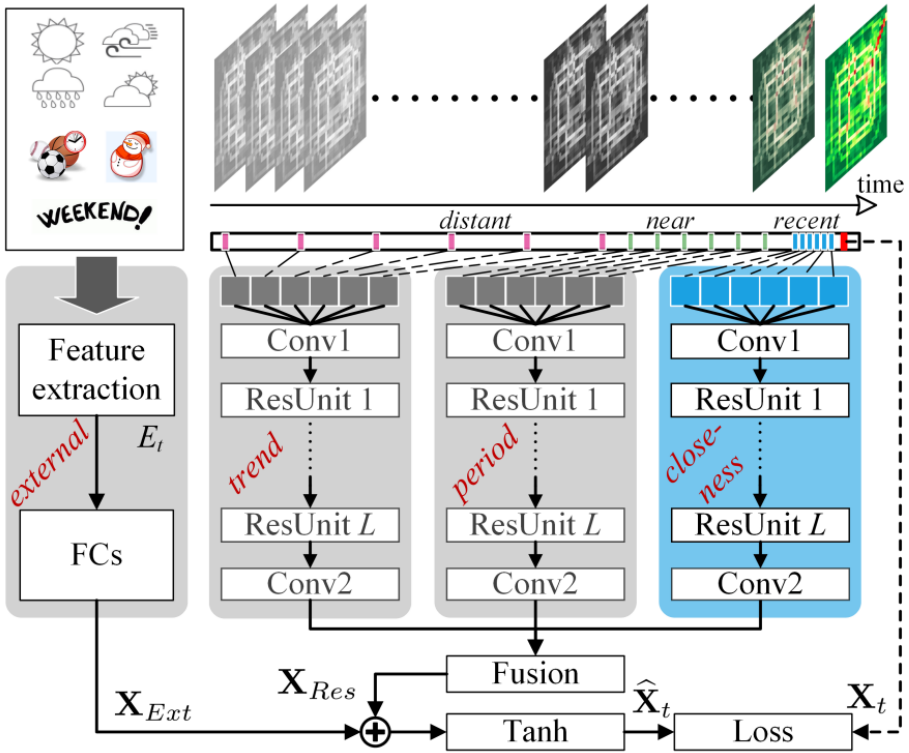




ST-ResNet

- Temporal dependencies
  - Distant
  - Near
  - Recent
- Spatial dependencies





J. Zhang et al., Deep Spatio-Temporal Residual Networks for Citywide Crowd Flows Prediction. AAAI 2017.

Experiments

- Datasets
  - TaxiBJ
  - BikeNYC

Table 1: Datasets (holidays include adjacent weekends).

Dataset	TaxiBJ	BikeNYC
Data type	Taxi GPS	Bike rent
Location	Beijing	New York
Time Span	7/1/2013 - 10/30/2013 3/1/2014 - 6/30/2014 3/1/2015 - 6/30/2015 11/1/2015 - 4/10/2016	4/1/2014 - 9/30/2014
Time interval	30 minutes	1 hour
Gird map size	(32, 32)	(16, 8)
Trajectory data		
Average sampling rate (s)	~ 60	\
# taxis/bikes	34,000+	6,800+
# available time interval	22,459	4,392
External factors (holidays and meteorology)		
# holidays	41	20
Weather conditions	16 types (*e.g.*, Sunny, Rainy)	\
Temperature / °C	$[-24.6, 41.0]$	\
Wind speed / mph	$[0, 48.6]$	\

Experiments

- Model comparison
- Ablation study

Table 2: Comparison among different methods on TaxiBJ

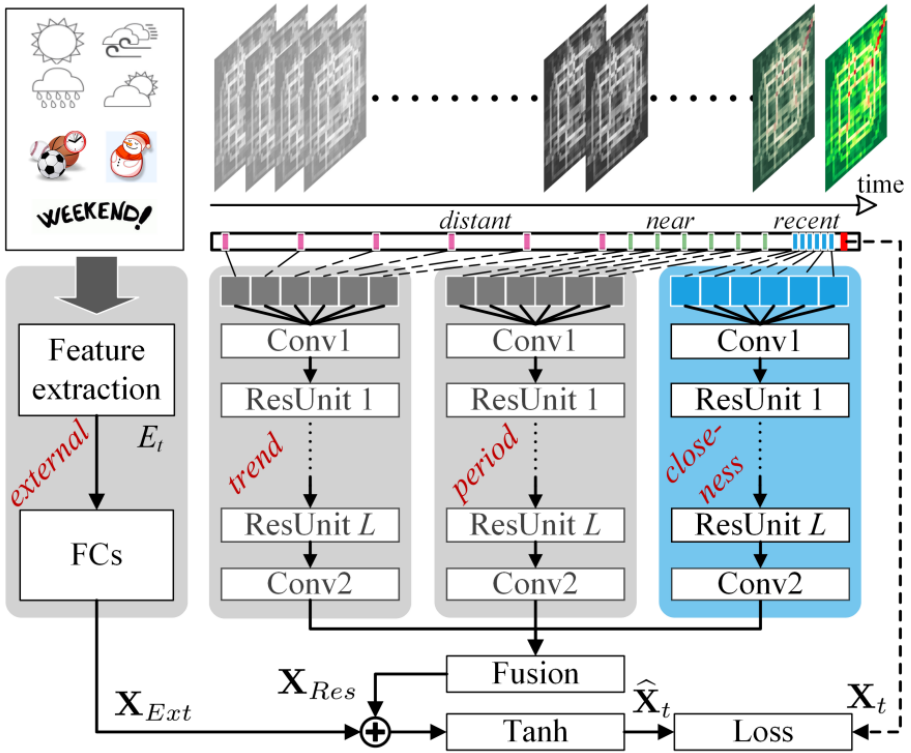
Model		RMSE
HA		57.69
ARIMA		22.78
SARIMA		26.88
VAR		22.88
ST-ANN		19.57
DeepST		18.18
	ST-ResNet [ours]	
L2-E	2 residual units + E	17.67
L4-E	4 residual units + E	17.51
L12-E	12 residual units + E	16.89
L12-E-BN	L12-E with BN	**16.69**
L12-single-E	12 residual units (1 conv) + E	17.40
L12	12 residual units	17.00
L12-E-noFusion	12 residual units + E without fusion	17.96

Table 3: Comparisons with baselines on BikeNYC. The results of ARIMA, SARIMA, VAR and 4 DeepST variants are taken from (Zhang et al. 2016).

Model	RMSE
ARIMA	10.07
SARIMA	10.56
VAR	9.92
DeepST-C	8.39
DeepST-CP	7.64
DeepST-CPT	7.56
DeepST-CPTM	7.43
ST-ResNet [ours, 4 residual units]	**6.33**

Drawbacks

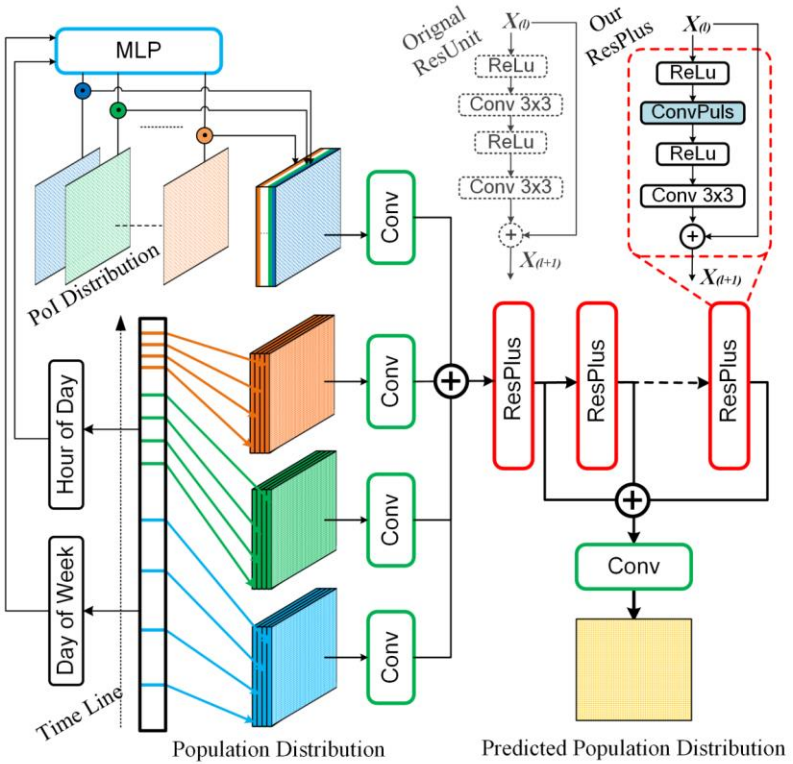
- Drawbacks of ST-ResNet
  - Cannot effectively handle long-range spatial dependencies
  - Hurdle of late fusion
  - No consideration of POIs



Z. Lin et al., DeepSTN+: Context-aware Spatial-Temporal Neural Network for Crowd Flow Prediction in Metropolis. AAAI 2018.

DeepSTN+

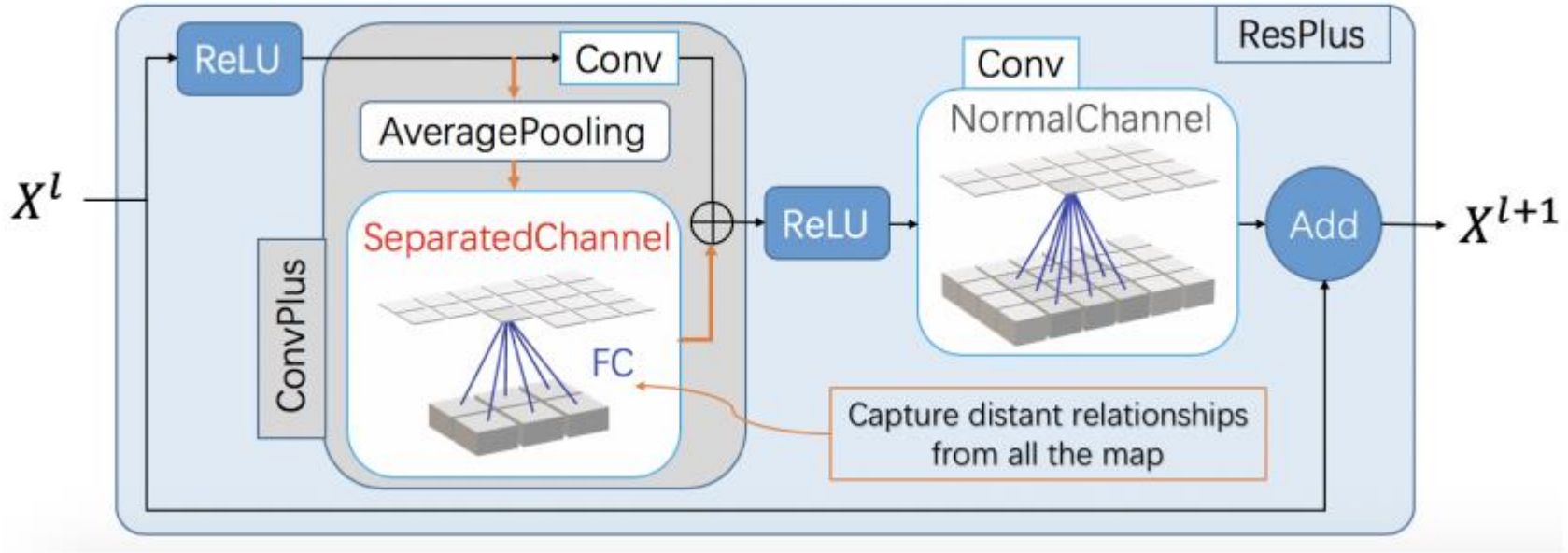
- Framework



Z. Lin et al., DeepSTN+: Context-aware Spatial-Temporal Neural Network for Crowd Flow Prediction in Metropolis. AAAI 2018.

DeepSTN+

- ResPlus Unit



DeepSTN+

- Training

Procedure: DeepSTN+ Training Procedure

Input: historical observations: $\{\mathbf{X}_0, \cdots, \mathbf{X}_{n-1}\}$;
PoI distributions: $\mathbf{X}^s$; time-vector:$\{\mathbf{I}_0, \cdots, \mathbf{I}_{n-1}\}$;
length of closeness, period, trend sequences: lc, lp, lt;
period span: p; trend span: t.

Output: Learned DeepSTN+ model

// construct the training data $\mathbb{D}$

$\mathbb{D} \Leftarrow \emptyset$

for all available time interval:

  $\mathbf{X}_i^c = [\mathbf{X}_{i-l_c}, \mathbf{X}_{i-(l_c-1)}, \cdots, \mathbf{X}_{i-1}]$

  $\mathbf{X}_i^p = [\mathbf{X}_{i-l_p \cdot p}, \mathbf{X}_{i-(l_p-1)\cdot p}, \cdots, \mathbf{X}_{i-p}]$

  $\mathbf{X}_i^t = [\mathbf{X}_{i-l_t \cdot t}, \mathbf{X}_{i-(l_t-1)\cdot t}, \cdots, \mathbf{X}_{i-t}]$

  put an training instance $(\{\mathbf{X}_i^c, \mathbf{X}_i^p, \mathbf{X}_i^t, \mathbf{X}^s, \mathbf{I}_i\}, \mathbf{X}_i)$ into $\mathbb{D}$

end // $\mathbf{X}_i$ is the target at time i

// train the model

initialize all learnable parameters θ in DeepSTN+

repeat

  randomly select a batch of instances $\boldsymbol{D}$ from $\mathbb{D}$

  optimize θ using Adam and $\boldsymbol{D}$

until model overfitting

Datasets

- Mobile social networks from Beijing
- Bike system in New York

Dataset	MobileBJ	BikeNYC
Data type	Mobile application	Bike rent
Location	Beijing	New York
Time span	4/1/2018-4/30/2018	4/1/2014-9/30/2014
Time interval	30 minutes	1 hour
Grid map size	(19,21)	(21,12)
PoI Num	264581	26202

Dataset	Point of Interests (PoI)
BikeNYC	Food, Residence, ShopServic, CollegeUniversity, NightlifeSpot, TravelTransport, ArtEntertainment, ProfessionalOtherPlace, OutdoorsRecreation
MobileBJ	Food, Hotel, Culture, Sports, Shopping, Factory, Recreation, Institution, MedicalCare, ScenicSpot, Education, Landmark, Residence, TravelTransport, BusinessAffairs, LifeService

Experiments

- Model comparison
- Ablation study

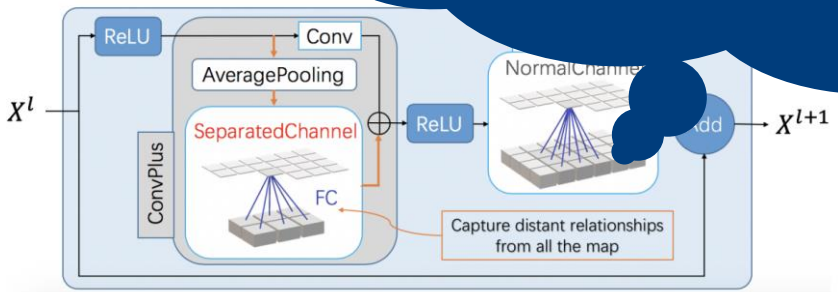
Model	RMSE	Δ	MAE
HA	7.885	21.79%	2.823
VAR	10.097	55.94%	5.49
ARIMA	10.894	68.25%	3.246
ConvLSTM	6.412	-0.97%	2.543
ST-ResNet	6.475	0	2.395
DeepSTN	6.213	-4.05%	2.388
DeepSTN+plus	6.128	-5.36%	2.362
DeepSTN+PoI	6.191	-4.39%	2.381
DeepSTN+PoI∗time	6.021	-7.01%	2.340
DeepSTN+plus+PoI∗time	**5.984**	**-7.58%**	**2.292**
DeepSTN+plus+PoI∗time+con	**5.955**	**-8.03%**	**2.285**

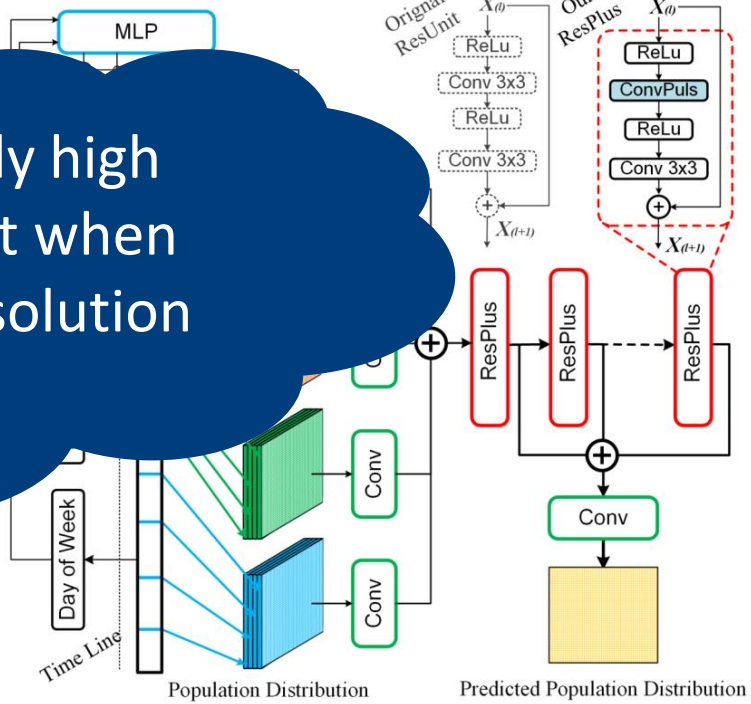
Model	RMSE	Δ	MAE
HA	7.885	21.79%	2.823
VAR	10.097	55.94%	5.49
ARIMA	10.894	68.25%	3.246
ConvLSTM	6.412	-0.97%	2.543
ST-ResNet	6.475	0	2.395
DeepSTN	6.213	-4.05%	2.388
DeepSTN+plus	6.128	-5.36%	2.362
DeepSTN+PoI	6.191	-4.39%	2.381
DeepSTN+PoI∗time	6.021	-7.01%	2.340
DeepSTN+plus+PoI∗time	**5.984**	**-7.58%**	**2.292**
DeepSTN+plus+PoI∗time+con	**5.955**	**-8.03%**	**2.285**

DeepSTN+

- Drawbacks of ST-ResNet
  - Cannot effectively handle l
    range spatial
  - Hurdle of
  - No conside

Inducing extremely high computational cost when processing high-resolution traffic data!

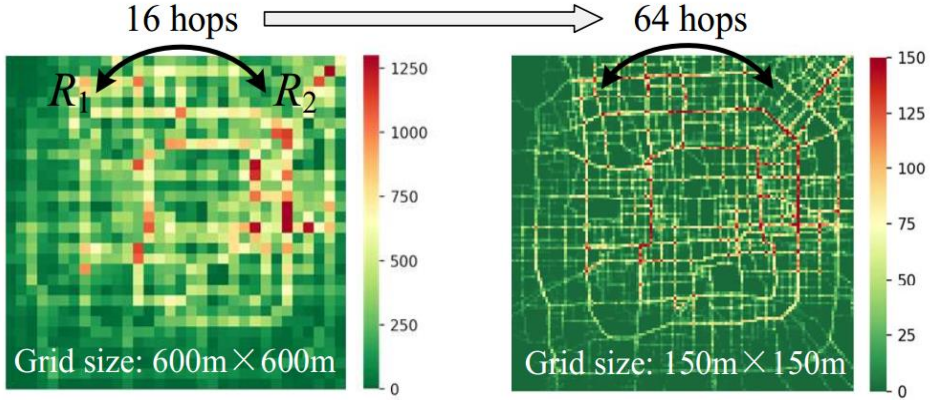




Z. Lin et al., DeepSTN+: Context-aware Spatial-Temporal Neural Network for Crowd Flow Prediction in Metropolis. AAAI 2018.

