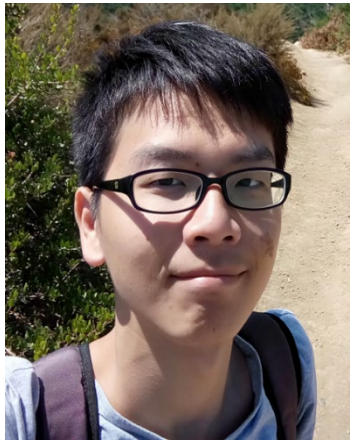
CVPR 2023

Learning A Sparse Transformer Network for Effective Image Deraining

Xiang Chen[1] Hao Li[1] Mingqiang Li[2] Jinshan Pan[1*]
[1]School of Computer Science and Engineering, Nanjing University of Science and Technology
[2]Information Science Academy, China Electronics Technology Group Corporation


Xiang Chen


Hao Li


Mingqiang Li


Jinshan Pan

Low-Level Vision

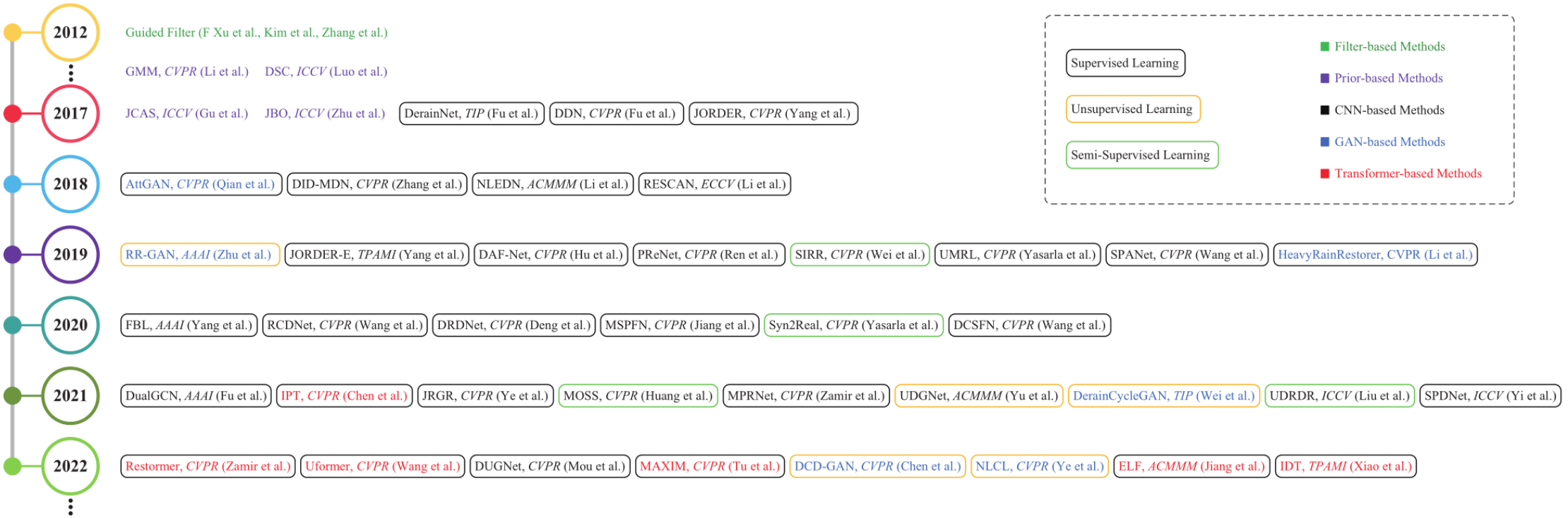
The inverse problem: $Y = F(X)$

Observation Y

- **Image Denoising**
- **Image Deblurring**
- **Image Dehazing**
- **Image Deraining**
- **Image Enhancement**
- **…**

Recovery X

Related Work



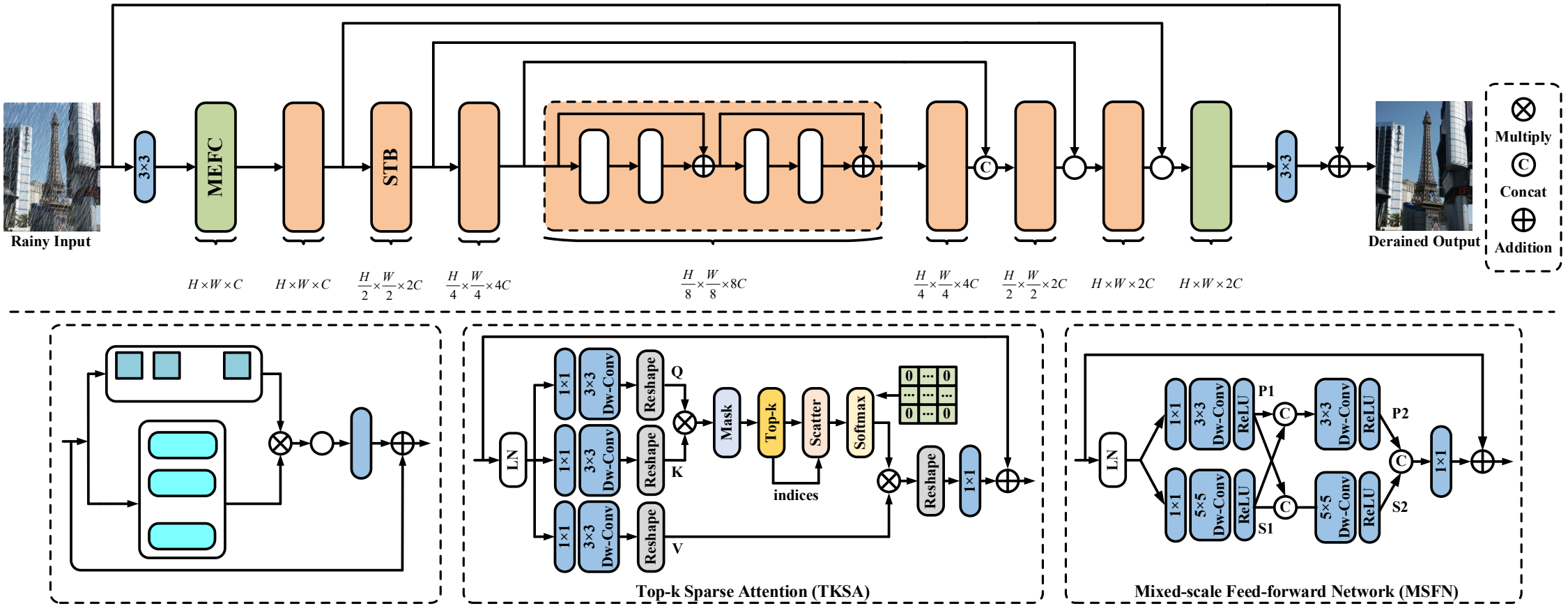
Towards High-quality Mapping Relationship