Fine-Grained Urban Flow Prediction

- Urban flow prediction benefits smart cities in many aspects. However, a critical prerequisite is having fine-grained knowledge of the city
  - Acquiring the traffic in a small area of interest can help allocate police resources more precisely while knowing that information at a district level is less useful.

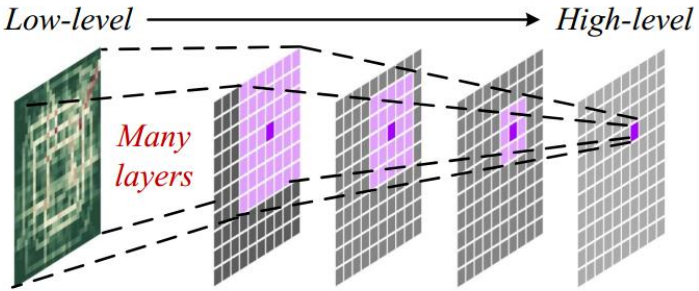


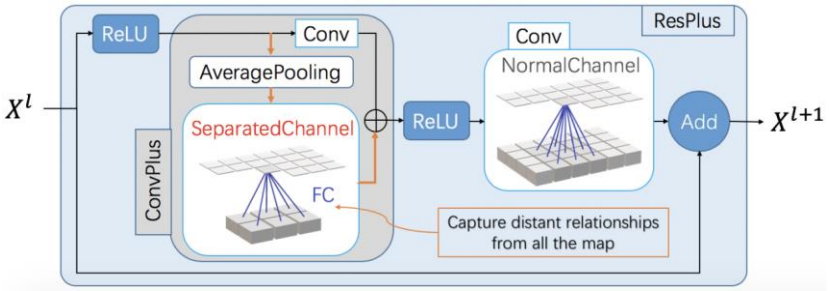
(a) Coarse-grained flows (b) Fine-grained flows

Y. Liang, R. Zimmermann et al., Fine-grained Urban Flow Prediction. WWW 2021.

Challenge

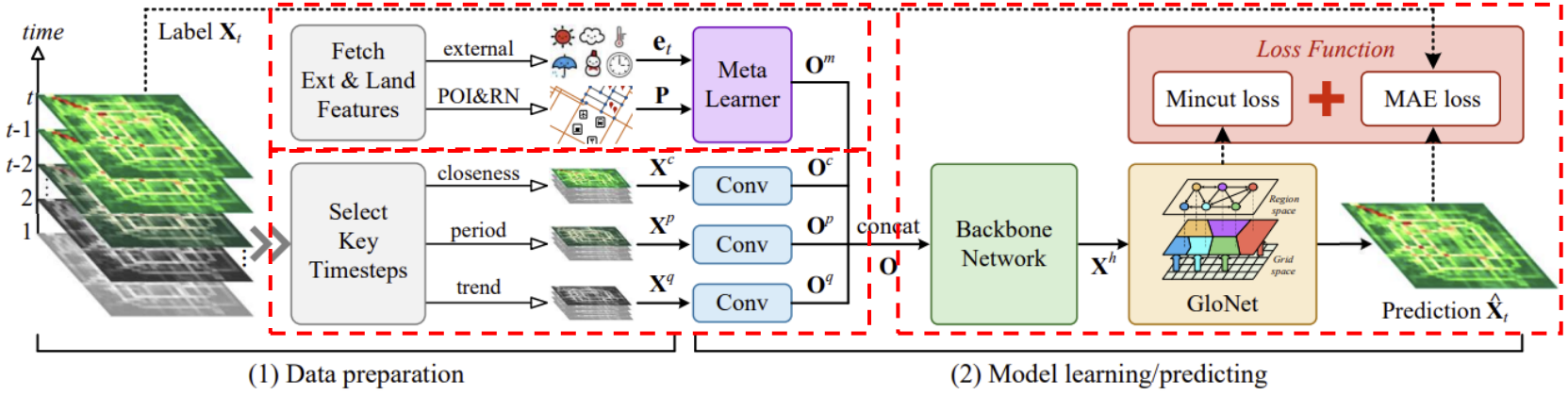
- Inefficiency to capture the global spatial dependencies
  - Stacking CNNs to increase receptive fields [Zhang et al. 2017]
  - Learning long-range spatial dependencies directly [Lin et al. 2019]





Spatio-Temporal Relation Network (STRN)

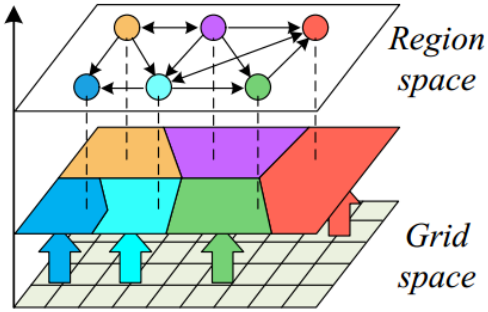
- Spatio-Temporal Relation Network (STRN)
  - Modeling temporal properties: closeness, periodicity, trend
  - Learning the impact of external factors
  - Learning local and global spatial dependencies

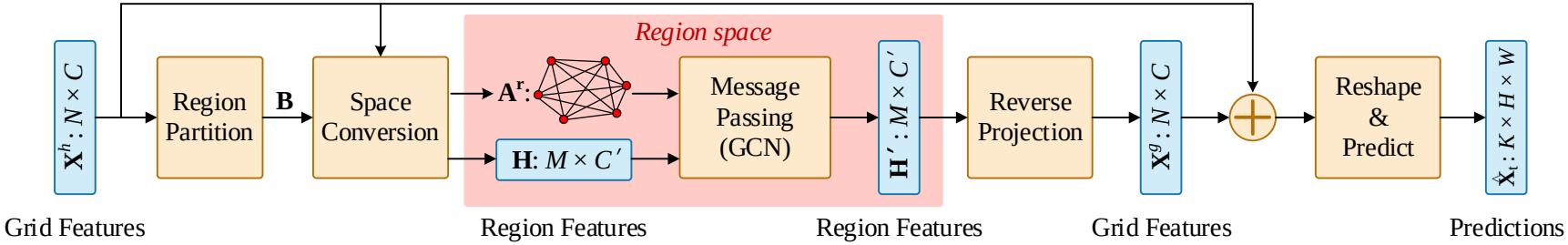


Y. Liang, R. Zimmermann et al., Fine-grained Urban Flow Prediction. WWW 2021.

Global Relation Module (GloNet)

- Insight: relational inference on a higher semantic level
- GloNet is composed of four major steps
  - Region partition
  - Space conversion
  - Message passing between regions
  - Reverse projection & Predict





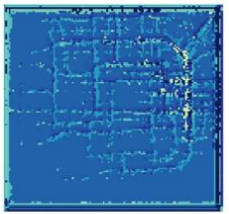
Global Relation Module (GloNet)

- Four major steps
  - Region partition: how to generate the assignment matrix $\mathbf{B} \in \mathbb{R}^{N \times M}$
  - Space conversion
  - Message passing between regions
  - Reverse projection & Predict

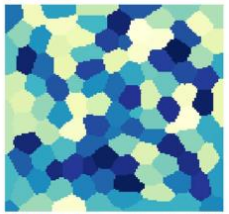
$$\mathbf{B} = \text{softmax}\left(\delta(\mathbf{X}^h)\right)$$

Mincut Loss

$$\mathcal{L}_m = \underbrace{-\frac{\text{Tr}(\mathbf{B}^T \tilde{\mathbf{A}}^g \mathbf{B})}{\text{Tr}(\mathbf{B}^T \tilde{\mathbf{D}}^g \mathbf{B})}}_{\mathcal{L}_c} + \underbrace{\left\| \frac{\mathbf{B}^T \mathbf{B}}{\|\mathbf{B}^T \mathbf{B}\|_F} - \frac{\mathbf{I}_M}{\sqrt{M}} \right\|_F}_{\mathcal{L}_o},$$



(a) Results at epoch 1 ($M = 100$)



(b) Results at epoch 150 ($M = 100$)

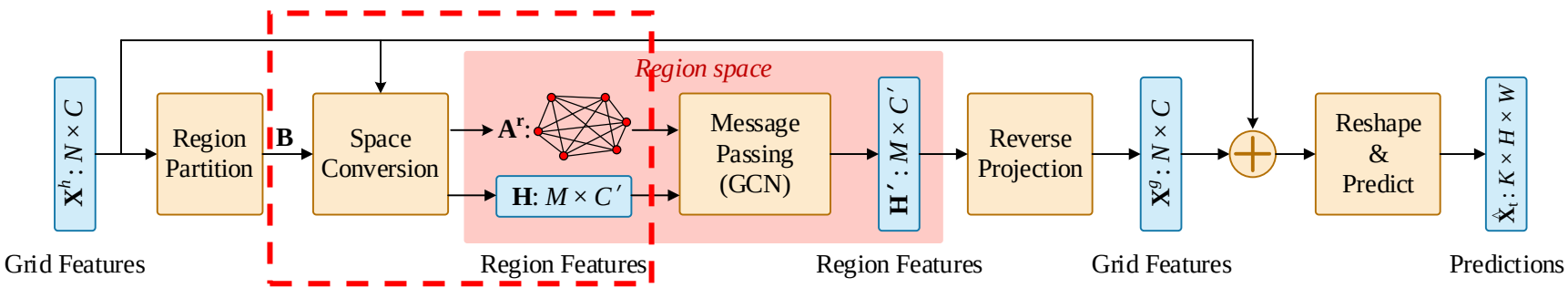
Global Relation Module (GloNet)

- Four major steps
  - Region partition
  - Space conversion
  - Message passing between regions
  - Reverse projection & Predict

$$\mathcal{L}_m = \underbrace{-\frac{\mathrm{Tr}(\mathbf{B}^T\tilde{\mathbf{A}}^g\mathbf{B})}{\mathrm{Tr}(\mathbf{B}^T\tilde{\mathbf{D}}^g\mathbf{B})}}_{\mathcal{L}_c} + \underbrace{\left\| \frac{\mathbf{B}^T\mathbf{B}}{\|\mathbf{B}^T\mathbf{B}\|_F} - \frac{\mathbf{I}_M}{\sqrt{M}} \right\|_F}_{\mathcal{L}_o},$$

We transform grid embeddings to their regional counterparts that are more friendly to capture global relations

$$\mathbf{H} = \mathbf{B}^\top \phi(\mathbf{X}^h), \quad \mathbf{A}^r = \mathbf{B}^\top \tilde{\mathbf{A}}^g \mathbf{B},$$



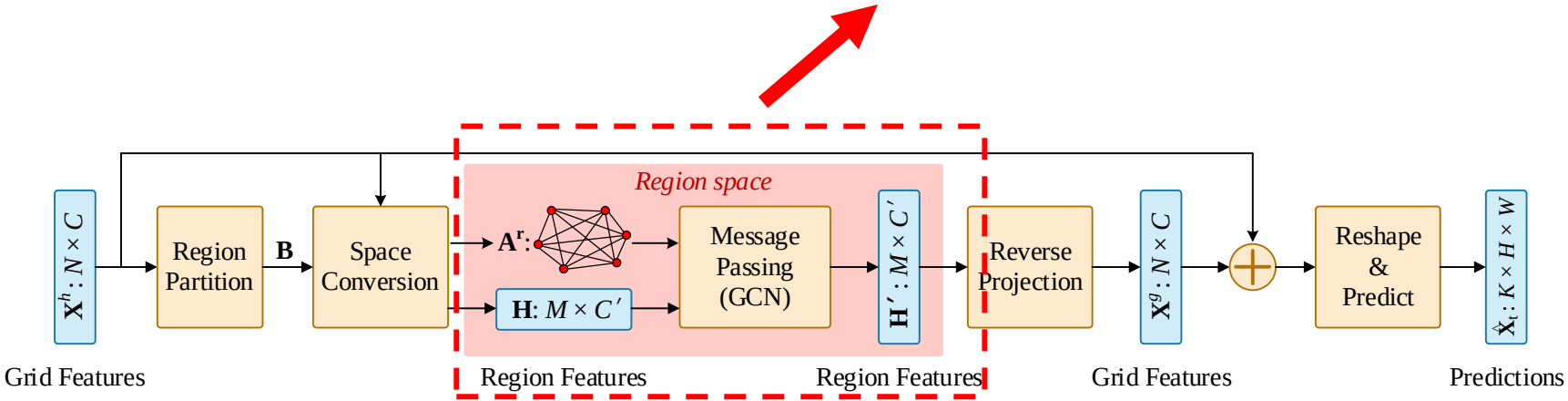
Global Relation Module (GloNet)

- Four major steps
  - Region partition
  - Space conversion
  - Message passing between regions
  - Reverse projection & Predict

Modeling the inter-region relations with Graph convolutional networks (GCNs)

$$\hat{\mathbf{A}}^r = \mathbf{A}^r - \mathrm{diag}\left(\mathbf{A}^r\right); \quad \tilde{\mathbf{A}}^r = \hat{\mathbf{D}}^{-\frac{1}{2}}\hat{\mathbf{A}}^r\hat{\mathbf{D}}^{-\frac{1}{2}}$$

$$\mathbf{H}' = f_{GCN}(\tilde{\mathbf{A}}^r, \mathbf{H}) = \tilde{\mathbf{A}}^r \,\mathrm{ReLU}\left(\tilde{\mathbf{A}}^r\mathbf{H}\mathbf{W}_1\right)\mathbf{W}_2$$

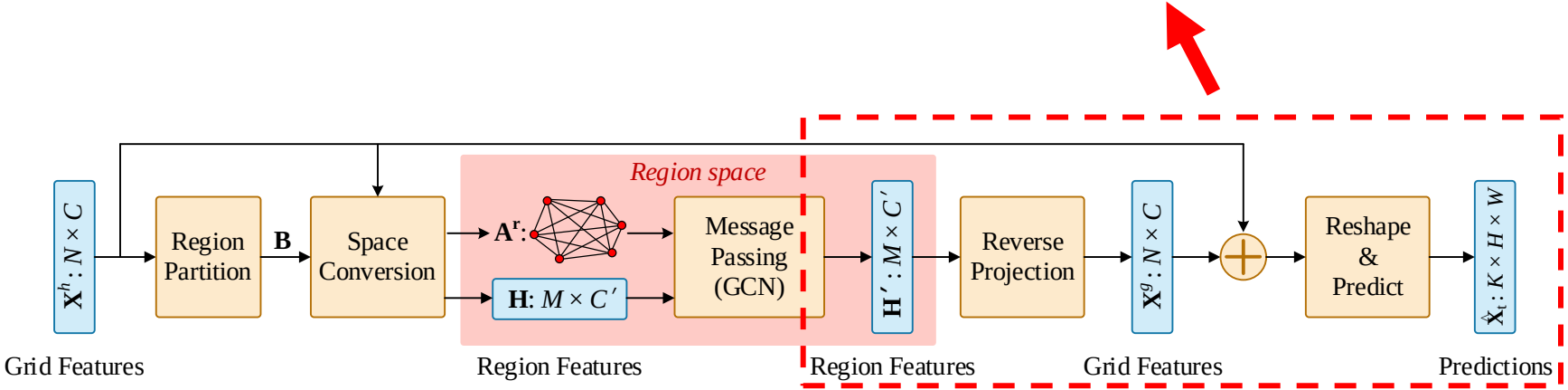


Global Relation Module (GloNet)

- Four major steps
  - Region partition
  - Space conversion
  - Message passing between regions
  - Reverse projection & Predict

Project back to the grid space and make final predictions

$$\mathbf{X}^g = \mathbf{B}\theta(\mathbf{H}')$$



Datasets

- We use the previous 12 steps to predict the next step

Dataset	TaxiBJ+	HappyValley
Data type	Taxi trip	Human flow
Resolution	128×128	50×100
Size of a grid	150m×150m	10m×10m
# Channels (K)	2 (inflow and outflow)	1 (staying flow)
Sampling rate	30 minutes	1 hour
Time span	P1: 07/01/2013-10/31/2013 P2: 02/01/2014-06/30/2014 P3: 03/01/2015-06/30/2015 P4: 11/01/2015-03/31/2016	01/01/2018- 10/31/2018
External factors (meteorology, time and event)		
Weather (*e.g.*, rainy)	16 types	8 types
Temperature/°C	[-24.6,41.0]	[-24.6,41.0]
Wind speed/mph	[0,48.6]	[0,48.6]
# holidays	41	33
Ticket prize/RMB	/	[29.9,260]
Land features (POIs, road networks)		
# of POIs	651,016 (20 types)	/
# features of RNs	5	/

Results on TaxiBJ+

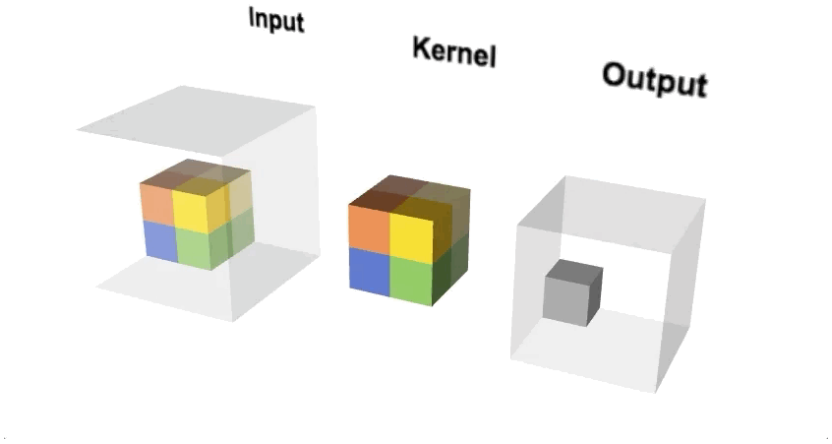
- STRN outperforms the SOTA method by average 8.1% to 11.5% in terms of MAE over the four time periods while using much fewer parameters

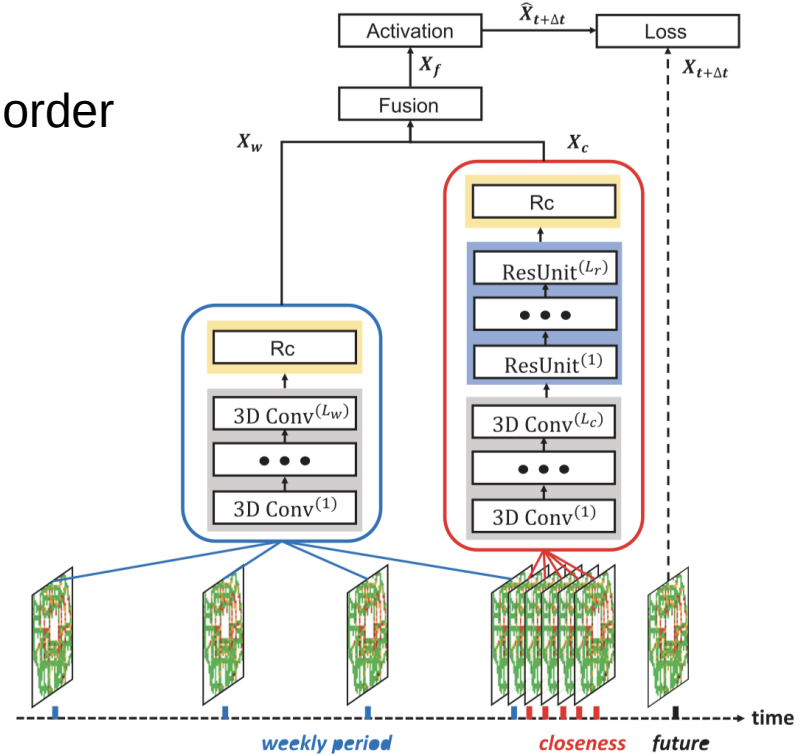
Table 2: Model comparison on TaxiBJ+, where the notation Δ indicates the reduction of MAE compared with DeepSTN+.

Method	#Param	P1			P2			P3			P4		
		MAE	Δ	RMSE	MAE	Δ	RMSE	MAE	Δ	RMSE	MAE	Δ	RMSE
ARIMA	<0.01M	2.46	+24.2%	5.37	2.91	+28.8%	6.34	3.02	+26.4%	6.55	2.08	+19.5%	4.47
VAR	<0.01M	2.41	+21.7%	5.21	2.84	+25.7%	6.18	2.92	+22.2%	6.38	2.06	+18.4%	4.35
LSTM	0.07M	2.27	+14.6%	5.04	2.68	+18.6%	6.03	2.78	+16.3%	6.21	1.88	+8.0%	4.20
ConvLSTM	3.45M	2.03	+2.5%	4.47	2.33	+3.1%	5.15	2.45	+2.5%	5.43	1.76	+1.1%	3.94
DeepST	0.46M	2.21	+11.6%	4.68	2.53	+11.9%	5.41	2.57	+7.5%	5.59	1.92	+10.3%	4.05
ST-ResNet	2.39M	2.14	+8.1%	4.58	2.48	+9.7%	5.29	2.61	+9.2%	5.55	1.83	+5.2%	3.88
ST-3DNet	0.89M	2.16	+9.1%	4.56	2.30	+1.8%	4.99	2.38	-0.4%	5.23	1.95	+12.1%	4.20
STDN	6.36M	2.08	+5.1%	4.40	2.32	+2.7%	4.98	2.44	+2.1%	5.23	1.85	+6.3%	3.85
DeepSTN+	0.27G	1.98	-	4.24	2.26	-	4.87	2.39	-	5.15	1.74	-	3.75
STRN	0.88M	**1.82**	**-8.1%**	**4.13**	**2.10**	**-7.1%**	**4.71**	**2.19**	**-8.4%**	**5.01**	**1.54**	**-11.5%**	**3.61**

Conv3D-based Solution

- ST-3DNet
  - 2D convolution cannot capture temporal order





S. Guo et al., Deep Spatial–Temporal 3D Convolutional Neural Networks for Traffic Data Forecasting. TITS 2019.

Results on TaxiBJ+

- STRN outperforms the SOTA method by average 8.1% to 11.5% in terms of MAE over the four time periods while using much fewer parameters

Table 2: Model comparison on TaxiBJ+, where the notation Δ indicates the reduction of MAE compared with DeepSTN+.

Method	#Param	P1			P2			P3			P4		
		MAE	Δ	RMSE	MAE	Δ	RMSE	MAE	Δ	RMSE	MAE	Δ	RMSE
ARIMA	<0.01M	2.46	+24.2%	5.37	2.91	+28.8%	6.34	3.02	+26.4%	6.55	2.08	+19.5%	4.47
VAR	<0.01M	2.41	+21.7%	5.21	2.84	+25.7%	6.18	2.92	+22.2%	6.38	2.06	+18.4%	4.35
LSTM	0.07M	2.27	+14.6%	5.04	2.68	+18.6%	6.03	2.78	+16.3%	6.21	1.88	+8.0%	4.20
ConvLSTM	3.45M	2.03	+2.5%	4.47	2.33	+3.1%	5.15	2.45	+2.5%	5.43	1.76	+1.1%	3.94
DeepST	0.46M	2.21	+11.6%	4.68	2.53	+11.9%	5.41	2.57	+7.5%	5.59	1.92	+10.3%	4.05
ST-ResNet	2.39M	2.14	+8.1%	4.58	2.48	+9.7%	5.29	2.61	+9.2%	5.55	1.83	+5.2%	3.88
ST-3DNet	0.89M	2.16	+9.1%	4.56	2.30	+1.8%	4.99	2.38	-0.4%	5.23	1.95	+12.1%	4.20
STDN	6.36M	2.08	+5.1%	4.40	2.32	+2.7%	4.98	2.44	+2.1%	5.23	1.85	+6.3%	3.85
DeepSTN+	0.27G	1.98	-	4.24	2.26	-	4.87	2.39	-	5.15	1.74	-	3.75
STRN	0.88M	**1.82**	**-8.1%**	**4.13**	**2.10**	**-7.1%**	**4.71**	**2.19**	**-8.4%**	**5.01**	**1.54**	**-11.5%**	**3.61**

Results on TaxiBJ+

- Effects of GloNet
  - Effectiveness

Variant	#Param	MAE	Δ	RMSE
Backbone	0.82M	1.97	-	4.22
STRN w/o GloNet	0.83M	1.93	-2.0%	4.20
STRN w/o ML	0.87M	1.85	-6.1%	4.15
STRN w/o dynamic	0.88M	1.87	-5.1%	4.16
STRN	0.88M	**1.82**	**-7.6%**	**4.13**

Results on TaxiBJ+

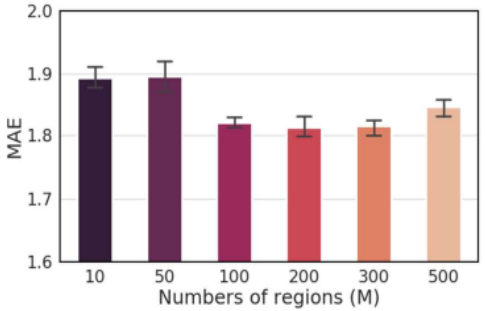
- Effects of GloNet
  - Effectiveness
  - Static vs. Dynamic

Variant	#Param	MAE	Δ	RMSE
Backbone	0.82M	1.97	-	4.22
STRN w/o GloNet	0.83M	1.93	-2.0%	4.20
STRN w/o ML	0.87M	1.85	-6.1%	4.15
STRN w/o dynamic	0.88M	1.87	-5.1%	4.16
STRN	0.88M	**1.82**	**-7.6%**	**4.13**

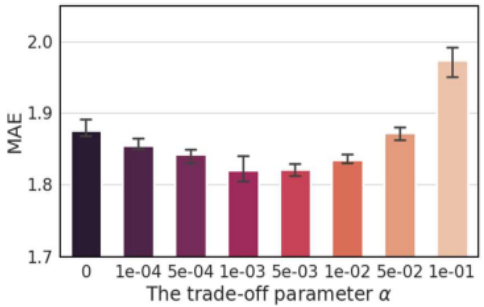
Results on TaxiBJ+

- Effects of GloNet
  - Effectiveness
  - Static vs. Dynamic region partition
  - Hyperparameters
    - Number of regions M
    - Trade-off parameter α

Variant	#Param	MAE	Δ	RMSE
Backbone	0.82M	1.97	-	4.22
STRN w/o GloNet	0.83M	1.93	-2.0%	4.20
STRN w/o ML	0.87M	1.85	-6.1%	4.15
STRN w/o dynamic	0.88M	1.87	-5.1%	4.16
STRN	0.88M	**1.82**	**-7.6%**	**4.13**



(c) Number of regions vs. MAE



(d) Trade-off parameter vs. MAE

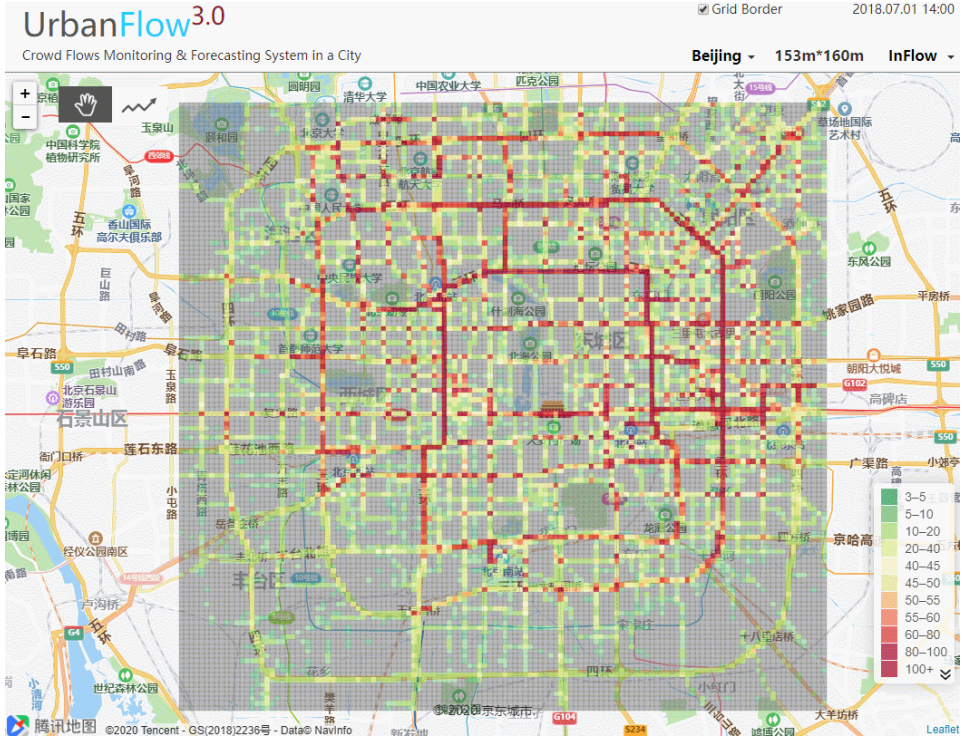
Results on HappyValley

- STRN also achieves SOTA performance on the HappyValley dataset

Model	#Param	MAE	Δ	RMSE
DeepST	0.26M	2.21	+10.6%	7.98
ST-ResNet	0.63M	2.20	+10.1%	7.91
ST-3DNet	0.52M	2.16	+8.1%	7.95
ConvLSTM	0.63M	1.98	-1.0%	7.86
STDN	0.97M	2.05	+2.5%	7.89
DeepSTN+	15.69M	2.00	-	7.88
STRN w/o GloNet	0.36M	1.97	-1.5%	7.90
STRN	0.37M	**1.85**	**-7.6%**	**7.80**

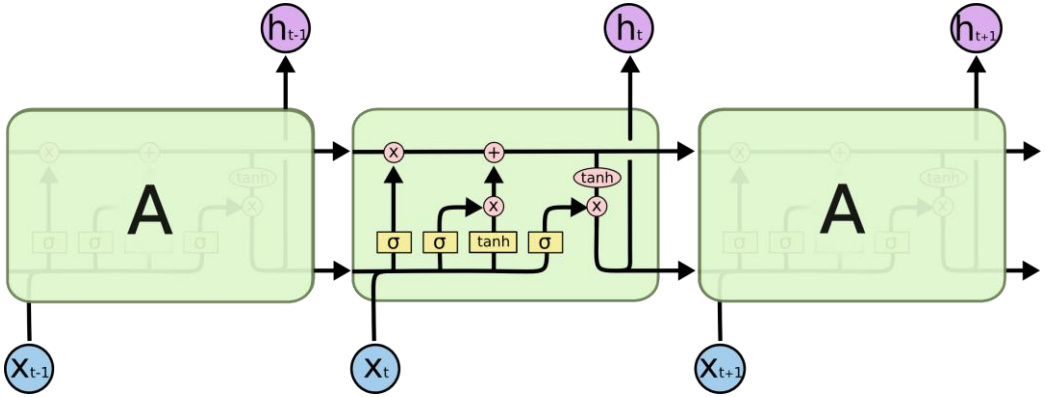
Conclusion

- Fine-grained urban flow prediction
- Methodology
  - –Local feature extraction
  - –GloNet: Global relation modeling
- Evaluation
  - –SOTA performance
  - –Scalability & Lightweight property
- Real deployment



ConvLSTM-based Solution

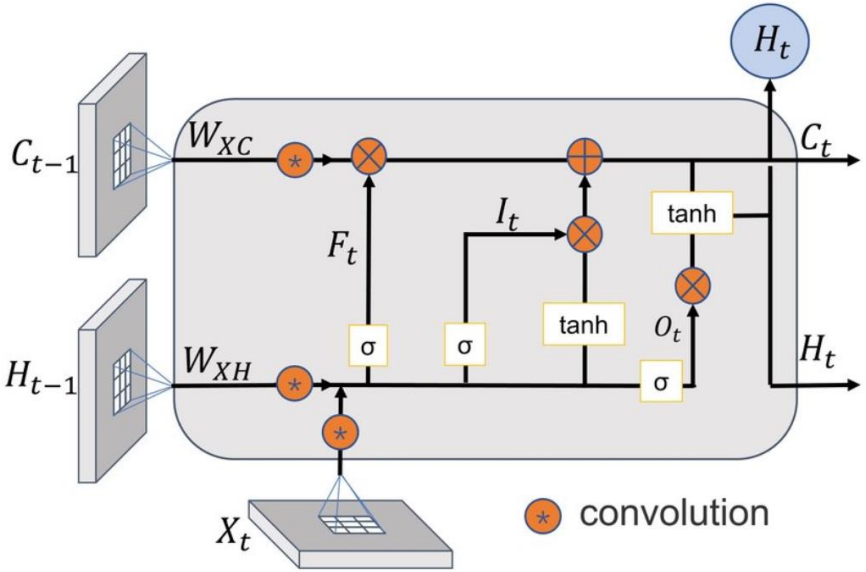
- What is LSTM (Long Short-Term Memory)?



$$
\begin{aligned}
i_t &= \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci} \circ c_{t-1} + b_i) \\
f_t &= \sigma(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf} \circ c_{t-1} + b_f) \\
c_t &= f_t \circ c_{t-1} + i_t \circ \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \\
o_t &= \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co} \circ c_t + b_o) \\
h_t &= o_t \circ \tanh(c_t)
\end{aligned}
$$

ConvLSTM-based Solution

- When Convolution meets LSTM



$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci} \circ c_{t-1} + b_i)$$
$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf} \circ c_{t-1} + b_f)$$
$$c_t = f_t \circ c_{t-1} + i_t \circ \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c)$$
$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co} \circ c_t + b_o)$$
$$h_t = o_t \circ \tanh(c_t)$$

Vanilla LSTM

$$i_t = \sigma(W_{xi} * \mathcal{X}_t + W_{hi} * \mathcal{H}_{t-1} + W_{ci} \circ \mathcal{C}_{t-1} + b_i)$$
$$f_t = \sigma(W_{xf} * \mathcal{X}_t + W_{hf} * \mathcal{H}_{t-1} + W_{cf} \circ \mathcal{C}_{t-1} + b_f)$$
$$\mathcal{C}_t = f_t \circ \mathcal{C}_{t-1} + i_t \circ \tanh(W_{xc} * \mathcal{X}_t + W_{hc} * \mathcal{H}_{t-1} + b_c)$$
$$o_t = \sigma(W_{xo} * \mathcal{X}_t + W_{ho} * \mathcal{H}_{t-1} + W_{co} \circ \mathcal{C}_t + b_o)$$
$$\mathcal{H}_t = o_t \circ \tanh(\mathcal{C}_t)$$

ConvLSTM

X. Shi et al., Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting. NIPS 2015.

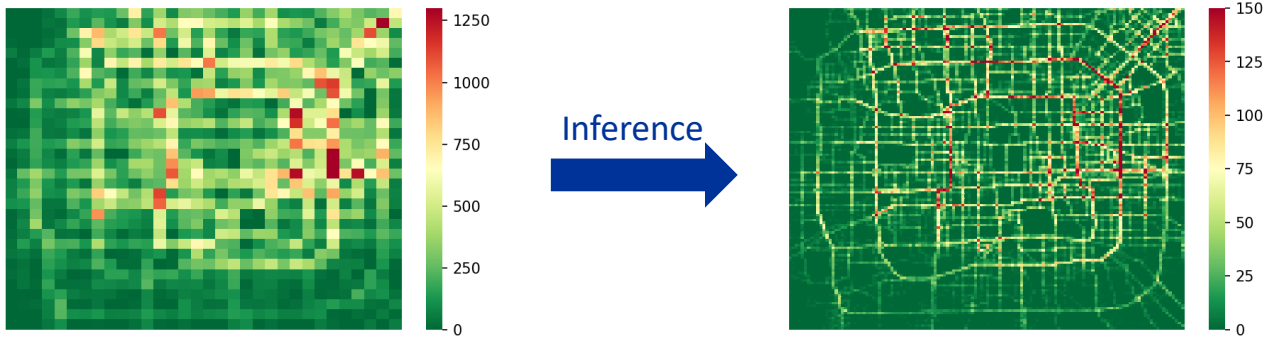
Application: Fine-Grained Urban Flow Inference

- Fine-grained urban data acquisition needs large-scale deployment of sensors, which is time-consuming and produces high costs
- To save costs, can we use coarse-grained data to generate fine-grained data?

Y. Liang, D. S. Rosenblum, Y. Zheng et al., UrbanFM: Inferring Fine-grained Urban Flows. KDD 2019.

Problem 1: Fine-Grained Urban Flow Inference

- We originally formalize the FUFI problem
  - Given an upscaling factor N and a coarse-grained flow map $\mathbf{X}^c \in \mathbb{R}_+^{I \times J}$, we aim to infer the fine-grained counterpart $\mathbf{X}^f \in \mathbb{R}_+^{NI \times NJ}$

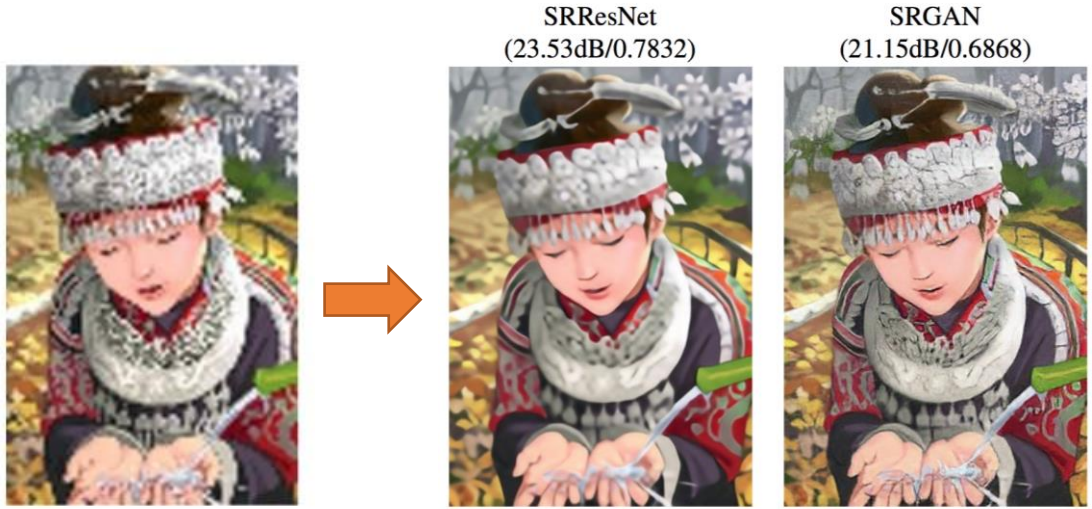


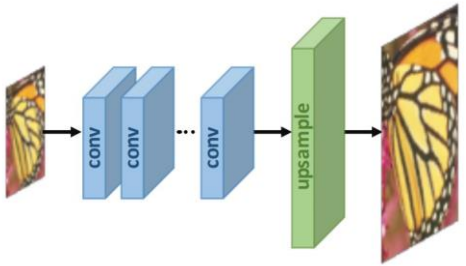
Coarse-grained crowd flows
(32 x 32)

Fine-grained crowd flows
(128 x 128)

Related Work

- Image super-resolution (SR)
  - Using CNNs to upscale and improve the quality of low-resolution images

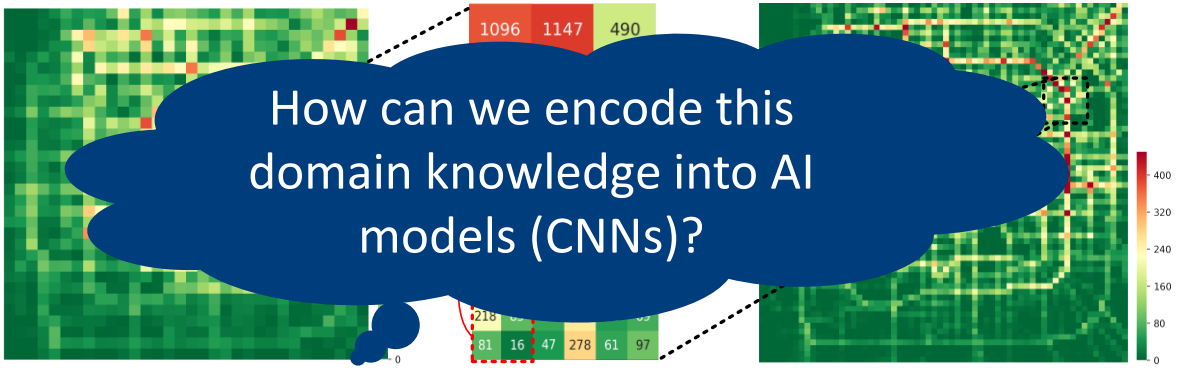




Ledig et al., Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network. CVPR 2017.