Motivation

- Most existing Transformers usually use **all similarities** of the tokens from the query-key pairs for the feature aggregation.
- However, if the tokens from the query are different from those of the key, the self-attention values estimated from these tokens also involve in feature aggregation, which accordingly **interferes with the clear image restoration**.
- Thus, these findings motivate us to explore the **most useful self-attention values** so that we can make full use of the features for better image restoration.

Sparse Transformer Network

- **Overall network architecture**



Low-Pass Filter and Self-Attention

- **Proposition**
  - ***Low-pass filter increases the value of the minimum-intensity pixel and decreases the value of the maximum-intensity pixel***

$$\min_{z \in \Omega} I(z) \leq B(x) \leq \max_{z \in \Omega} I(z)$$

⇩

Over-smoothed results

Low-Pass Filter and Self-Attention

- **Self-attention is a special low-pass filter**

$$\mathbf{Z} = \mathbf{S} \circledast \mathbf{V} = \mathrm{softmax}\left(\frac{\boldsymbol{Q}(\boldsymbol{F})\boldsymbol{\mathcal{K}}(\boldsymbol{F})^T}{\sqrt{d}}\right)\boldsymbol{\mathcal{V}}(\boldsymbol{F})$$

$\mathbf{S}_{xy} \geq 0$ and $\sum_y \mathbf{S}_{xy} = 1$

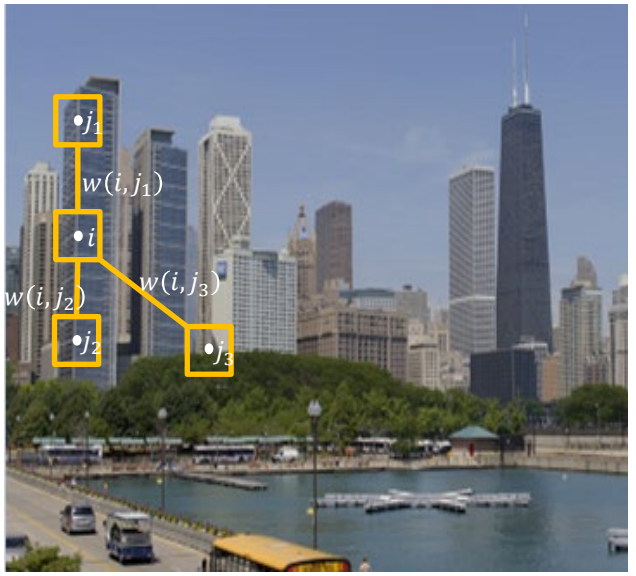
- $\mathrm{softmax}(\cdot)$ **usually interferes with the similarities of each token**

Low-Pass Filter and Self-Attention

- $\mathrm{softmax}(\cdot)$ **usually interferes with the similarities of each token**

	j_1	j_2	j_3	$\cdots$	j_n
$\vdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
$i+1$	0.680	0.537	0.403	$\cdots$	$\cdots$
i	0.794	0.862	0.351	$\cdots$	$\cdots$
$i-1$	0.475	0.584	0.236	$\cdots$	$\cdots$
$\vdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$

softmax



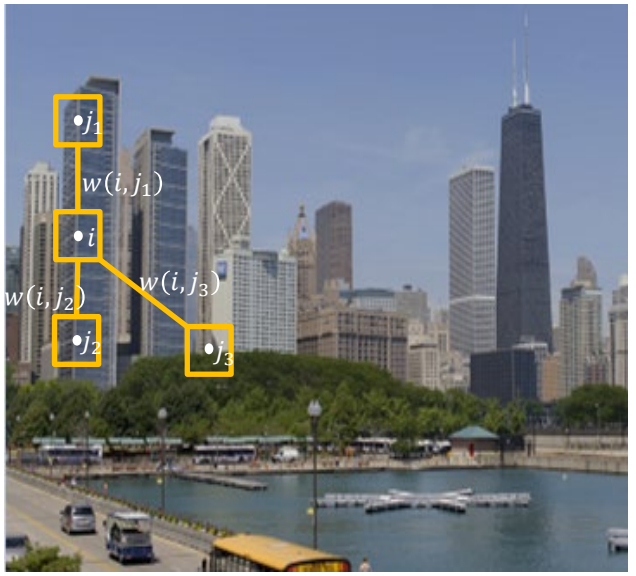
	j_1	j_2	j_3	$\cdots$	j_n	
	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\vdots$
	0.141	0.112	0.084	$\cdots$	$\cdots$	$i+1$
	0.179	0.194	0.079	$\cdots$	$\cdots$	i
	0.084	0.103	0.041	$\cdots$	$\cdots$	$i-1$
	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\vdots$