Challenges & Difference to Image SR

- Domain knowledge: **Spatial hierarchy**
  - This implies a crucial structural constraint: the sum of the flow volumes in sub-regions strictly equals that of the corresponding super-region



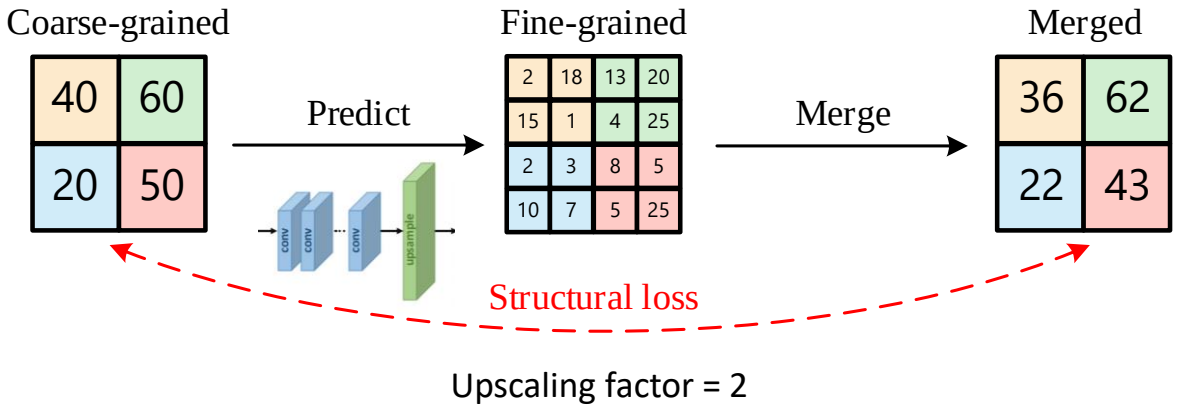
(a) Coarse-grained crowd flows (32x32)

(b) Fine-grained crowd flows (64x64)

An Intuitive Idea

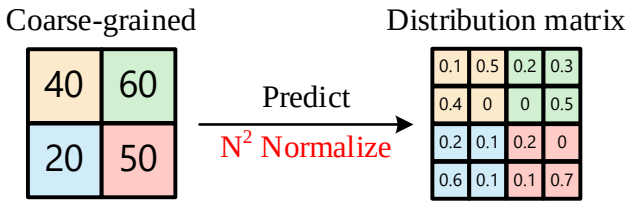
- Using structural loss to constrain the spatial hierarchy
  - A regularization term
  - Two drawbacks
    - ➢ It cannot exactly guarantee the spatial hierarchy
    - ➢ It cannot reduce the inference errors in our empirical study

$$L_s = \sum_{i,j} \left\| x^c_{i,j} - \sum_{i',j'} \tilde{x}^f_{i',j'} \right\|_F \quad s.t. \lfloor \frac{i'}{N} \rfloor = i, \lfloor \frac{j'}{N} \rfloor = j.$$



Our Method: Distributional Upsampling

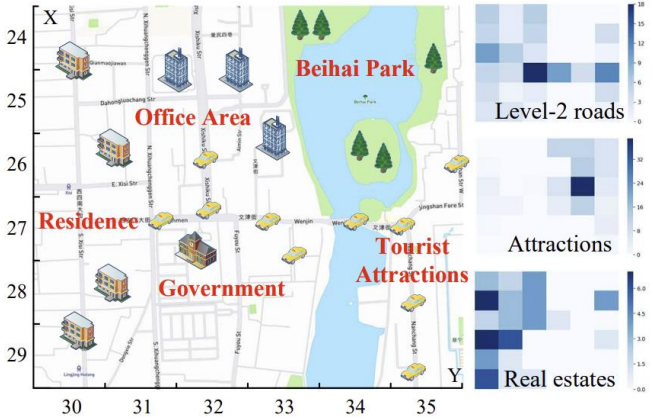
- Main idea: **changing the prediction target to a distribution matrix**
  - N^2-Normalization: performing normalization in each N^2 areas
    - Guarantee the spatial hierarchy between coarse-and fine-grained data
    - Efficient: no learnable parameters, easy to implement



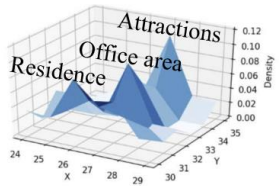
Assigning how the flows in a region will be assigned to the subregions.

Challenges & Difference to Image SR

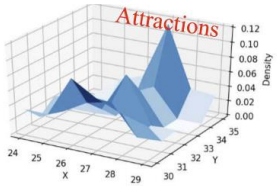
- **External factors**
  - Time of day
  - Weather condition
  - Events



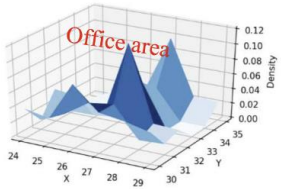
(a) A core area of Beijing and heatmaps of several geospatial attributes



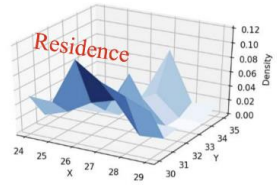
(b) 10 am on weekdays



(c) 10 am on weekends



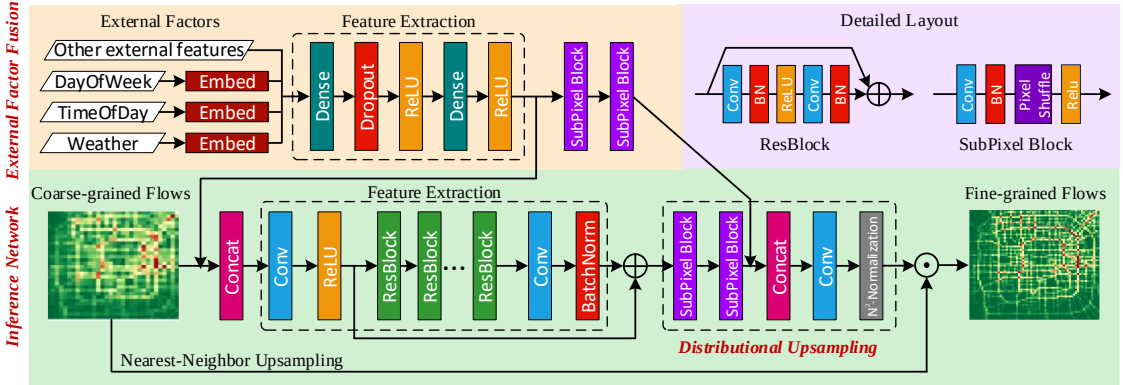
(d) 10 am on weekdays with thunderstorm



(e) 8 pm on weekdays

Solution: Urban Flow Magnifier (UrbanFM)

- By jointly considering the spatial hierarchy and external factors, we present UrbanFM to resolve the FUFI problem
  - Flow inference network
  - External factor fusion
  - Distributional upsampling



Y. Liang, D. S. Rosenblum, Y. Zheng et al., UrbanFM: Inferring Fine-grained Urban Flows. KDD 2019.

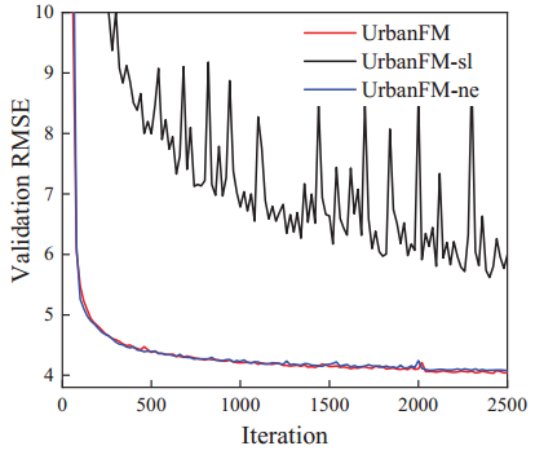
Results on TaxiBJ

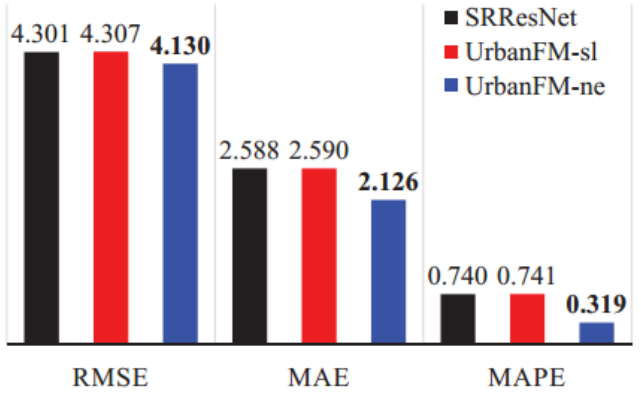
- Dataset: TaxiBJ with 4 time periods
- Evaluation metrics: RMSE, MAE, MAPE (lower is better)

Methods	P1			P2			P3			P4		
	RMSE	MAE	MAPE	RMSE	MAE	MAPE	RMSE	MAE	MAPE	RMSE	MAE	MAPE
MEAN	20.918	12.019	4.469	20.918	12.019	5.364	27.442	16.029	5.612	19.049	11.070	4.192
HA	4.741	2.251	0.336	5.381	2.551	0.334	5.594	2.674	0.328	4.125	2.023	0.323
SRCNN	4.297	2.491	0.714	4.612	2.681	0.689	4.815	2.829	0.727	3.838	2.289	0.665
ESPCN	4.206	2.497	0.732	4.569	2.727	0.732	4.744	2.862	0.773	3.728	2.228	0.711
DeepSD	4.156	2.368	0.614	4.554	2.612	0.621	4.692	2.739	0.682	3.877	2.297	0.652
VDSR	4.159	2.213	0.467	4.586	2.498	0.486	4.730	2.548	0.461	3.654	1.978	0.411
SRResNet	4.164	2.457	0.713	4.524	2.660	0.688	4.690	2.775	0.717	3.667	2.189	0.637
UrbanFM-ne	4.015	2.047	0.332	4.386	2.258	0.320	4.559	2.352	0.316	3.559	1.845	0.309
UrbanFM	**3.950**	**2.011**	**0.327**	**4.329**	**2.224**	**0.313**	**4.496**	**2.318**	**0.315**	**3.501**	**1.815**	**0.308**

The Advantages of Changing Target

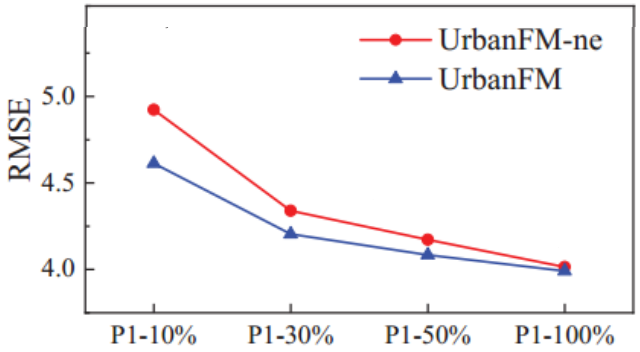
- Changing the prediction target to the distribution matrix
  - Significantly improving training efficiency: UrbanFM vs UrbanFM-sl (using structural loss)
  - Achieving much better performance, especially on MAE and MAPE

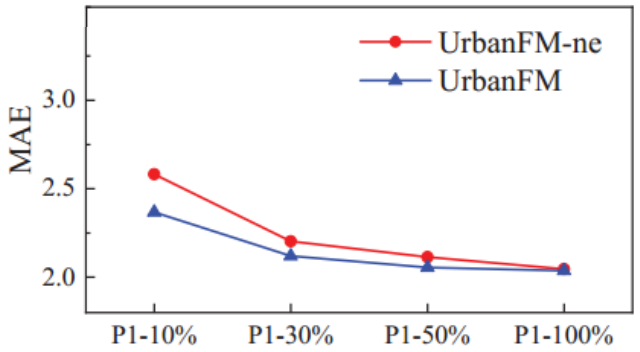




How do External Factors Contribute to FUFI

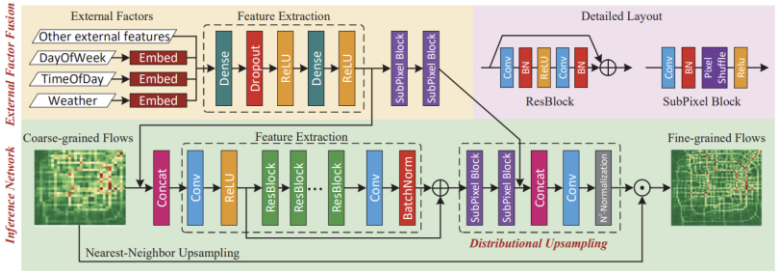
- UrbanFM vs. UrbanFM-ne (no external factors)
- When the training size grows, the improvement of using external factors decreases
  - External factors work as inductive bias (i.e., prior knowledge) to the model
  - If we do not have sufficient data samples, we can leverage such prior knowledge to benefit models

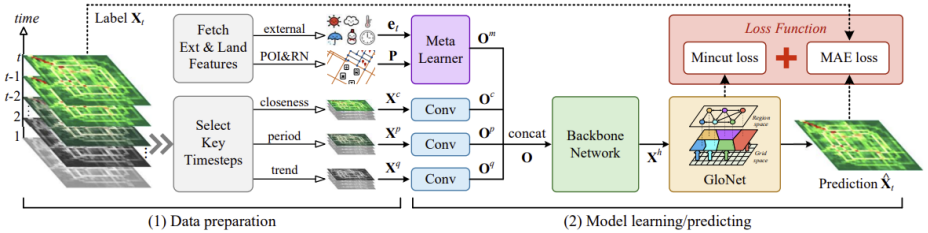




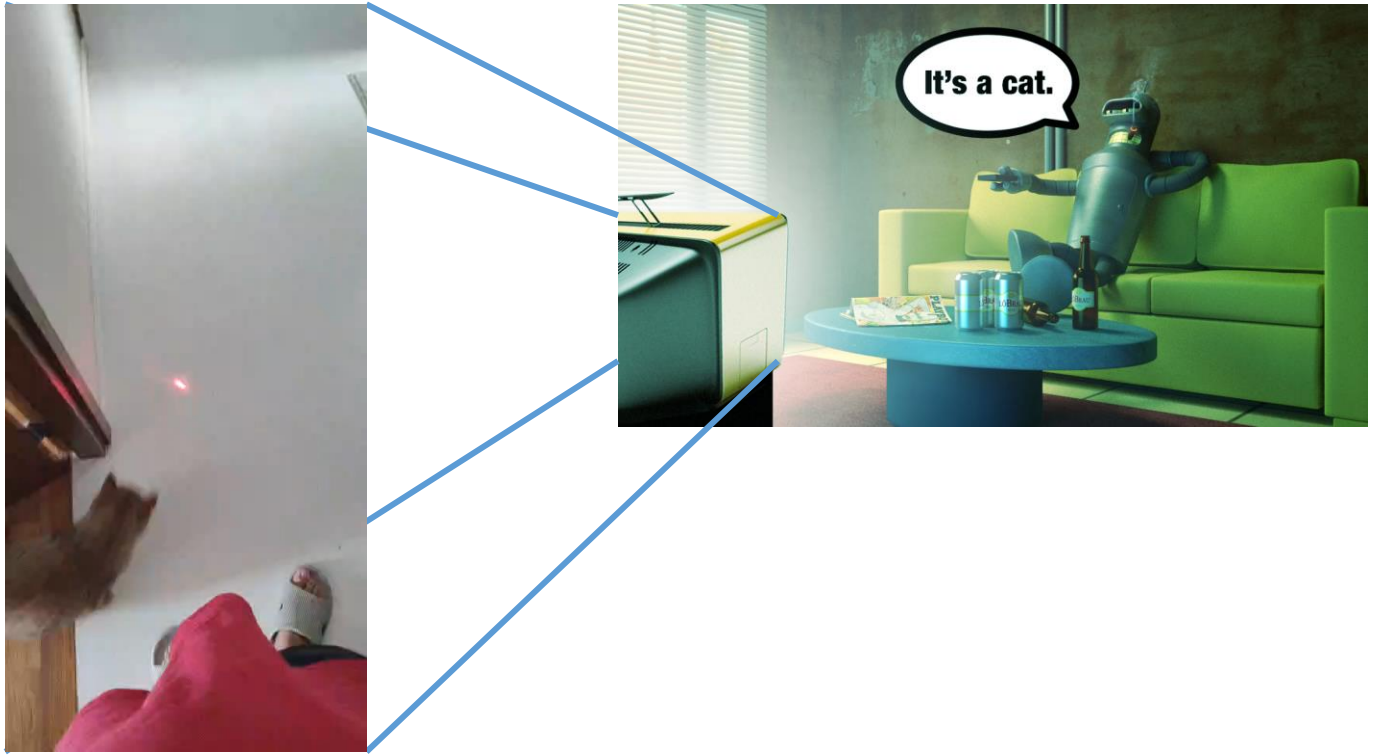
Conclusions on ST Grid Data

- When modeling ST grid data, leveraging appropriate domain knowledge can benefit the AI models in many aspects, e.g.,
  - Changing the learning target to guarantee the spatial hierarchy not only improves training efficiency, but also achieves better performance in FUFI
  - External factors work as prior knowledge to the neural networks
- **Our methods can easily generalize to other ST grid data**
  - Examples: covid-19 cases, urban crime, regional sales, bike-sharing demand





Application: Video Recognition

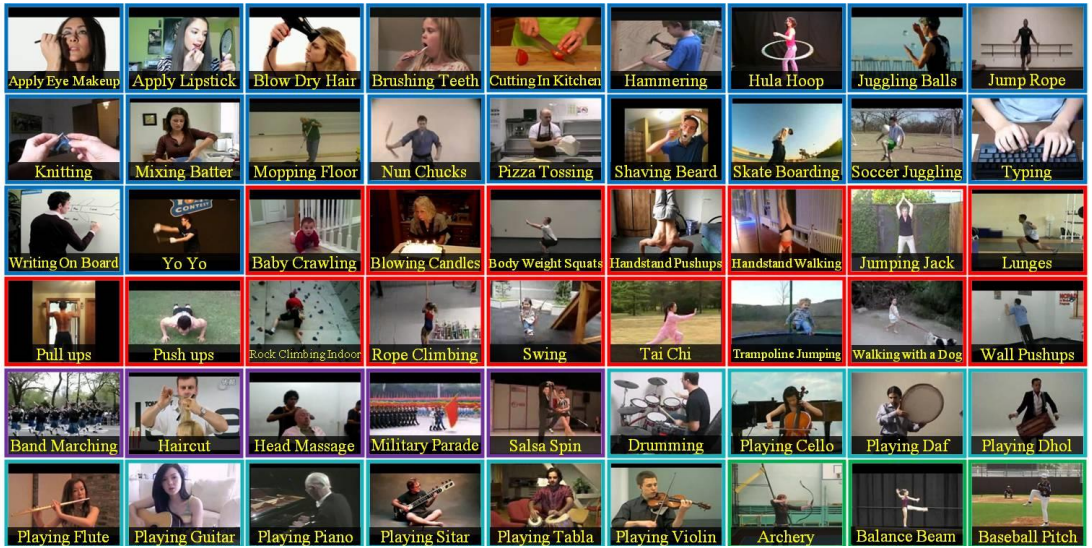


Datasets

- UCF-101
- HMDB-51
- Kinetics-400, Kinetics-600
- SSv2
- Epic Kitchens-100

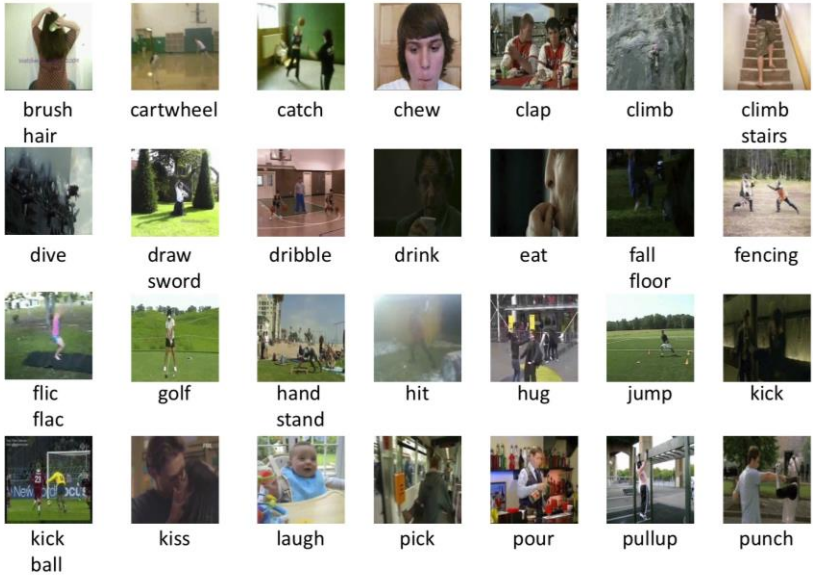
UCF-101

- Link
- It contains 13320 videos with an average length of 180 frames per video and 101 action categories.
- Used by VidTr



HMDB-51

- Link
- It has 6,766 videos and 51 action categories
- Used by VidTr



Kinetics-400 and Kinetics-600

- Introduction link
- Used by all Video ViTs

and 600, containing 400 and 600 classes respectively. As these are dynamic datasets (videos may be removed from YouTube), we note our dataset sizes are approximately 267 000 and 446 000 respectively.

	Year	Actions	Clips per class	Total
Kinetics-400	2017	400	400-1000	300k
Kinetics-600	2018	600	600-1000	500k

Something-Something-v2 (SSv2)

- Link
- Used by all Video ViTs

20BN-SOMETHING-SOMETHING-DATASET	
Total number of videos	220,847
Training Set	168,913
Validation Set	24,777
Test Set (w/o labels)	27,157
Labels	174

Epic Kitchens-100

- Link
- Used by ViViT and X-ViT

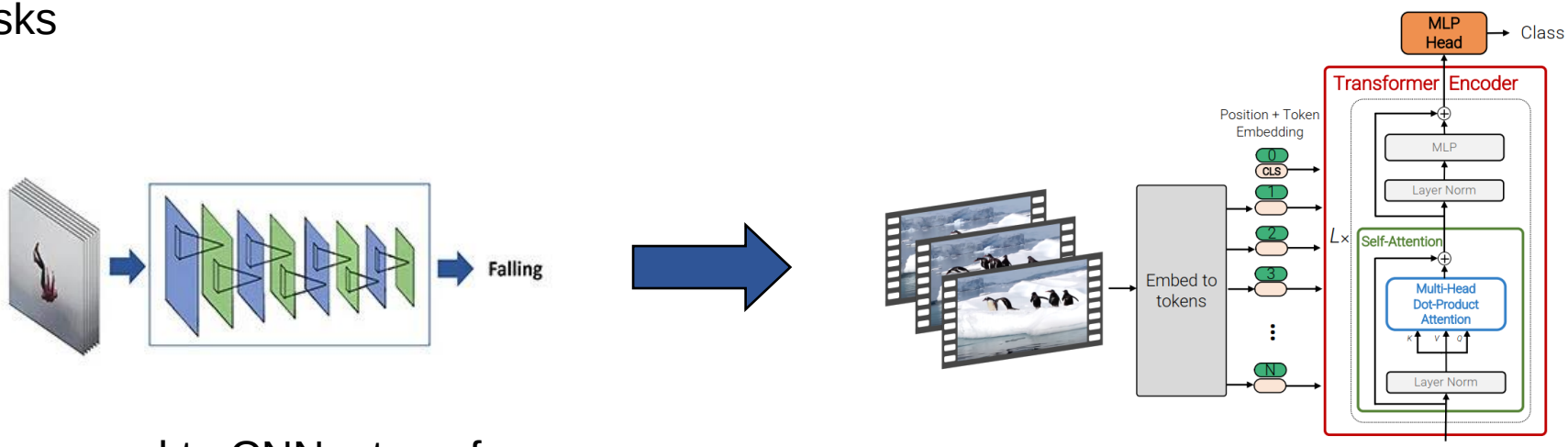
recognition" protocol. Here, each video is labelled with a "verb" and a "noun" and we therefore predict both categories using a single network with two "heads". The top-scoring verb and action pair predicted by the network form an "action", and action accuracy is the primary metric.

Characteristics

- 45 kitchens - 4 cities
- Head-mounted camera
- **100** hours of recording - Full HD
- 20M frames
- Multi-language narrations
- 90K action segments
- 20K unique narrations
- 97 verb classes, 300 noun classes
- 5 challenges

Transformer for Video Recognition

- We are witnessing a model shift from CNNs to Transformers on video recognition tasks

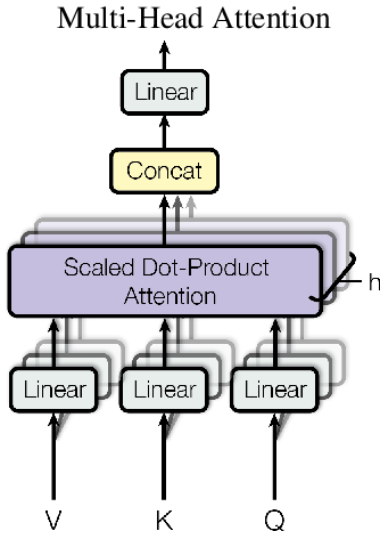


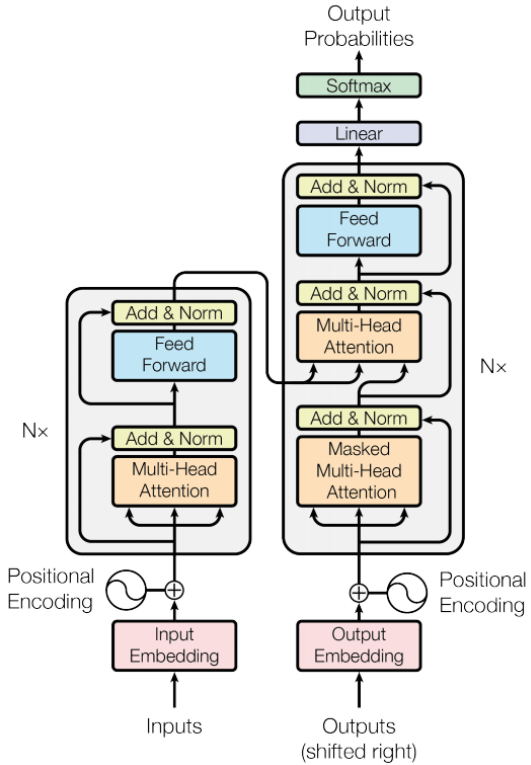
- Compared to CNNs, transformers are
  - Imposing less restrictive inductive bias
  - Friendly to capturing long-range spatio-temporal dependencies

Y. Liang et al., DualFormer: Local-Global Stratified Transformer for Efficient Video Recognition, ECCV 2022

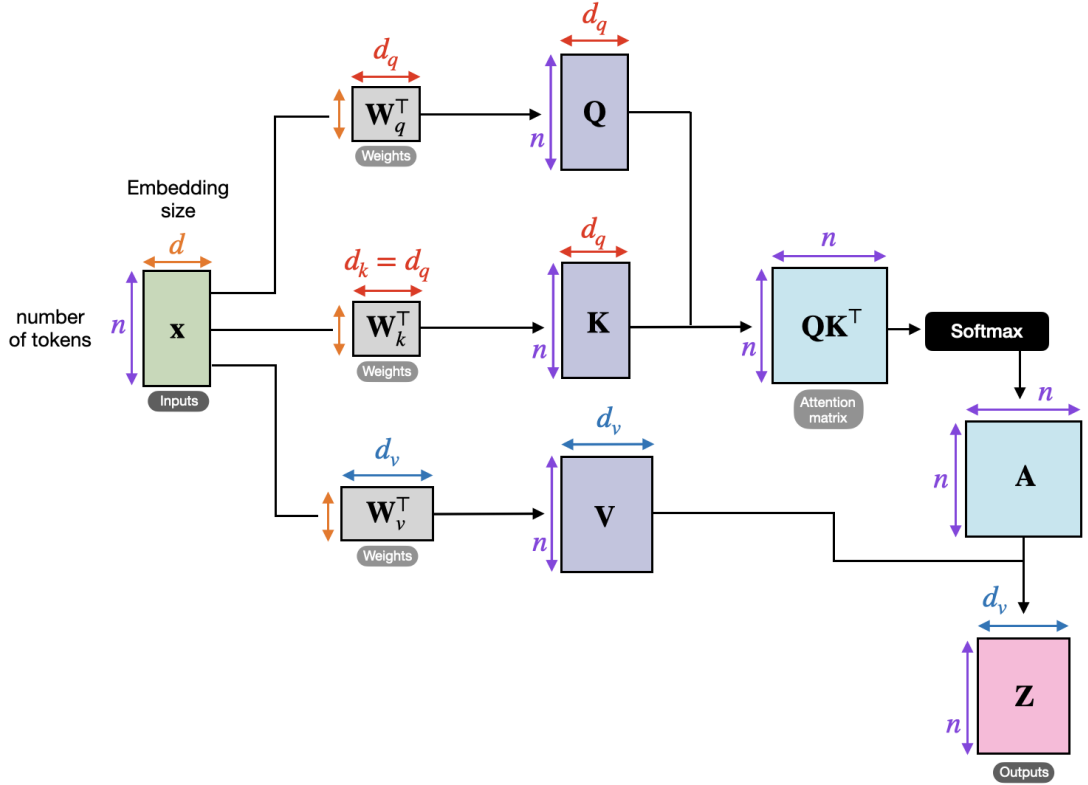
Preliminary

- Transformer
  - Key component: self-attention
  - Originally designed for NMT



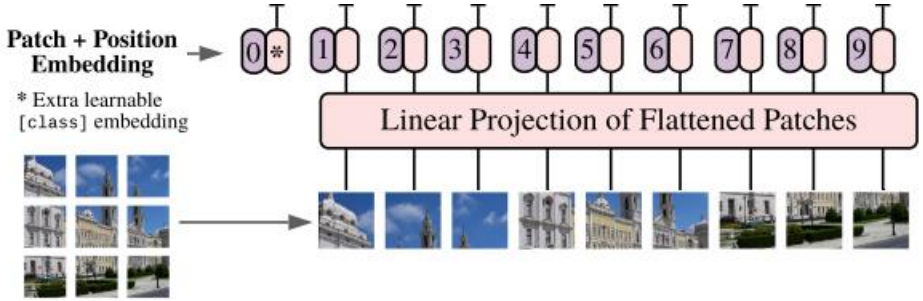


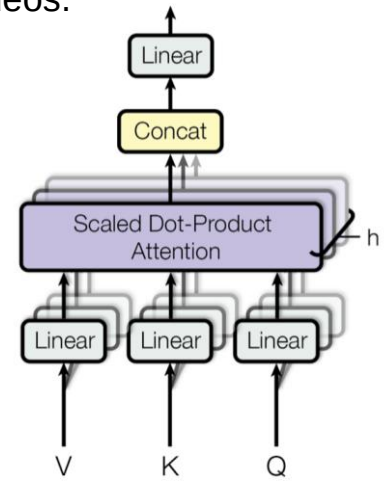
Self Attention Mechanism



Challenge

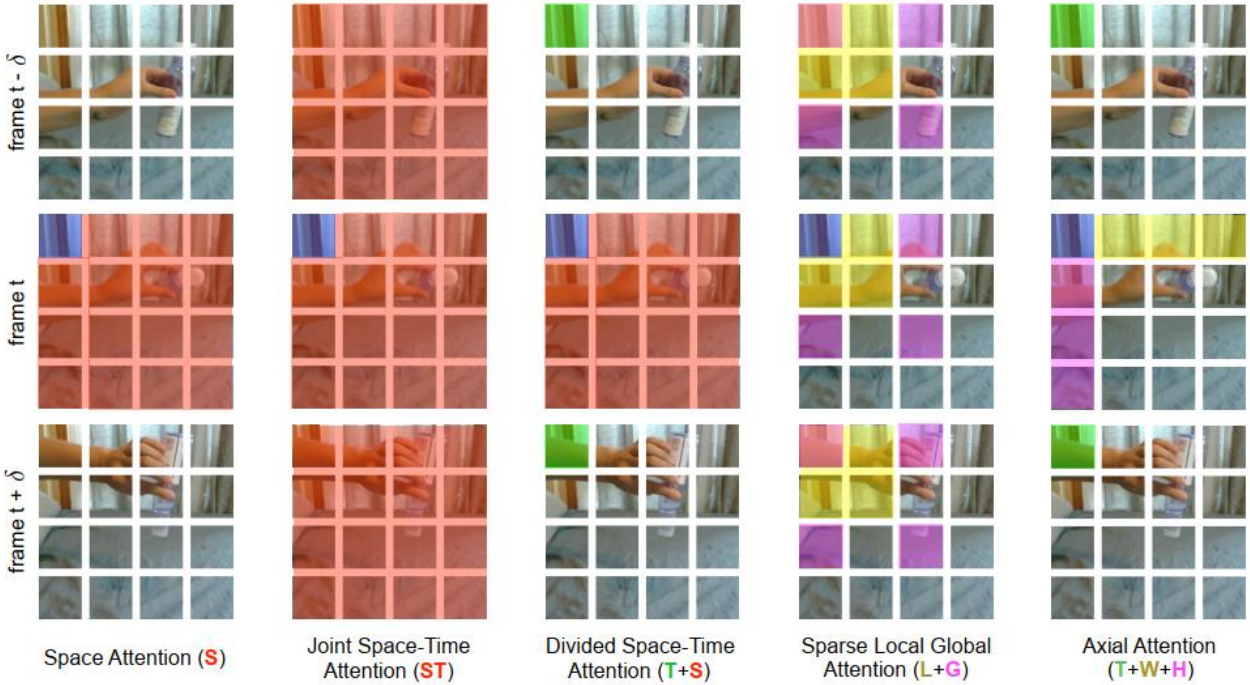
- One of the challenges when applying transformers to video data is their efficiency.
- Due to the Multihead Self-Attention (MSA) operation, the computational costs of video transformers grows quadratically with the increasing number of tokens
  - The number of tokens is usually much larger than images
  - May even become totally unaffordable for some high spatial resolution or long videos.





Existing Solutions - TimeSformer

- Towards efficient design: Space-time factorization



Bertasius et al., Is Space-Time Attention All You Need for Video Understanding, *ICML*, 2021

Existing Solutions - ViViT

- Towards efficient design: Space-time factorization

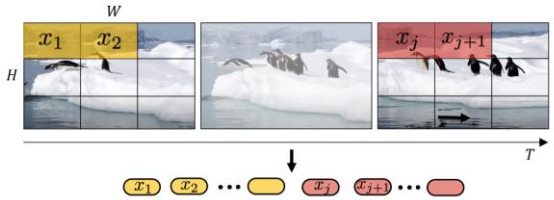


Figure 2: Uniform frame sampling: We simply sample n_t frames, and embed each 2D frame independently following ViT [15].

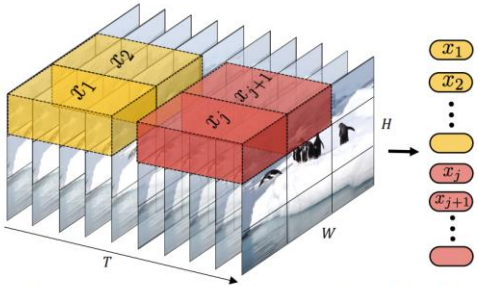


Figure 3: Tubelet embedding. We extract and linearly embed non-overlapping tubelets that span the spatio-temporal input volume.