Low-Pass Filter and Self-Attention

- **Keep the information of the most relevant tokens**

	j_1	j_2	j_3	$\cdots$	j_n
$\vdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
$i+1$	0.680	0.537	0.403	$\cdots$	$\cdots$
i	0.794	0.862	0.351	$\cdots$	$\cdots$
$i-1$	0.475	0.584	0.236	$\cdots$	$\cdots$
$\vdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$

softmax



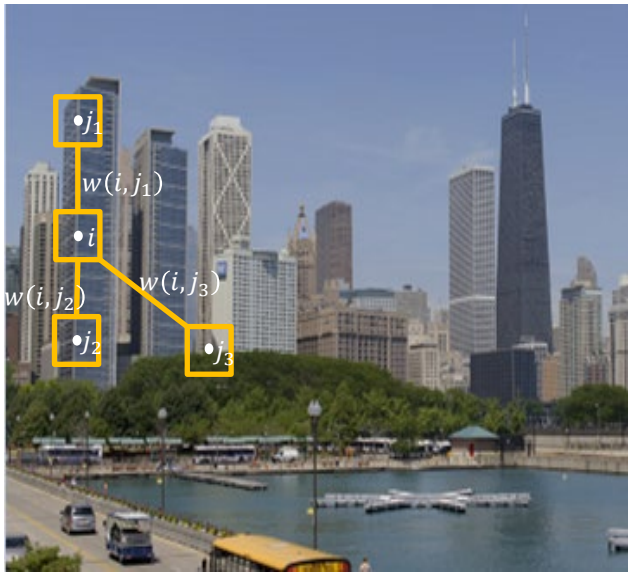
	j_1	j_2	j_3	$\cdots$	j_n	
	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\vdots$
	0.135	0.107	0.090	$\cdots$	$\cdots$	$i+1$
	0.171	0.186	0.084	$\cdots$	$\cdots$	i
	0.088	0.095	0.046	$\cdots$	$\cdots$	$i-1$
	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\vdots$

Low-Pass Filter and Self-Attention

- Keep the information of the most relevant tokens

	j_1	j_2	j_3	$\cdots$	j_n
$\vdots$	...	...	...	...	...
$i+1$	0.680	0.537	0.403	...	...
i	0.794	0.862	0.351	...	...
$i-1$	0.475	0.584	0.236	...	...
$\vdots$	...	...	...	...	...

softmax



	j_1	j_2	j_3	$\cdots$	j_n	
	...	...	...	...	...	$\vdots$
	0.[illegible]5	0.[illegible]7	0.[illegible]0	...	...	$i+1$
	0.[illegible]1	0.[illegible]6	0.[illegible]4	...	...	i
	0.[illegible]8	0.[illegible]5	0.[illegible]6	...	...	$i-1$
	...	...	...	...	...	$\vdots$

Low-Pass Filter and Self-Attention

- Keep the information of the most relevant tokens

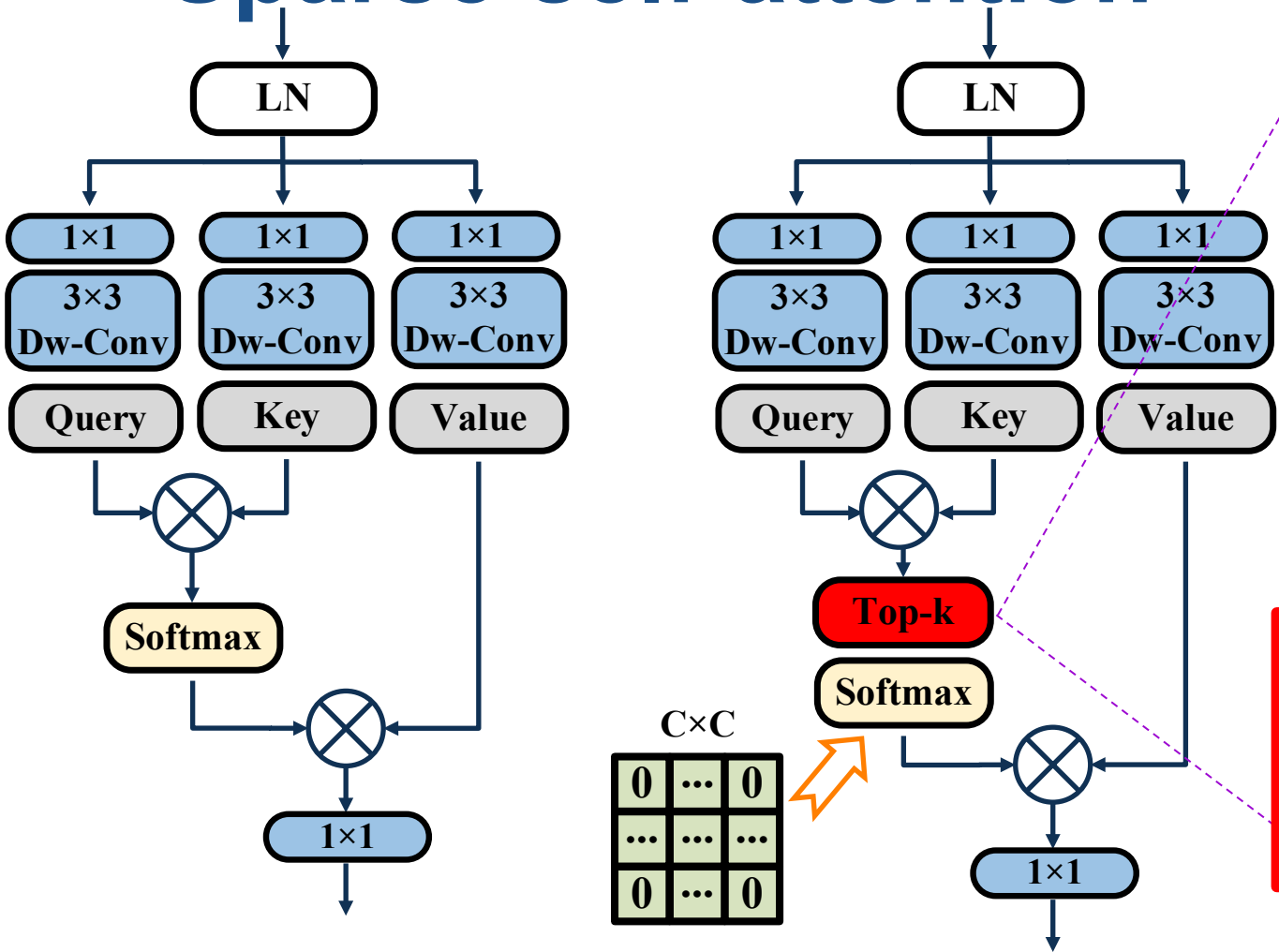
How to remove irrelevant tokens?