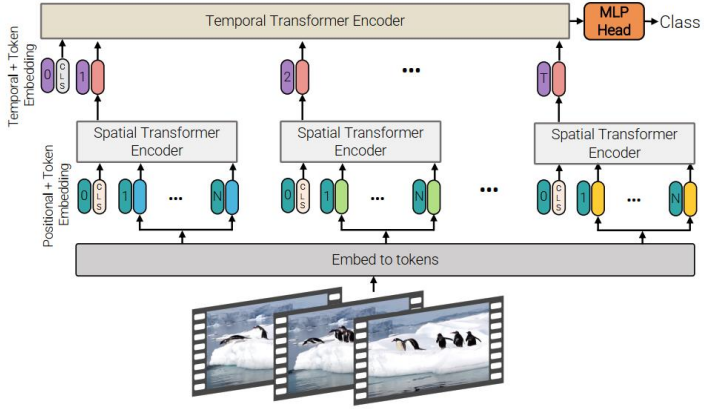
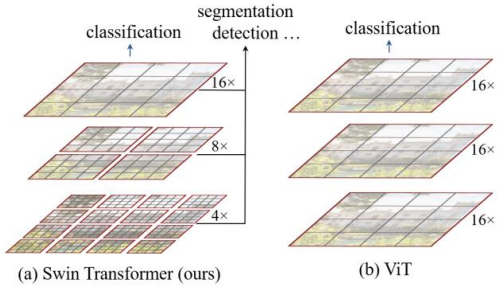


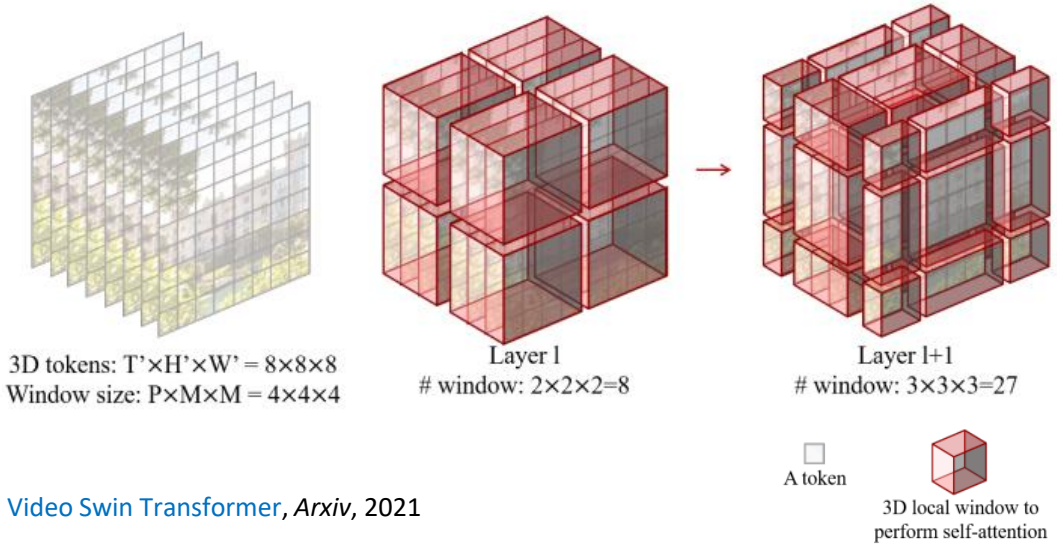
Figure 4: Factorised encoder (Model 2). This model consists of two transformer encoders in series: the first models interactions between tokens extracted from the same temporal index to produce a latent representation per time-index. The second transformer models interactions between time steps. It thus corresponds to a "late fusion" of spatial- and temporal information.

Arnab et al., ViViT: A Video Vision Transformer, *Arxiv*, 2021

Existing Solutions - Swin

- SOTA method - Video Swin Transformers
  - By extending a 2D Swin Transformer to a 3D version
  - Introducing inductive bias to Transformers
    - Locality
    - Hierarchy
    - Translation invariance
  - Advantage
    - More efficient
    - More accurate

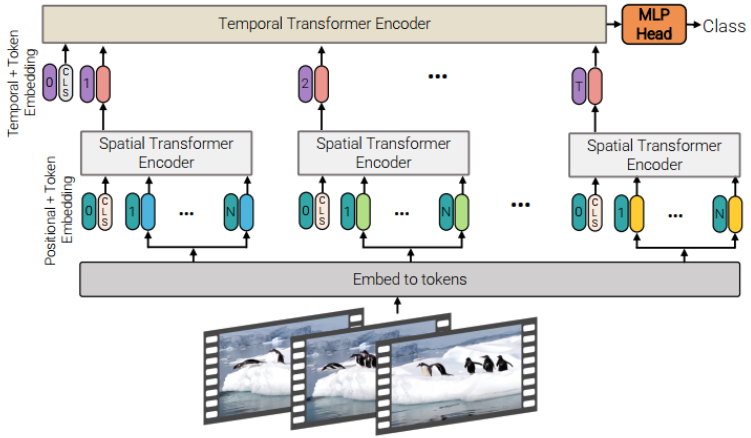


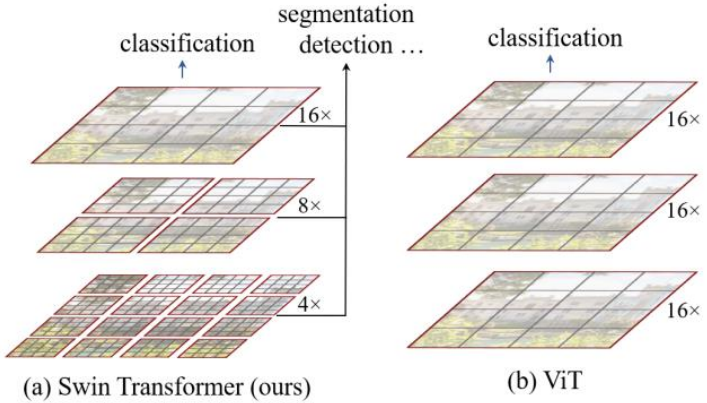


Liu et al., Video Swin Transformer, *Arxiv*, 2021

Limitation of Existing Video Transformers

- Though effective, both the space-time factorization and the local window-based attention scheme contradict the aim of applying full space-time attention, i.e., to *jointly* capture local and global spatiotemporal dependencies within one layer.

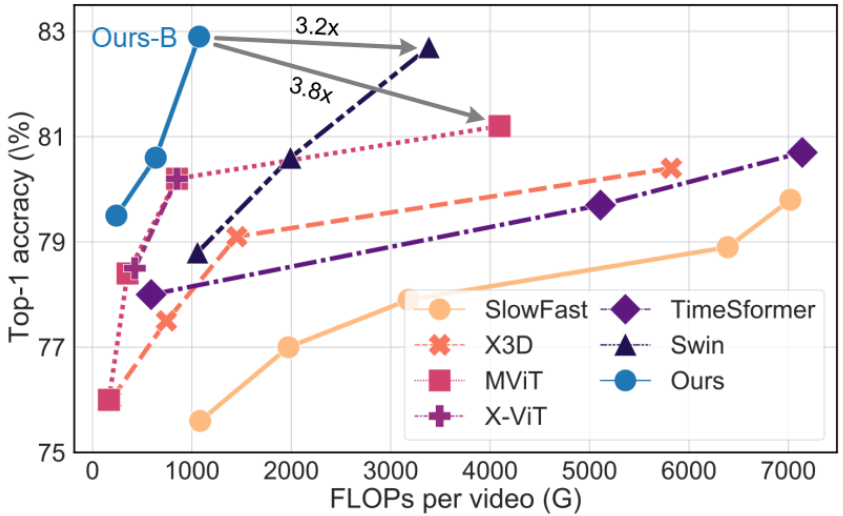




Contributions

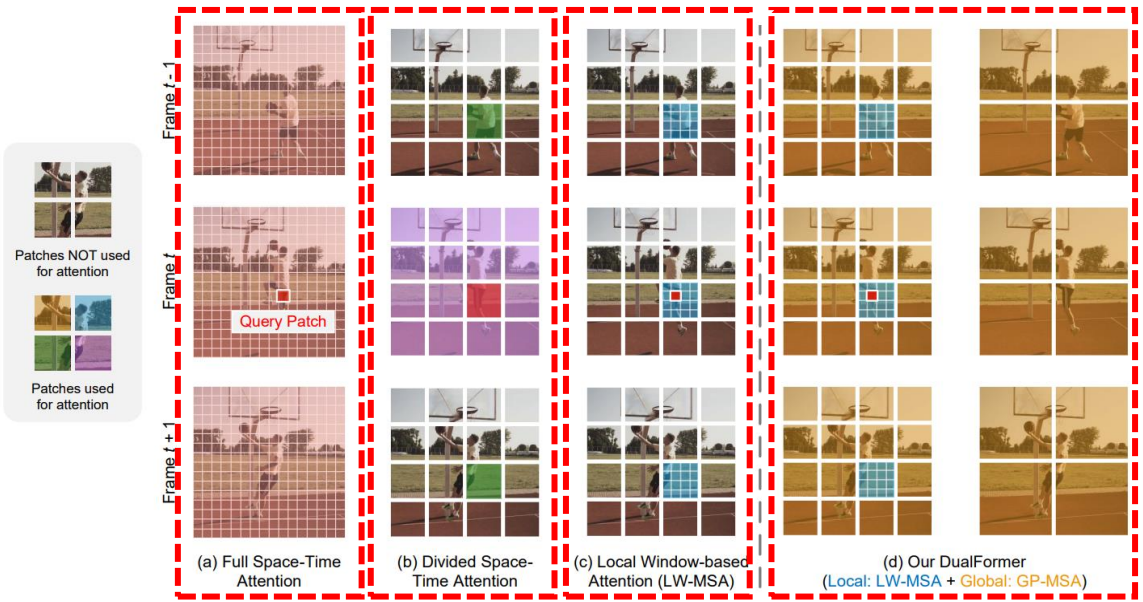
- In this paper, we present a new transformer architecture entitled DualFormer for efficient video recognition, which stratifies the full space-time attention into dual cascaded levels
  - Local Window-based Attention (LW-MSA) for extracting short-range interactions among nearby tokens
  - Global Pyramid-based Attention (GP-MSA) for capturing long-range dependencies between the query token and the coarse-grained global pyramid contexts

- We conduct extensive experiments on five popular video benchmarks to validate the superiority of our DualFormer
  - in terms of accuracy and FLOPs

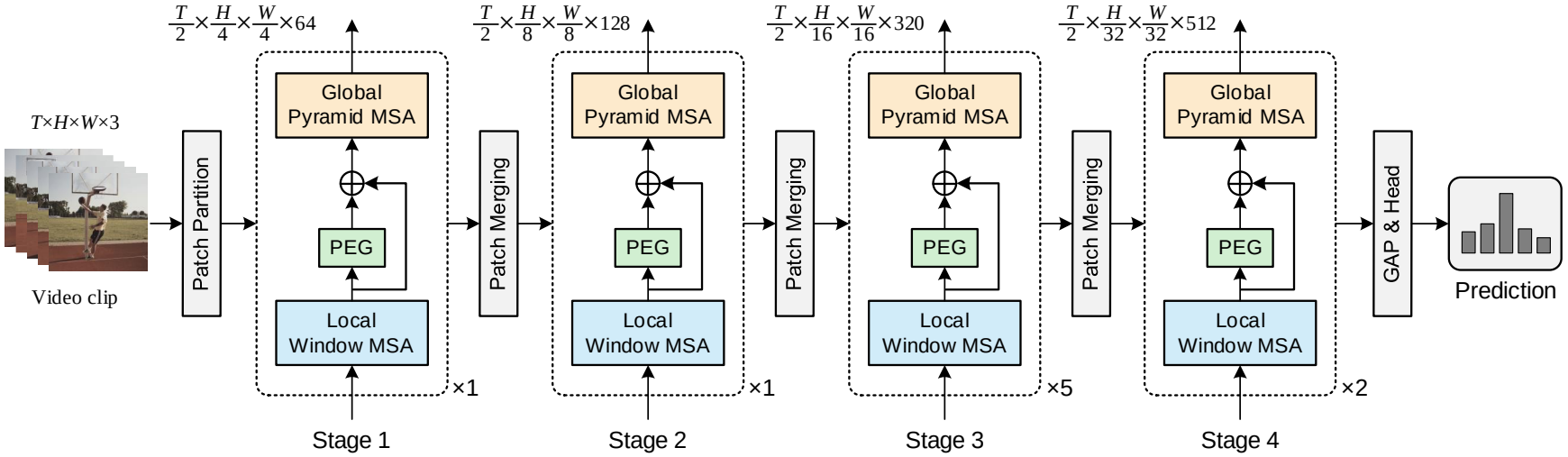


Comparison to Existing Space-Time MSAs

- We denote in red the query patch and in non-red colors its attention targets for each scheme.
- Multiple highlighted colors within a scheme indicate the MSA separately applied along different dimensions.



Overview of Architecture



PEG: position encoding generator

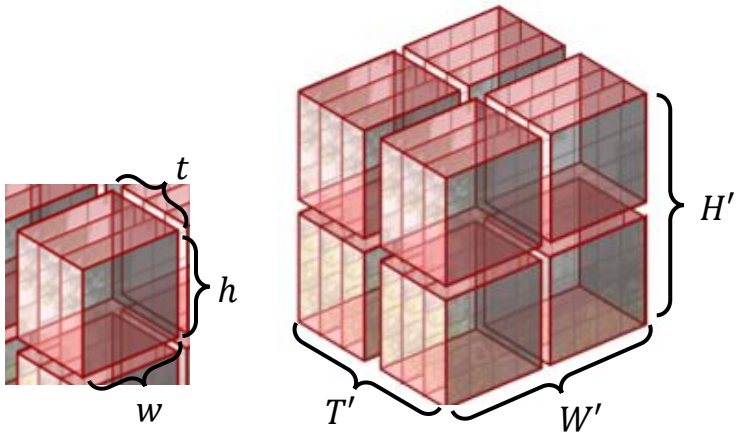
Local Window-based MSA (LW-MSA)

- LW-MSA conducts self-attention in non-overlapped 3D windows
  - **Target**: extracting short-range interactions among nearby tokens
  - Assume we partition the input feature map $T' \times H' \times W'$ into $t \times h \times w$ local windows
  - The number of tokens is $M = T'H'W'$ and the feature dimension is D
  - Computational complexity

$$\mathcal{O}(\text{LW-MSA}) = (thw)^2 D \times \frac{T'H'W'}{thw} = thwMD$$

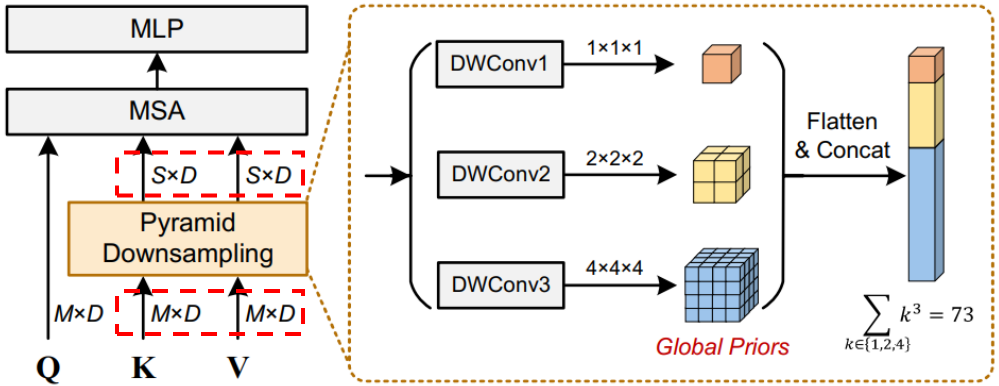
$\frac{M}{thw}$ times less

Standard MSA $\mathcal{O}(M^2 D)$

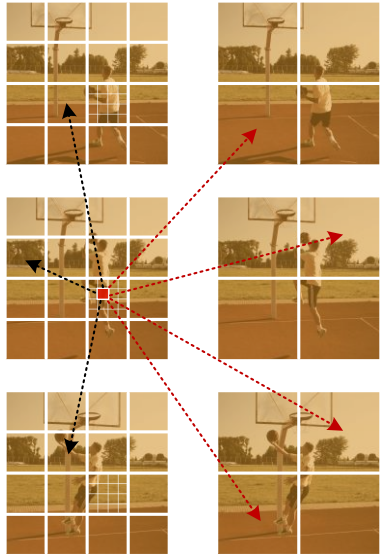


Global Pyramid-based MSA (GP-MSA)

- We further devise GP-MSA for capturing long-range dependencies between the query token and the coarse-grained global pyramid contexts
  - GP-MSA first applies the Pyramid Downsampling to reduce the number of keys/values
  - Then, performing MSA to learn representations



The number of tokens is $M = T'H'W'$ and S is the number of global priors.



Complexity of GP-MSA

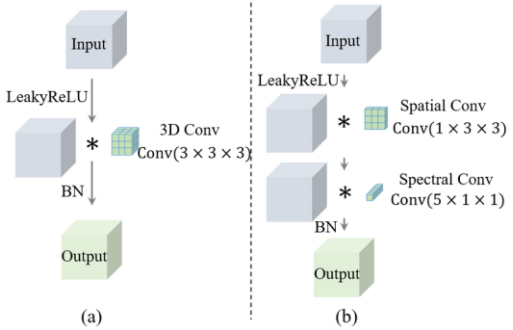
- Without loss of generality, suppose we have N_g pyramid scales and denote the size of global prior at the i-th scale as $\left(k_1^i, k_2^i, k_3^i\right)$
- The computational complexity of GP-MSA is

$$\mathcal{O}(\text{GP-MSA}) = \underbrace{MD\sum_{i=1}^{N_g} k_1^i k_2^i k_3^i}_{\text{MSA}} + \underbrace{\sum_{i=1}^{N_g}\left(\frac{T'H'W'}{k_1^i k_2^i k_3^i} k_1^i k_2^i k_3^i D\right)}_{\text{DWConv}}$$

$$= \left(\sum_{i=1}^{N_g} k_1^i k_2^i k_3^i + N_g\right)MD = (S + N_g)MD,$$

- Without loss of generality, suppose we have N_g pyramid scales and denote the size of global prior at the i-th scale as $\left(k_1^i, k_2^i, k_3^i\right)$
- The computational complexity of GP-MSA is

Decomposing 3D Conv to 2D+1D



$$= \left(S + \sum_{i=1}^{N_g}\left(\frac{k_1^i}{T'} + \frac{k_2^i k_3^i}{H'W'}\right)\right)MD$$

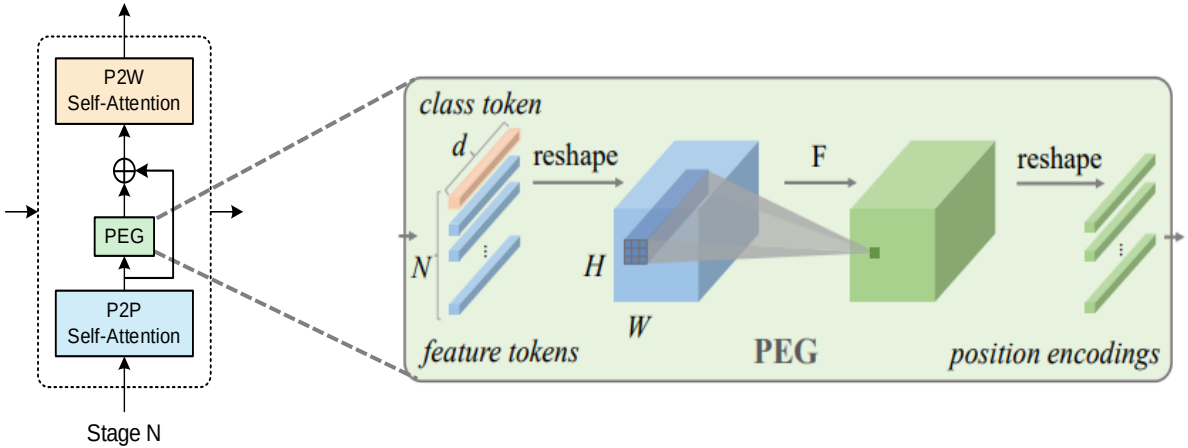
$$\approx \underline{SMD} \ll \underbrace{(S + N_g)MD}_{\text{Previous}} \ll \underbrace{M^2 D}_{\text{MSA}}.$$

$\frac{S}{M}$ time less than original MSA

For instance, at the first stage where *S* is 456 while *M* is 50176, the complexity has been reduced by ~110 times.

Position Encoding Generator (PEG)

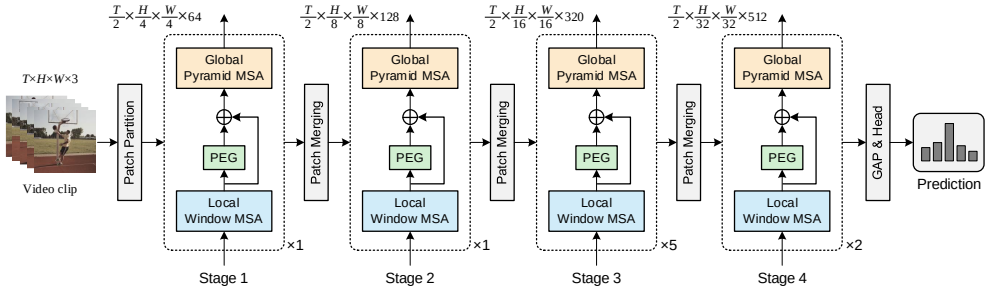
- As the self-attention operation is permutation-invariant, various position encoding methods are proposed to provide position information in video transformers
- Our implementation – Depth-wise convolution-based PEG
  - It encodes position information and can process variable-length inputs on the fly.
  - It can encode the absolute location of tokens to some extent



Chu et al., Conditional positional encodings for vision transformers , *arxiv*, 2021

Model Configuration

- Following Swin Transformer, we have three versions of DualFormer
- Each version has very similar parameter numbers to Swin

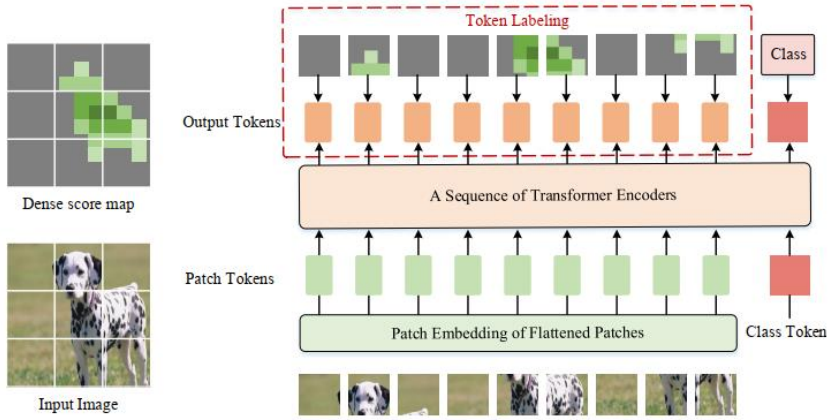


Stage	Layer	Tiny	Small	Base
Stage 1	Patch Merging	$p_1 = (2,4,4)$ $C_1 = 64$	$p_1 = (2,4,4)$ $C_1 = 96$	$p_1 = (2,4,4)$ $C_1 = 128$
Output: $\frac{T}{2}, \frac{H}{4}, \frac{W}{4}$	LW-MSA GP-MSA	$\begin{bmatrix}(8,7,7)\\(4,4,4)\\(8,7,7)\end{bmatrix}\times 1$	$\begin{bmatrix}(8,7,7)\\(4,4,4)\\(8,7,7)\end{bmatrix}\times 1$	$\begin{bmatrix}(8,7,7)\\(4,4,4)\\(8,7,7)\end{bmatrix}\times 1$
Stage 2	Patch Merging	$p_2 = (1,2,2)$ $C_2 = 128$	$p_2 = (1,2,2)$ $C_2 = 192$	$p_2 = (1,2,2)$ $C_2 = 256$
Output: $\frac{T}{2}, \frac{H}{8}, \frac{W}{8}$	LW-MSA GP-MSA	$\begin{bmatrix}(8,7,7)\\(4,4,4)\\(8,7,7)\end{bmatrix}\times 1$	$\begin{bmatrix}(8,7,7)\\(4,4,4)\\(8,7,7)\end{bmatrix}\times 1$	$\begin{bmatrix}(8,7,7)\\(4,4,4)\\(8,7,7)\end{bmatrix}\times 1$
Stage 3	Patch Merging	$p_3 = (1,2,2)$ $C_3 = 256$	$p_3 = (1,2,2)$ $C_3 = 384$	$p_3 = (1,2,2)$ $C_3 = 512$
Output: $\frac{T}{2}, \frac{H}{16}, \frac{W}{16}$	LW-MSA GP-MSA	$\begin{bmatrix}(8,7,7)\\(8,7,7)\end{bmatrix}\times 5$	$\begin{bmatrix}(8,7,7)\\(8,7,7)\end{bmatrix}\times 9$	$\begin{bmatrix}(8,7,7)\\(8,7,7)\end{bmatrix}\times 9$
Stage 4	Patch Merging	$p_4 = (1,2,2)$ $C_4 = 512$	$p_4 = (1,2,2)$ $C_4 = 768$	$p_4 = (1,2,2)$ $C_4 = 1024$
Output: $\frac{T}{2}, \frac{H}{32}, \frac{W}{32}$	LW-MSA GP-MSA	$\begin{bmatrix}(8,7,7)\\(whole)\end{bmatrix}\times 2$	$\begin{bmatrix}(8,7,7)\\(whole)\end{bmatrix}\times 1$	$\begin{bmatrix}(8,7,7)\\(whole)\end{bmatrix}\times 1$

Video-based Token Labeling

- **Basic idea**: all high-level tokens matter
- For the tiny and small version of DualFormer, we employ token labelling to improve their performance (using DualFormer-Base as the annotation model)

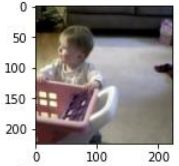
$$L_{total} = H(X^{cls}, y^{cls}) + \beta \cdot L_{tl},$$
$$= H(X^{cls}, y^{cls}) + \beta \cdot \frac{1}{N} \sum_{i=1}^{N} H(X^i, y^i),$$



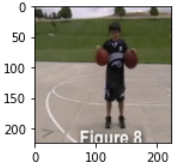
Jiang et al., All Tokens Matter: Token Labeling for Training Better Vision Transformers, *Arxiv*, 2021

Examples of Token Labeling

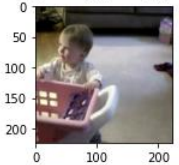
77	289	252	262	304	149	127
262	115	37	252	252	262	204
127	37	37	262	77	262	127
351	289	289	289	77	325	131
262	262	262	262	262	262	262
262	262	262	262	325	1	304
262	262	262	252	149	304	115



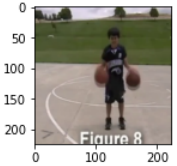
220	99	220	220	99	220	99
220	99	99	99	99	99	99
99	99	99	99	99	99	99
99	99	99	99	99	99	99
99	99	99	99	99	99	99
99	99	99	99	99	99	99
220	99	99	99	99	99	99



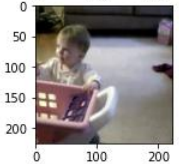
77	262	252	289	289	289	289
127	37	37	252	252	304	204
262	115	289	262	252	262	131
351	289	289	262	262	325	131
262	262	262	262	262	262	262
262	262	262	262	262	304	258
262	262	262	262	262	149	289



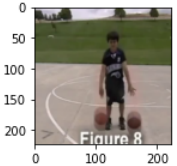
220	220	99	99	99	220	220
99	99	99	99	99	99	99
99	99	99	99	99	99	99
220	99	99	99	99	99	99
99	99	99	99	99	99	99
99	99	99	99	99	99	99
99	99	99	99	99	99	99



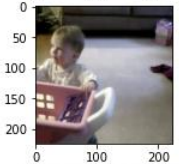
374	262	252	77	149	289	262
262	37	37	252	77	289	204
115	392	127	262	77	262	131
115	289	289	262	262	77	131
262	262	262	262	262	262	262
262	262	262	262	262	84	84
262	262	262	262	258	304	304



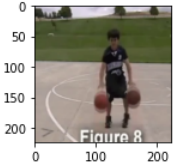
99	99	99	220	99	220	99
99	99	99	99	99	99	99
99	99	99	99	99	99	99
99	99	99	99	99	99	99
99	99	99	99	99	99	99
220	99	99	99	99	99	99
220	99	99	99	99	99	99



77	289	252	262	289	289	127
262	37	289	262	77	289	204
252	289	289	262	77	262	325
289	289	319	289	262	325	131
262	262	262	262	262	325	325
262	262	262	262	252	77	304
262	262	262	233	262	149	127

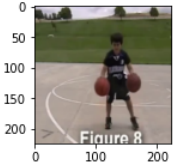


220	99	220	220	99	220	220
99	99	99	99	99	99	99
99	99	99	99	99	99	99
99	99	99	99	99	99	99
99	99	99	99	99	99	99
99	99	99	99	99	99	99
220	99	99	99	99	99	99



262	331	252	262	262	289	115
289	37	127	252	77	204	204
319	289	252	262	77	262	325
289	262	289	262	77	131	131
262	262	262	262	262	325	325
262	262	262	262	262	149	289
262	262	262	262	289	115	304

99	99	99	220	99	220	220
99	99	99	99	99	99	99
99	99	99	99	99	99	99
99	99	99	99	99	99	99
99	99	99	99	99	99	99
220	99	99	99	99	99	99
220	99	99	99	99	99	99

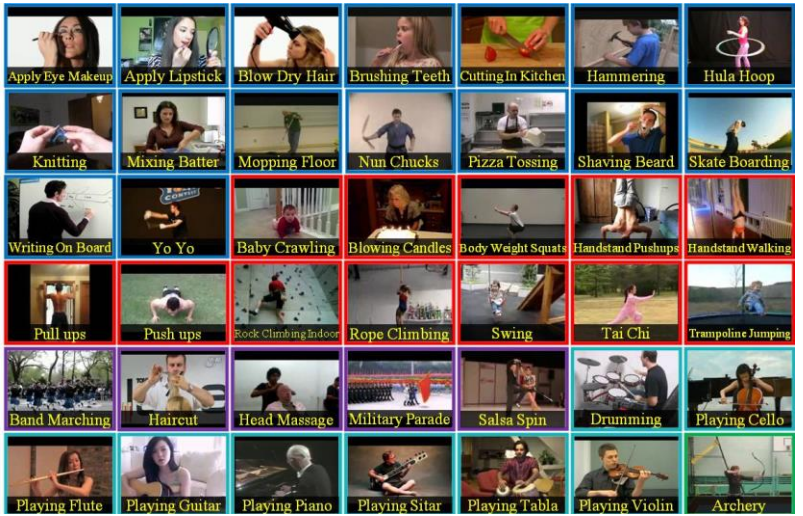


Experiments

- Video recognition benchmarks
  - Kinetics-400 and 600
  - Diving-48
  - UCF-101 and HMDB-51

Tips: Training DualFormer-T on Kinetics-400 takes ~31 hours on 8 A100 GPUs, while training a larger model DualFormer-B on Kinetics-400 requires around 3 days on 8 A100 GPUs.

	Year	Actions	Clips per class	Total
Kinetics-400	2017	400	400-1000	300k
Kinetics-600	2018	600	600-1000	500k

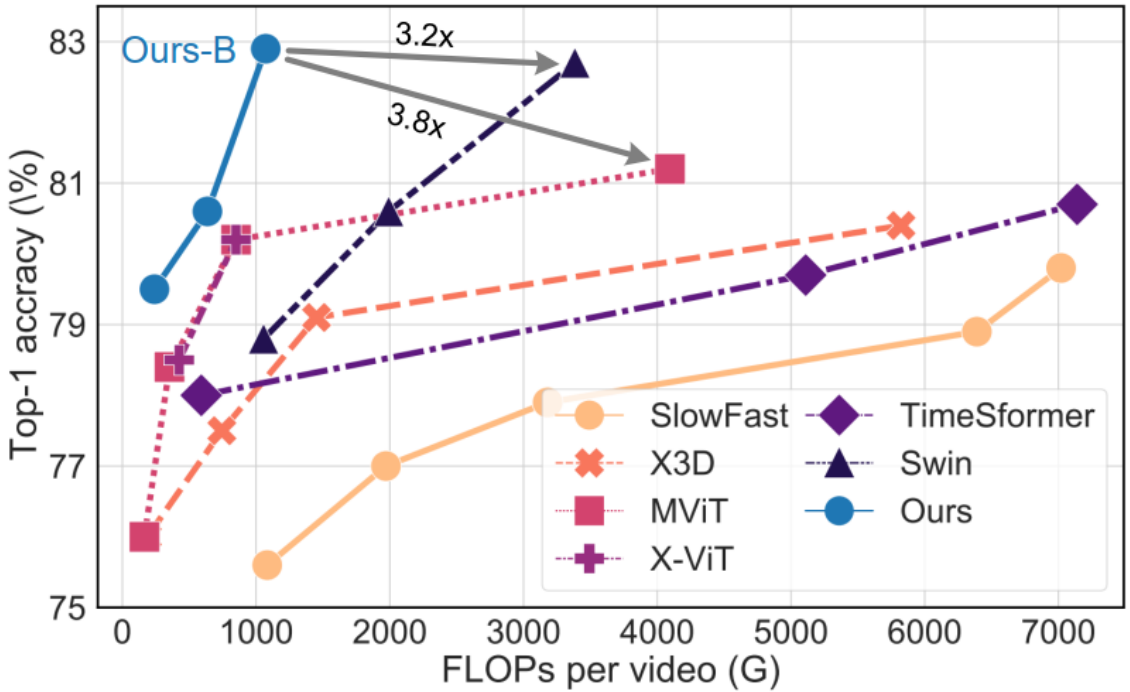


Comparison with SOTAs on Kinetics Datasets

- Our DualFormer achieves comparable performance while using 3.2x and 16.2x fewer FLOPs than Swin and ViViT, respectively

Method	Pretrain	Input	Views	Overall FLOPs	Param	Kinetics-400		Kinetics-600	
						Top-1	Top-5	Top-1	Top-5
R(2+1)D [51]	-	32 × 2	10 × 1	750	61.8	72.0	90.0	-	-
I3D [6]	ImageNet-1K	32 × 2	-	108	25.0	72.1	90.3	-	-
SlowFast+NL [15]	-	-	10 × 3	7020	59.9	79.8	93.9	81.8	95.1
X3D-XL [14]	-	16 × 5	10 × 3	1452	11.0	79.1	93.9	81.9	95.5
X3D-XXL [14]	-	16 × 5	10 × 3	5823	20.3	80.4	94.6	-	-
ip-CSN-152 [50]	IG-65M	8	10 × 3	3270	32.8	82.5	95.3	-	-
ViT-B-VTN [40]	ImageNet-21K	250 × 1	1 × 1	4218	11.0	78.6	93.7	-	-
TimeSformer-L [3]	ImageNet-21K	96 × 4	1 × 3	7140	121.4	80.7	94.7	82.2	95.5
MViT-B, 32×3 [13]	-	32 × 3	1 × 5	850	36.6	80.2	94.4	83.8	96.3
MViT-B, 64×3 [13]	-	64 × 3	3 × 3	4095	36.6	81.2	95.1	-	-
VidTr-L [63]	ImageNet-21K	32 × 2	10 × 3	10530	-	78.6	93.5	-	-
X-ViT (16×) [4]	ImageNet-21K	16 × 4	1 × 3	850	-	80.2	94.7	84.5	96.3
ViViT-L/16×2 [1]	ImageNet-21K	32 × 2	4 × 3	17352	310.8	80.6	94.7	82.5	95.6
ViViT-L/16×2 [1]	JFT-300M	32 × 2	4 × 3	17352	310.8	82.8	95.5	84.3	96.2
Swin-T [38]	ImageNet-1K	32 × 2	4 × 3	1056	28.2	78.8	93.6	-	-
Swin-S [38]	ImageNet-1K	32 × 2	4 × 3	1992	49.8	80.6	94.5	-	-
Swin-B [38]	ImageNet-1K	32 × 2	4 × 3	3384	88.1	80.6	94.6	-	-
Swin-B [38]	ImageNet-21K	32 × 2	4 × 3	3384	88.1	82.7	95.5	84.0	96.5
DualFormer-T (ours)	ImageNet-1K	32 × 2	4 × 1	240	21.8	79.5	94.1	-	-
DualFormer-S (ours)	ImageNet-1K	32 × 2	4 × 1	636	48.9	80.6	94.9	-	-
DualFormer-B (ours)	ImageNet-1K	32 × 2	4 × 1	1072	86.8	81.1	95.0	-	-
DualFormer-B (ours)	ImageNet-21K	32 × 2	4 × 1	1072	86.8	**82.9**	**95.5**	**85.2**	**96.6**

Comparison with SOTAs on Kinetics-400



Comparison with SOTAs on Other Datasets

- Diving-48: a temporally-heavy dataset
- HMDB-51 and UCF-101: test the transfer learning ability of DualFormer

Method	Input	Views	FLOPs	DIV	HMDB	UCF
I3D [6]	64×1	-	-	-	74.3	95.1
TSM [34]	8	-	-	-	70.7	94.5
TeiNet [37]	16	-	-	-	73.3	96.7
SlowFast [15]	16×8	-	-	77.6	-	-
VidTr-M [63]	16×4	10×3	5370	-	74.4	96.6
VidTr-L [63]	32×4	10×3	10530	-	74.4	96.7
TimeSformer [3]	8×4	1×3	590	75.0	-	-
TimeSformer-L [3]	96×4	1×3	7140	81.0	-	-
DualFormer-T*	16×4	4×1	28	75.4	74.6	96.3
DualFormer-T	16×4	4×1	28	75.9	75.0	96.6
DualFormer-S	32×4	4×1	636	81.2	76.2	97.4
DualFormer-S	32×4	4×3	1908	**81.8**	**76.4**	**97.5**

Ablation Study

- Effects of LW-MSA & GP-MSA
  - LL and GG mean only local or global MSA at that stage
  - G_1 and G_2 mean the (4,4,4) and (8,7,7) pyramid scale

Variants	FLOPs	Param	Top-1	Top-5
(LL, LL, LL, LL)	244	21.7	78.4	93.3
(GG, GG, GG, GG)	228	21.8	77.6	93.2
(LL, LL, LG, LG)	236	21.7	78.8	93.5
(LG, LG, LL, LL)	244	21.8	79.3	94.0
(LG_1, LG_1, LG_1, LG_1)	224	21.8	78.4	93.4
(LG_2, LG_2, LG_2, LG_2)	232	21.8	79.3	93.9
(LG, LG, LG, LG)	240	21.8	79.5	94.1

Ablation Study

- Effect of window size in LW-MSA
- Effect of position encoding

Input	Window Size	FLOPs	Top-1	Top-5
16×4	4×7×7	104	78.0	93.2
16×4	8×7×7	112	78.4	93.3
32×2	4×7×7	224	79.1	93.9
32×2	8×7×7	240	79.5	94.1
32×2	16×7×7	272	**79.7**	94.4
32×2	8×14×14	324	**79.7**	**94.5**

Method	Top-1
w.o PEG	78.9
Swin [38]	79.3
DWConv	**79.5**

Ablation Study

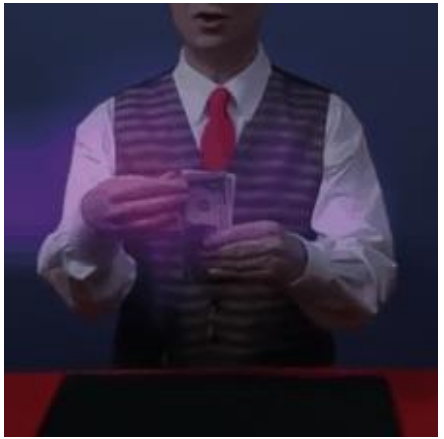
- Effects of token labelling
- Variant of token labelling

Method	TokenLabel	Top-1 (%)	Top-5 (%)
DualFormer-T		79.0	93.7
DualFormer-T	✓	79.5	94.1
DualFormer-S		80.3	94.5
DualFormer-S	✓	80.6	94.9

Method	Top-1	Top-5
w/o TL	79.0	93.7
EMA	79.1	93.7
TL-top1	79.4	94.0
TL-top5	79.5	94.1
Soft label	79.5	94.1
w. Focal loss	79.5	94.2