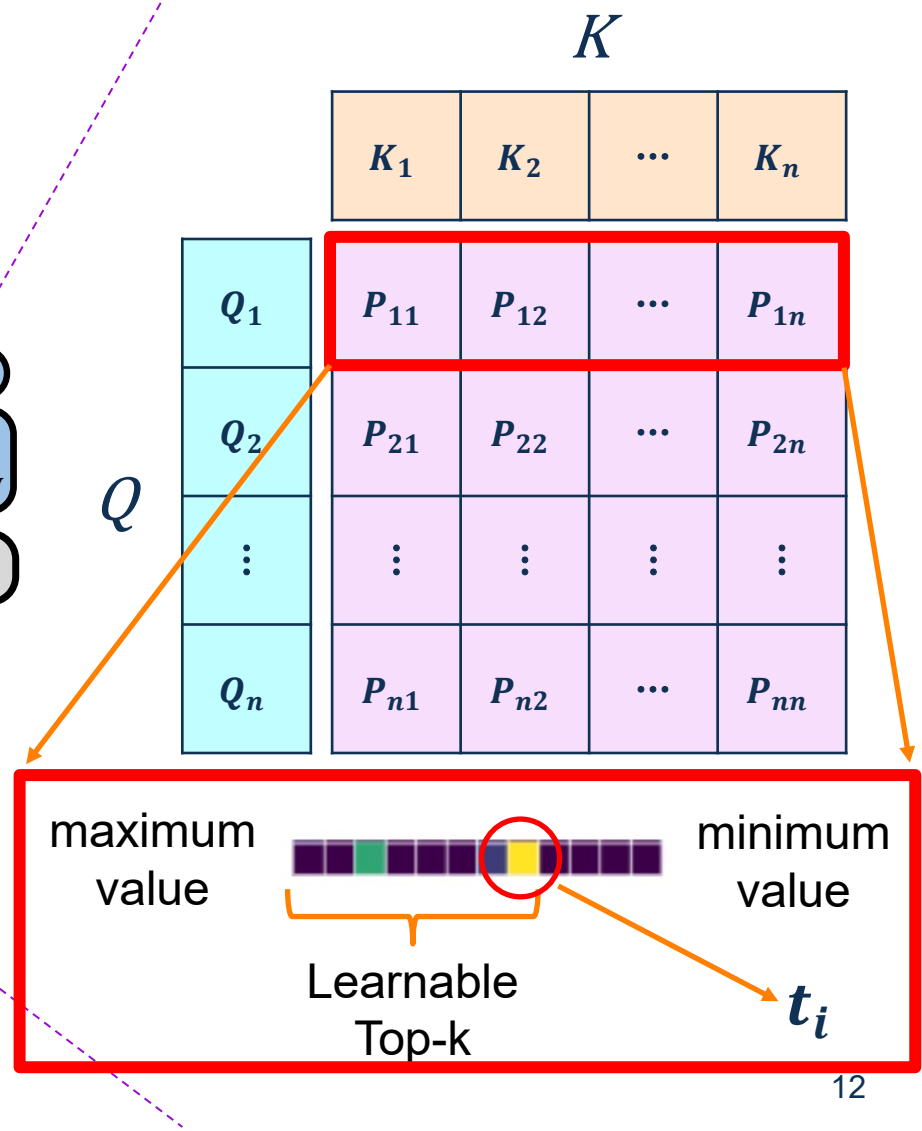
$i+1$	0.660	0.357	0.405		
i	0.794	0.862	0.351	⋯	⋯
$i-1$	0.475	0.584	0.236	⋯	⋯
⋮	⋯	⋯	⋯	⋯	⋯