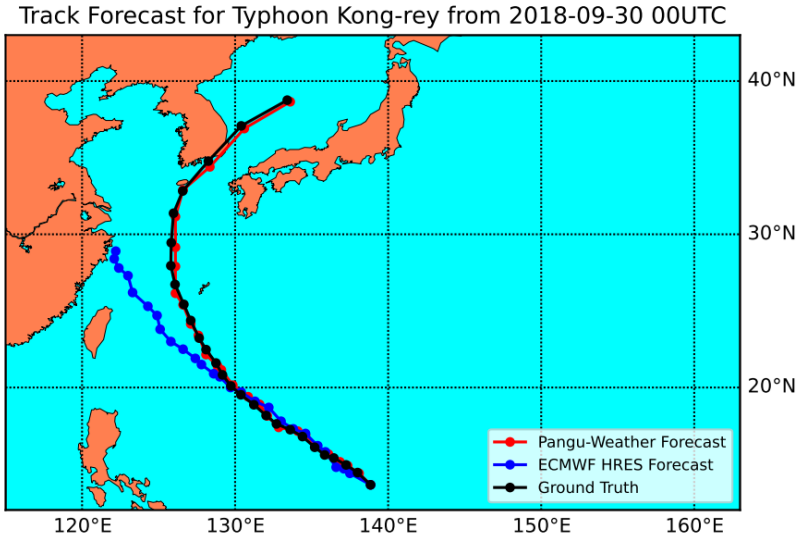
Visualization

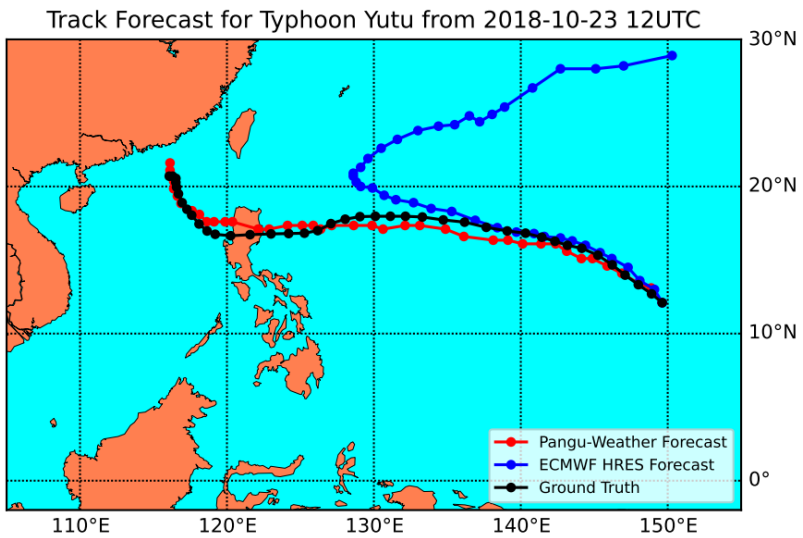
- We employ Grad-CAM to visualize the final layer



Application: Weather Forecasting

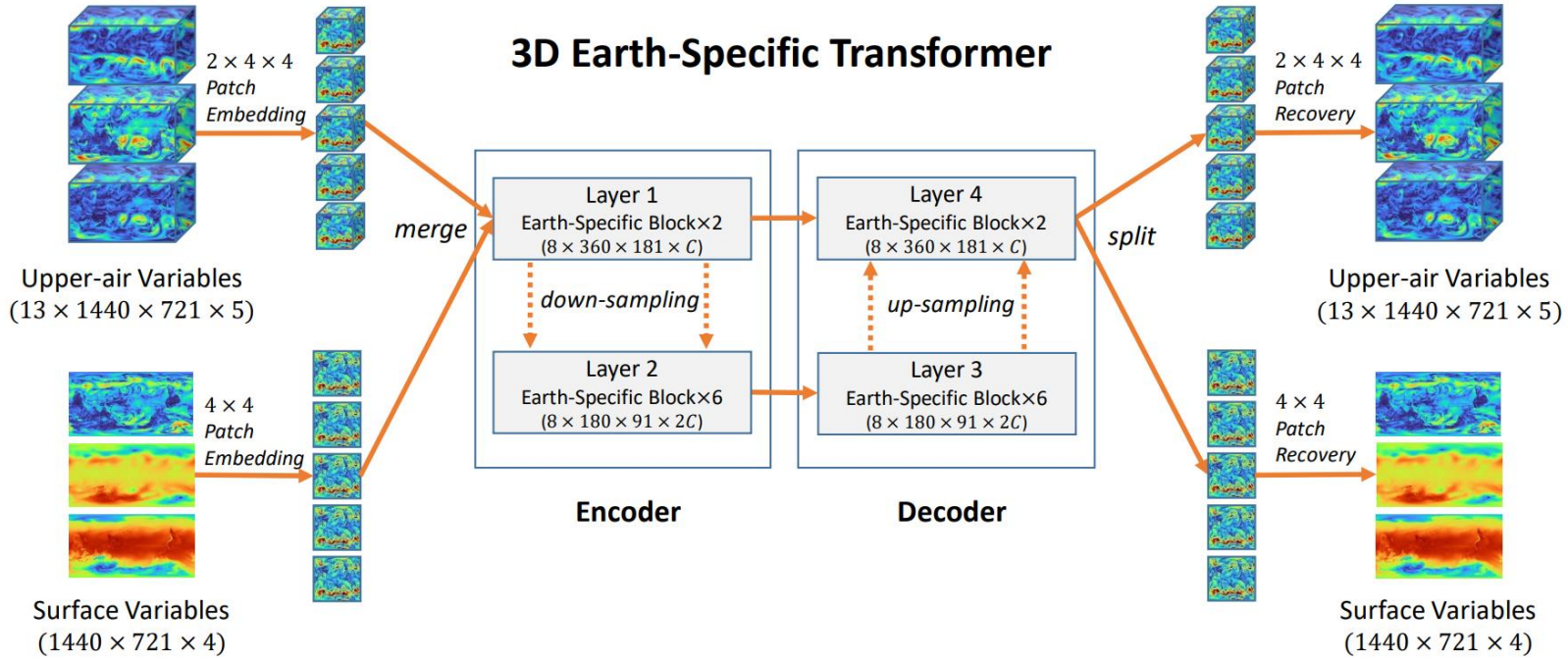
- Pangu-weather





K. Bi et al., Accurate medium-range global weather forecasting with 3D neural networks. Nature 2023.

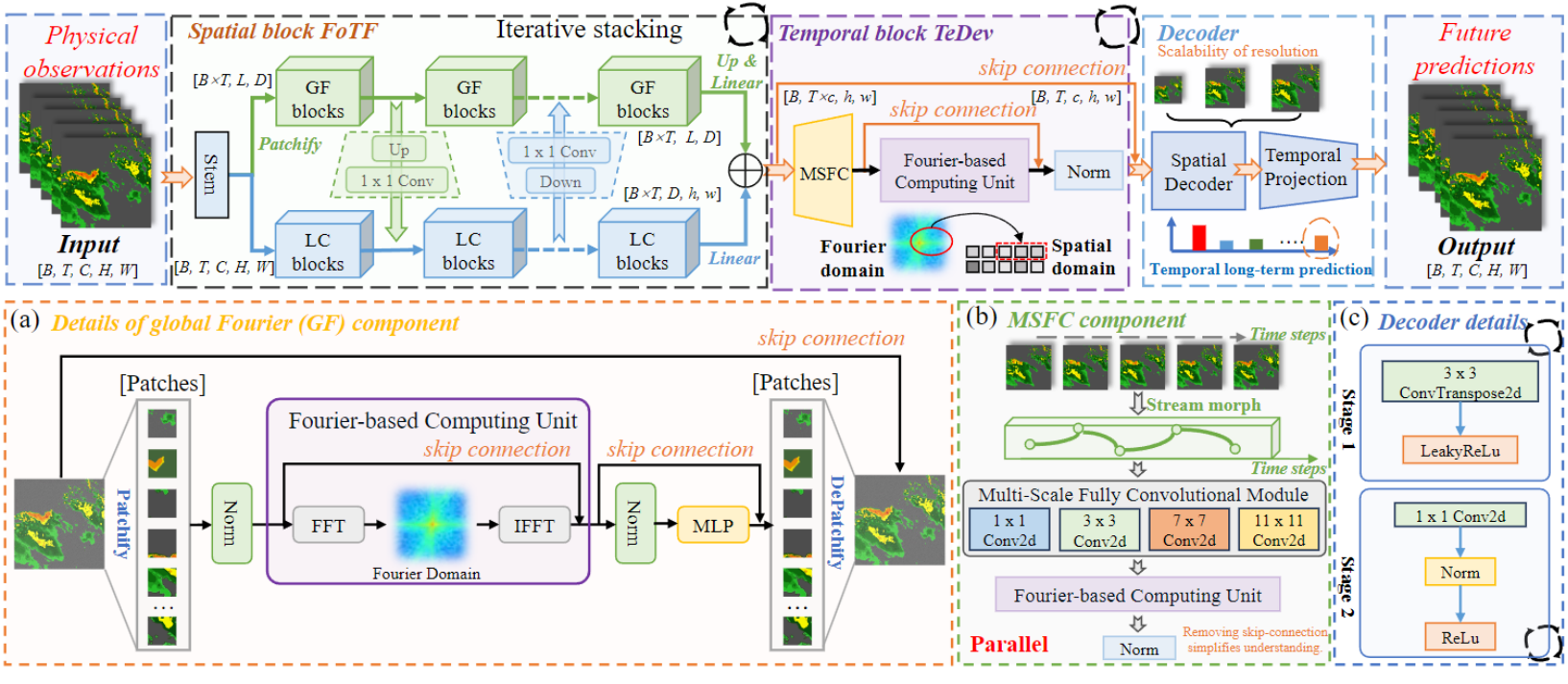
Pangu-Weather



K. Bi et al., Accurate medium-range global weather forecasting with 3D neural networks. Nature 2023.

Earth Science

- EarthFarseer



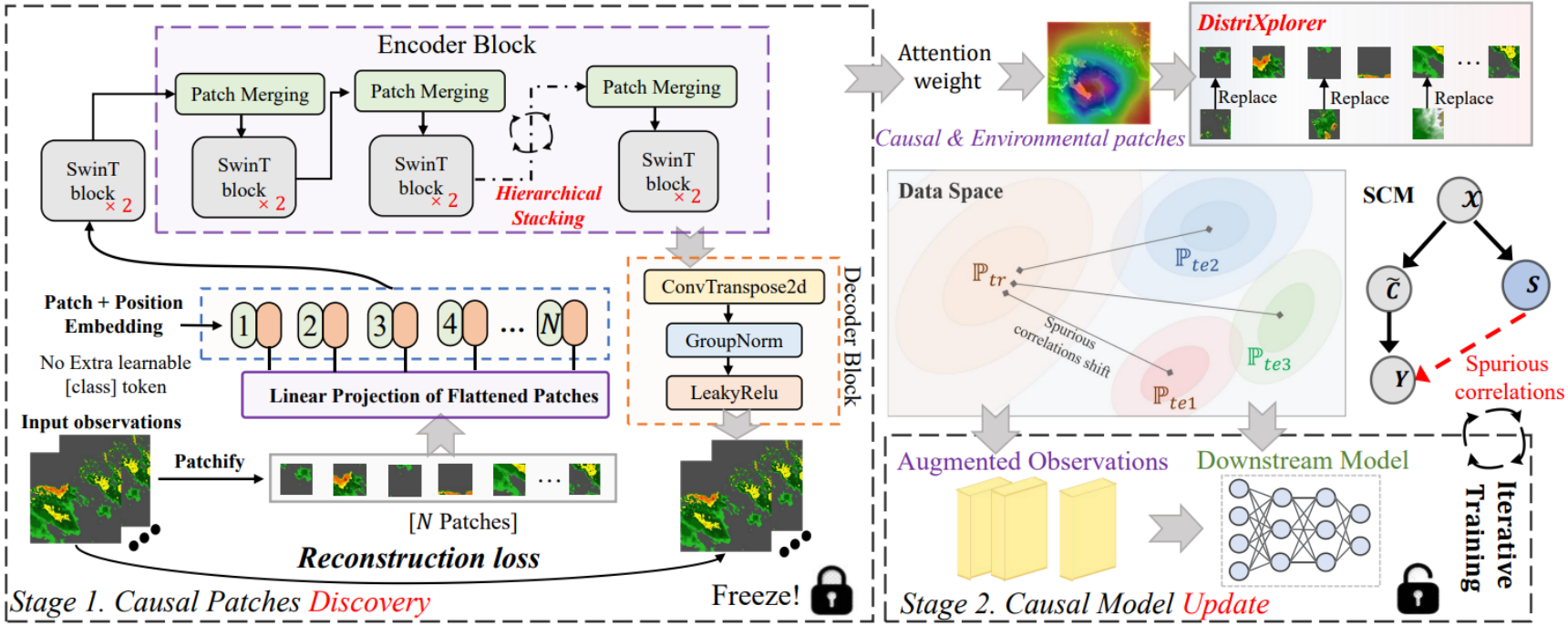
H. Wu, **Y. Liang** et al., Earthfarseer: Versatile Spatio-Temporal Dynamical Systems Modeling in One Model. AAAI 2023.

Experimental Results

Datasets	Metrics	Models												
		ConvLSTM	PredRNN-v2	E3D-LSTM	SimVP	VIT	SwinT	Rainformer	Earthformer	PhyDnet	Vid-ODE	PDE-STD	FourcastNet	Ours
MovingMNIST	MSE	103.3	56.8	41.3	15.1	62.1	54.4	85.8	41.8	24.4	22.9	23.1	60.3	14.9
	MAE	182.9	126.1	86.4	49.8	134.9	111.7	189.2	92.8	70.3	69.2	68.2	129.8	33.2
TaxiBJ+	MAE	5.5	4.3	4.1	3.0	3.4	3.2	- -	- -	4.2	3.9	3.7	- -	2.1
	MAPE	0.621	0.469	0.422	0.307	0.362	0.306	- -	- -	0.459	0.413	0.342	- -	0.243
KTH	MSE	126.2	51.2	86.2	40.9	57.4	52.1	77.3	48.2	66.9	49.8	65.7	102.1	31.8
	MAE	128.3	50.6	85.6	43.4	59.2	55.3	79.3	52.3	68.7	50.1	65.9	104.9	32.9
SEVIR	MSE	3.8	3.9	4.2	3.4	4.4	4.3	4.0	3.7	4.8	4.5	4.4	4.6	2.8
	CSI-M × 100	41.9	40.8	40.4	45.9	37.1	38.2	36.6	44.2	39.4	34.2	36.2	33.1	47.1
RainNet	RMSE	0.688	0.636	0.613	0.533	0.472	0.458	0.533	0.444	0.533	0.469	0.463	0.454	0.437
	MSE	0.472	0.405	0.376	0.284	0.223	0.210	0.284	0.197	0.282	0.220	0.215	0.206	0.191
PD	MSE	10.9	9.6	10.1	5.4	8.7	8.4	8.6	7.2	6.9	4.8	3.7	5.1	2.2
	MAE	100.3	95.4	100.2	50.9	81.2	79.5	80.9	73.4	68.7	47.6	38.9	52.4	21.8
RD	MSE × 10	21.2	20.9	18.2	9.5	13.2	12.1	9.7	11.4	10.8	9.8	9.8	10.2	9.4
	MAE	52.7	50.1	42.6	17.8	27.3	25.9	43.2	45.9	22.6	20.7	20.3	21.9	16.8
2DSWE	MSE × 100	11.2	8.9	6.4	3.1	8.1	7.6	7.8	7.4	4.9	4.5	4.3	5.2	2.6
	MAE	54.3	53.1	30.2	17.2	52.7	50.3	51.4	49.2	20.1	19.8	19.5	21.7	10.5
Avg Ranking		6.2	3.5	4.3	3.3	3.3	3.5	5.2	4.7	5.3	4.6	5.2	4.8	1.7

Causal Discovery for ST Grid Data

- NuwaDynamics



K. Wang, **Y. Liang*** et al., NuwaDynamics: Discovering and Updating in Causal Spatio-Temporal Modeling. ICLR 2024.

Experimental Results

Backbone ($10 \rightarrow 10$)	Metric	TaxiBJ+		KTH		SEVIR (CSI-M*)		RainNet		PD ($6 \rightarrow 6$)		FireSys	
		Ori	+NuWa	Ori	+NuWa	Ori	+NuWa	Ori	+NuWa	Ori	+NuWa	Ori	+NuWa
The upstream architecture is Transformer based and maintain consistency between upstream and downstream structures.													
ViT [2020]	MAE	3.48	2.27	59.32	34.56	37.07	46.88	0.78	0.74	83.45	24.70	3.21	3.09
	MSE	0.16	0.07	57.88	35.43	4.53	3.16	0.23	0.19	8.99	2.45	8.27	8.19
	Δ		0.09		22.45		1.37		0.04		6.51		0.08
SwinT [2021]	MAE	3.22	2.18	55.44	33.45	38.22	45.68	0.67	0.66	79.53	26.38	2.98	2.76
	MSE	0.21	0.11	52.38	33.11	4.37	2.84	0.22	0.19	8.47	3.15	7.96	7.65
	Δ		0.10		19.27		1.89		0.03		5.32		0.31
Rainformer [2022]	MAE	––	––	80.32	40.77	36.68	46.88	1.21	1.17	81.23	30.54	4.65	4.55
	MSE	––	––	77.99	40.75	4.02	3.38	0.30	0.21	8.63	2.51	11.27	10.72
	Δ	––	––		37.24		0.64		0.09		6.12		0.55
Earthformer [2022b]	MAE	––	––	52.37	42.91	44.21	46.33	1.98	1.54	73.24	30.78	1.97	1.57
	MSE	––	––	48.65	37.21	3.88	2.96	0.20	0.19	7.32	2.44	5.17	4.94
	Δ	––	––		11.44		0.92		0.01		4.88		0.23
The upstream architecture is ViT and downstream does not specify a particular model architecture.													
ConvLSTM [2015]	MAE	5.52	3.27	128.33	53.10	41.93	44.88	3.98	3.64	100.44	58.39	11.21	10.97
	MSE	0.33	0.27	126.32	89.35	3.84	3.17	0.49	0.30	10.98	5.47	17.22	16.43
	Δ		0.06		36.97		0.67		0.19		5.51		0.79
PredRNN-V2 [2022b]	MAE	4.33	3.25	51.38	40.37	40.83	44.99	2.67	2.43	95.43	72.77	4.32	3.97
	MSE	0.27	0.20	51.36	45.76	3.98	3.17	0.41	0.33	9.65	7.35	5.87	4.53
	Δ		0.07		5.60		0.81		0.08		2.30		1.34
E3D-LSTM [2018b]	MAE	4.25	3.27	86.37	52.98	40.56	45.38	3.88	3.72	100.23	78.34	4.98	4.65
	MSE	0.29	0.25	87.69	59.49	4.37	3.89	0.38	0.29	10.34	7.35	8.76	8.12
	Δ		0.04		28.20		0.48		0.09		2.99		0.64
SimVP [2022a]	MAE	3.07	2.56	43.39	33.98	45.98	47.09	1.27	1.02	50.93	31.55	1.98	1.54
	MSE	0.14	0.07	40.93	32.89	3.44	2.92	0.28	0.20	5.48	3.24	2.65	2.42
	Δ		0.07		8.04		0.52		0.08		2.24		0.23

Q&A

Thanks!

CityMind Lab

amazon Microsoft Tencent Alibaba.com 京东 JD.COM CAINIAO 菜鸟