					$i+1$
0.1	0.6	0.4	⋯	⋯	i
0.8	0.5	0.6	⋯	⋯	$i-1$
⋯	⋯	⋯	⋯	⋯	⋮

Sparse Transformer

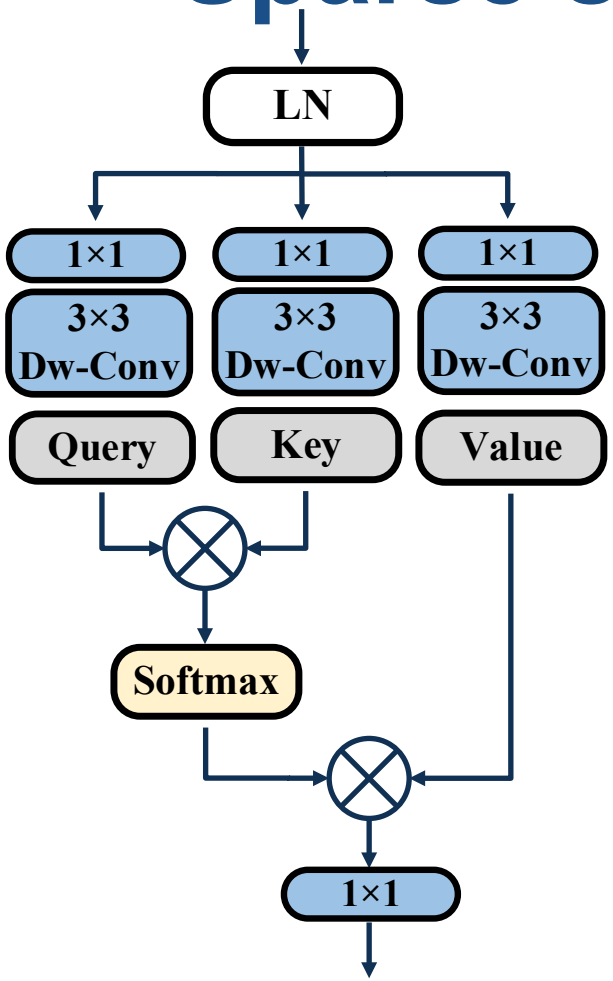
- Sparse self-attention

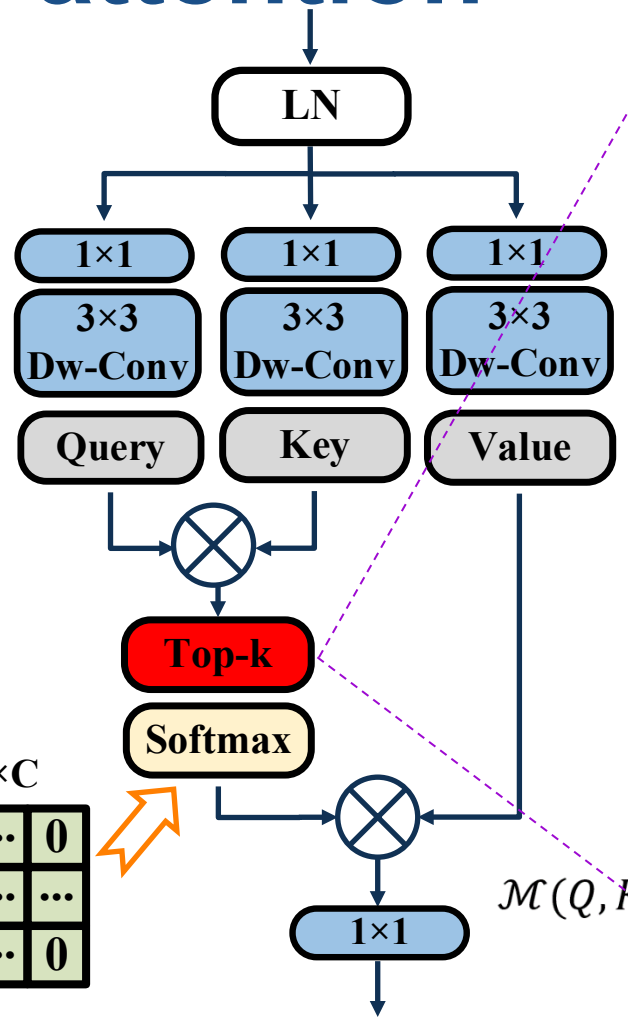


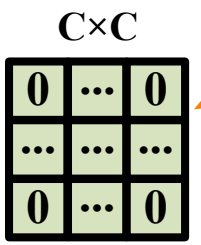


Sparse Transformer

- Sparse self-attention







K

Q	K_1	K_2	···	K_n
Q_1	P_{11}	0	···	P_{1n}
Q_2	0	P_{22}	···	0
⋮	⋮	⋮	⋮	⋮
Q_n	0	0	···	P_{nn}

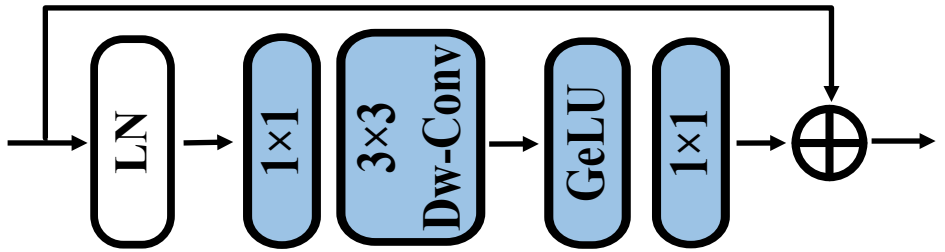
maximum value

minimum value

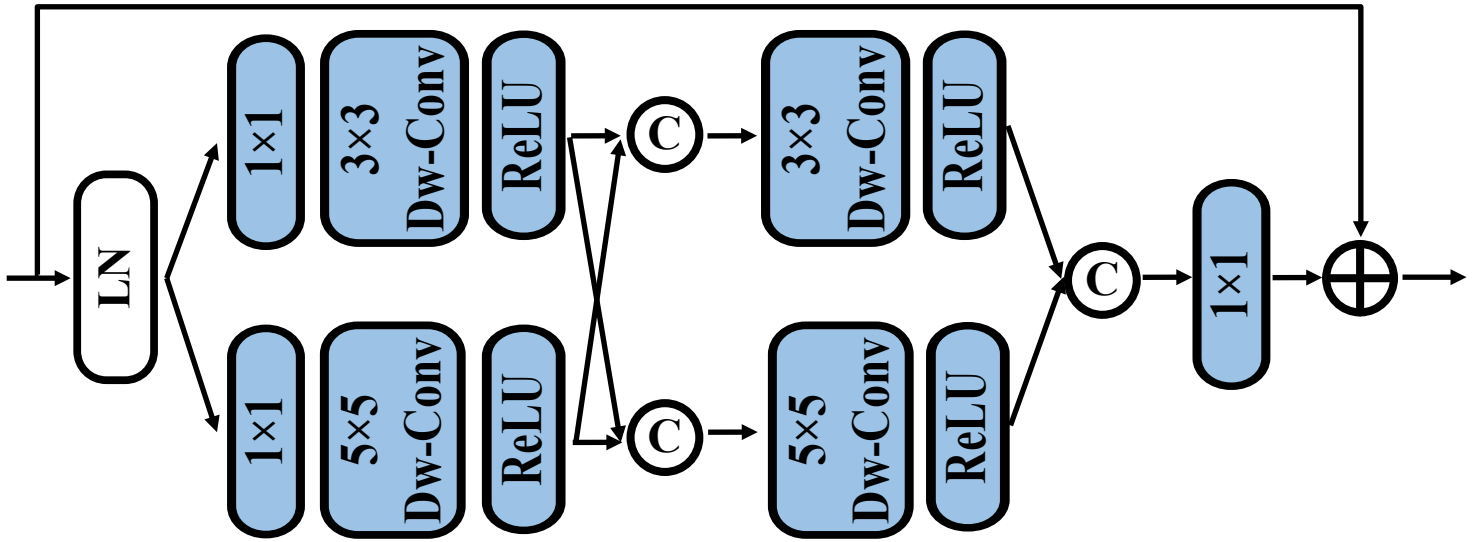
$$\mathcal{M}(Q,K)_{ij} = \begin{cases} P_{ij} \text{ if } P_{ij} \geq t_i(k\text{-th largest value of row } i) \\ 0 \text{ if } P_{ij} < t_i(k\text{-th largest value of row } i) \end{cases}$$

Sparse Transformer

- **Mixed-scale feed-forward network**



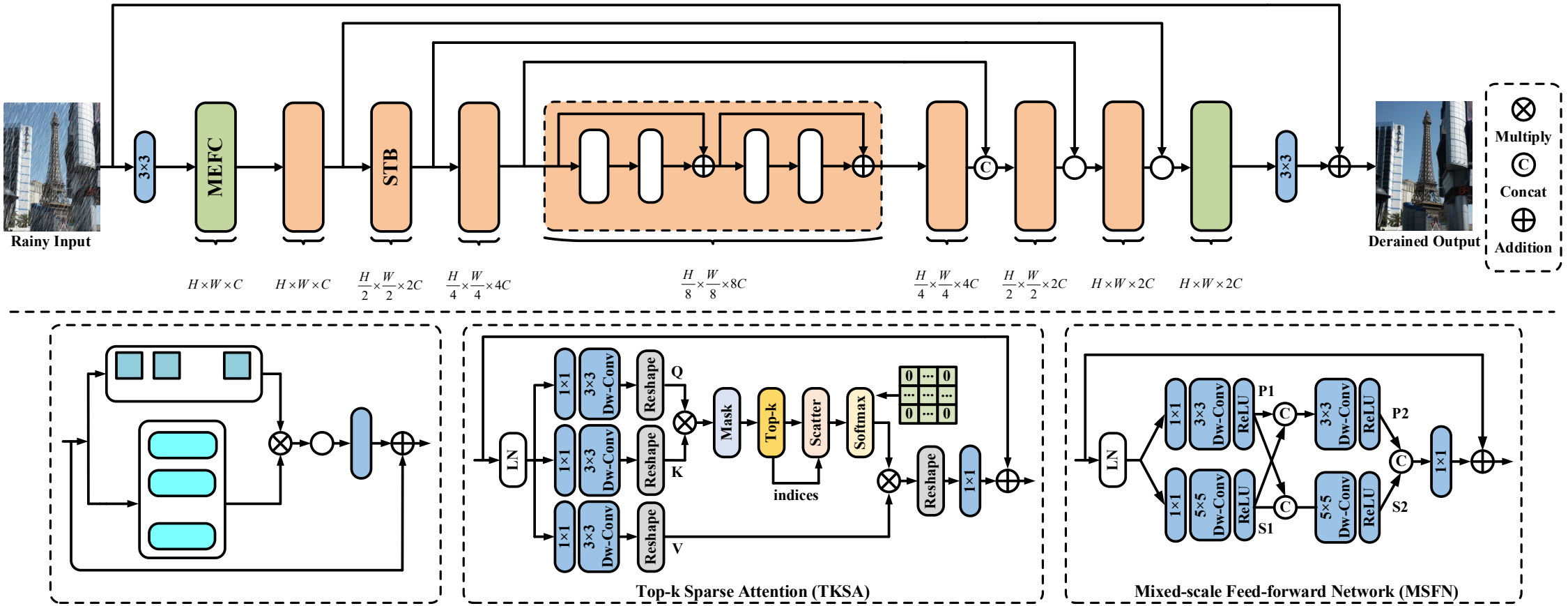
Single-scale feed-forward network



Mixed-scale feed-forward network

Sparse Transformer Network

- **Overall network architecture**



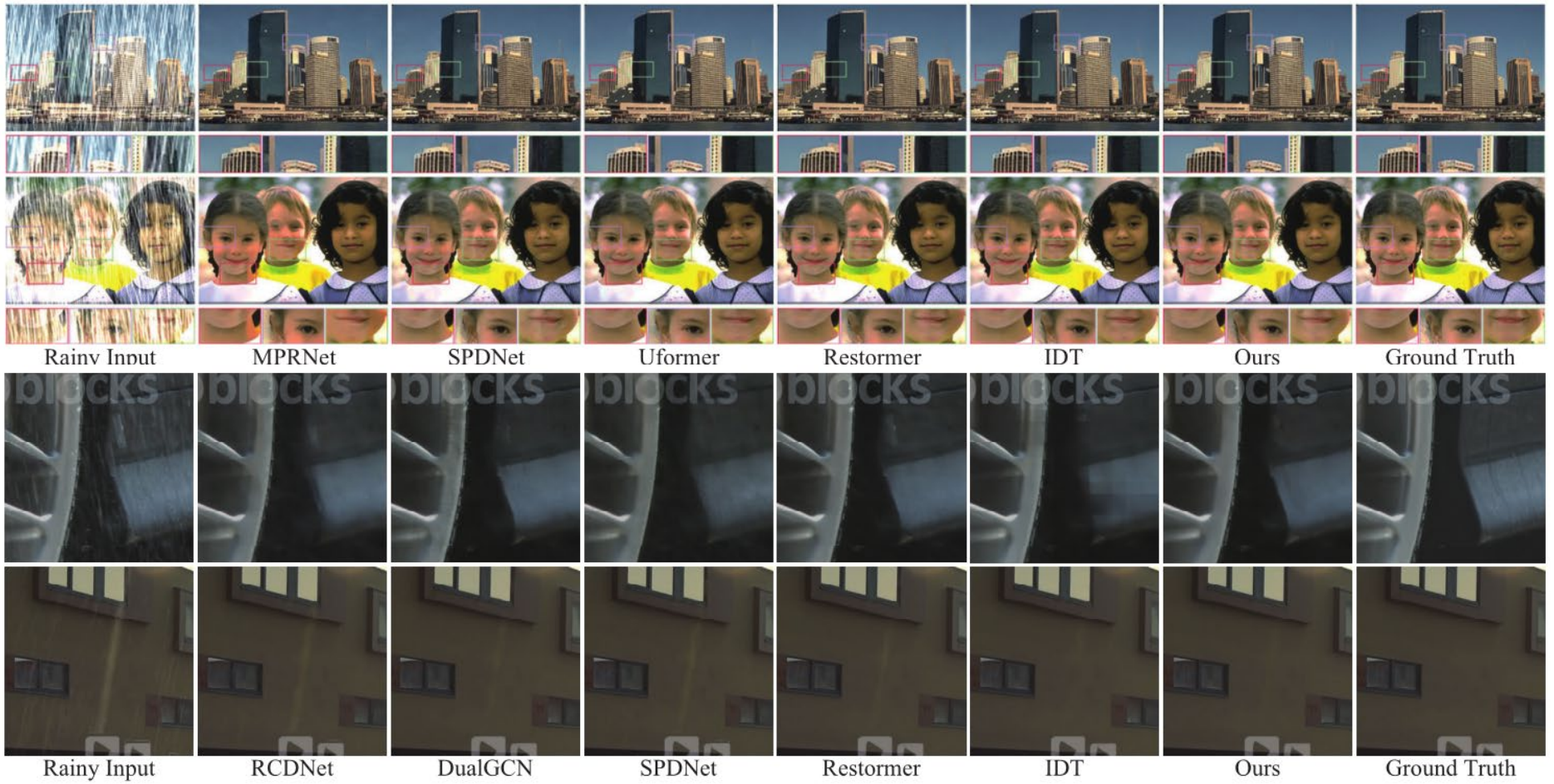
Sparse Transformer Network

- **Quantitative evaluations**

Datasets	Rain200L		Rain200H		DID-Data		DDN-Data		SPA-Data	
Metrics	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
MSPFN	38.58	0.9827	29.36	0.9034	33.72	0.9550	32.99	0.9333	43.43	0.9843
RCDNet	39.17	0.9885	30.24	0.9048	34.08	0.9532	33.04	0.9472	43.36	0.9831
MPRNet	39.47	0.9825	30.67	0.9110	33.99	0.9590	33.10	0.9347	43.64	0.9844
DualGCN	40.73	0.9886	31.15	0.9125	34.37	0.9620	33.01	0.9489	44.18	0.9902
SPDNet	40.50	0.9875	31.28	0.9207	34.57	0.9560	33.15	0.9457	43.20	0.9871
Uformer	40.20	0.9860	30.80	0.9105	35.02	0.9621	33.95	0.9545	46.13	0.9913
Restormer	40.99	0.9890	32.00	0.9329	35.29	0.9641	34.20	0.9571	47.98	0.9921
IDT	40.74	0.9884	32.10	**0.9344**	34.89	0.9623	33.84	0.9549	47.35	**0.9930**
Ours	**41.23**	**0.9894**	**32.17**	0.9326	**35.35**	**0.9646**	**34.35**	**0.9588**	**48.54**	0.9924

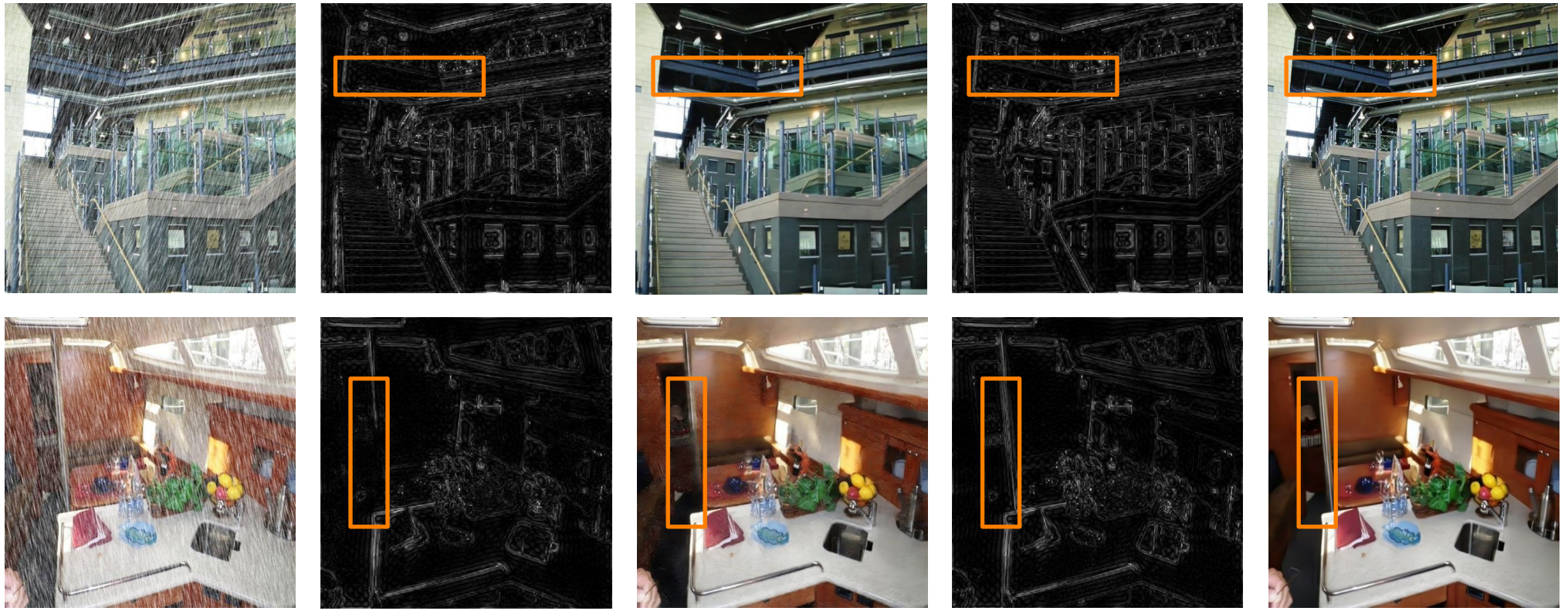
Sparse Transformer Network

- Qualitative evaluations



Sparse Transformer Network

- Is sparse attention effective?

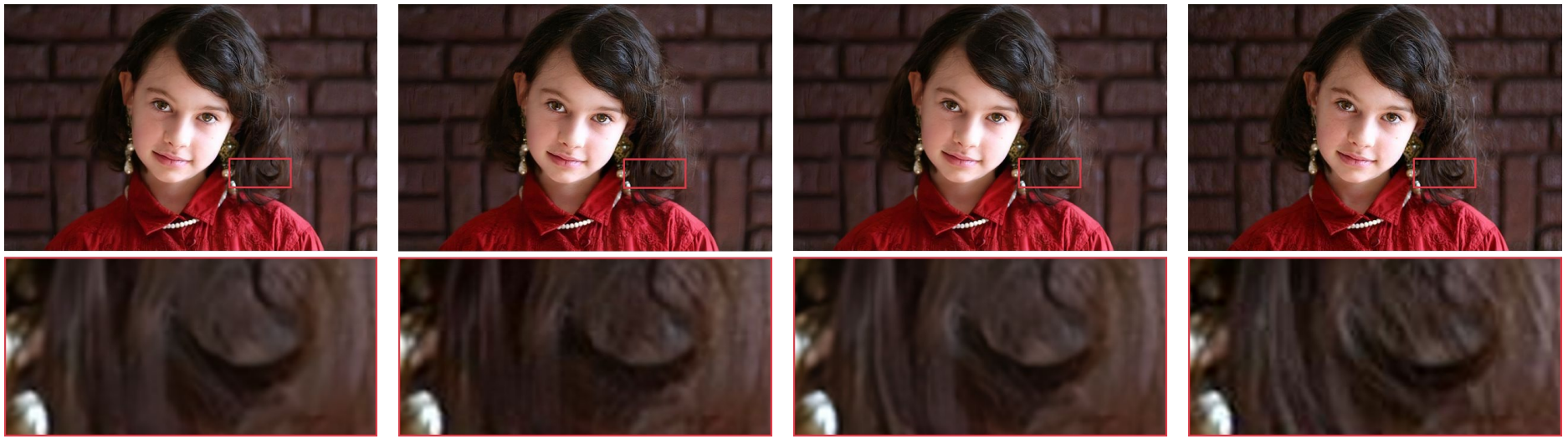


Rainy Input | w/o Top-k | w Top-k

Sparse Transformer Network

- **Does mixed-scale strategy help features refinement?**

Models	Feed-forward Network (FN)	Dconv FN (DFN)	Gated-Dconv FN (GDFN)	Mixed-Scale FN (MSFN)
PSNR / SSIM	31.84 / 0.9275	31.88 / 0.9279	31.97 / 0.9286	**32.18 / 0.9330**



FN DFN GDFN MSFN

Sparse Transformer Network

- **Limitations**
  - **Model complexity is relatively high**

Comparison of complexity on a 256 × 256 image

Methods	MSPFN	MPRNet	Uformer	Restormer	IDT	Ours
FLOPs (G)	595.5	175.8	45.9	174.7	61.8	242.9

  - **The proposed model is less efficient**

Methods	MSPFN	MPRNet	Uformer	Restormer	IDT	Ours
Parameters (M)	13.4	3.63	50.8	26.1	16.4	33.7
Run-times (s)	0.41	0.10	0.19	0.28	0.28	0.29

Conclusion

- Not all non-local information helps image restoration
- Adaptively learning most useful information from non-local patches (tokens) help image restoration

Thanks!

https://github.com/cschenxiang/DRSformer