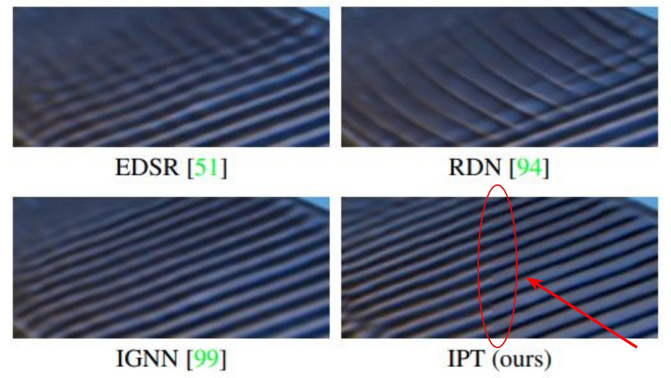
MAXIM: Multi-Axis MLP for Image Processing

Zhengzhong Tu[1] Hossein Talebi[2] Han Zhang[2] Feng Yang[2] Peyman Milanfar[2] Alan Bovik[1] Yinxiao Li[2]
[1] University of Texas at Austin [2] Google Research

Problem Statement

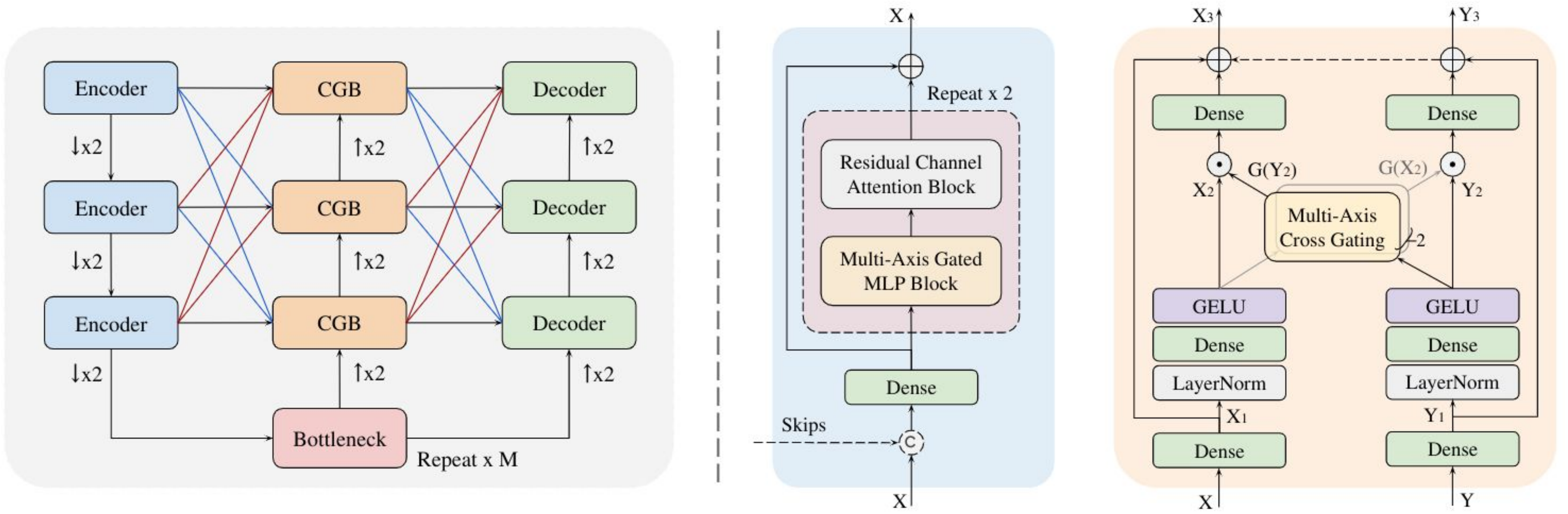
Develop efficient Transformer/MLPs for low-level vision

- low-level vision like **denoising**, **deblurring**, **dehazing**, etc. requires **high-resolution** image-to-image processing
- Vision Transformers/MLPs are promising on high-level tasks, but it is **non-trivial to adapt** them to low-level (image processing) problems
- The model needs to be '**fully-convolutional**', i.e., train on small patches and **inference on full resolution**. Otherwise, the model will cause patch-boundary artifacts [R1]:



[R1] Pre-Trained Image Processing Transformer, CVPR 21, arxiv.org/abs/2012.00364

Our Method: MAXIM Architecture

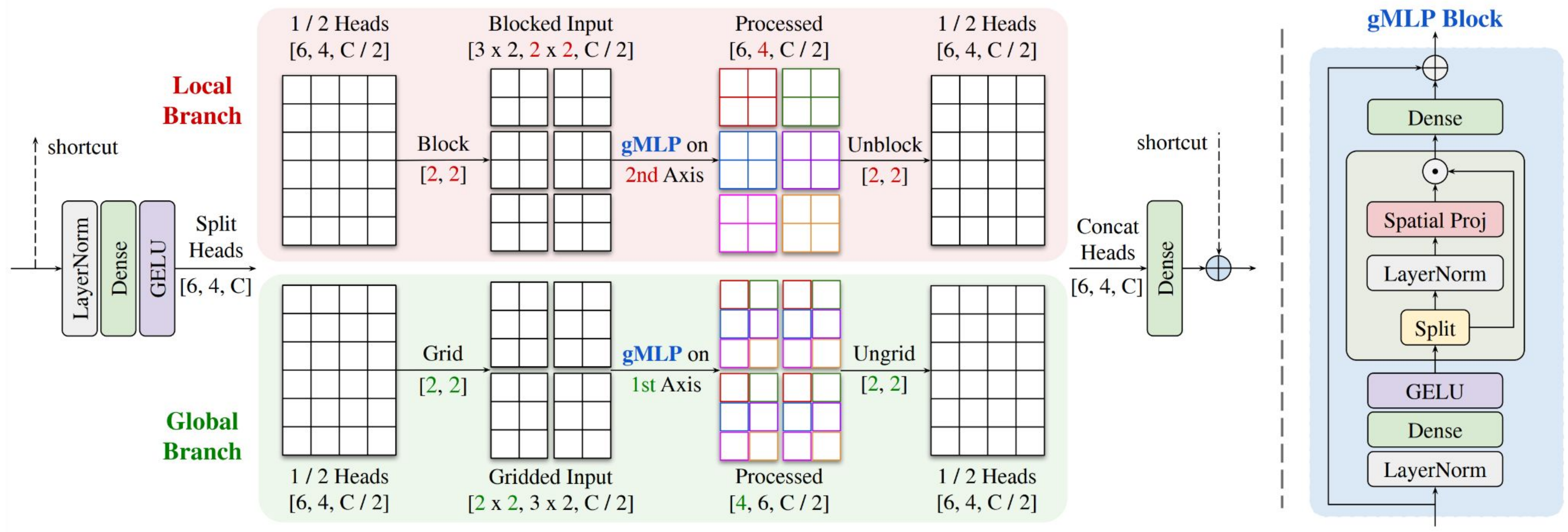


(a) MAXIM Backbone (b) Encoder / Decoder / Bottleneck (c) Cross Gating Block (CGB)

Our proposed MAXIM model is:

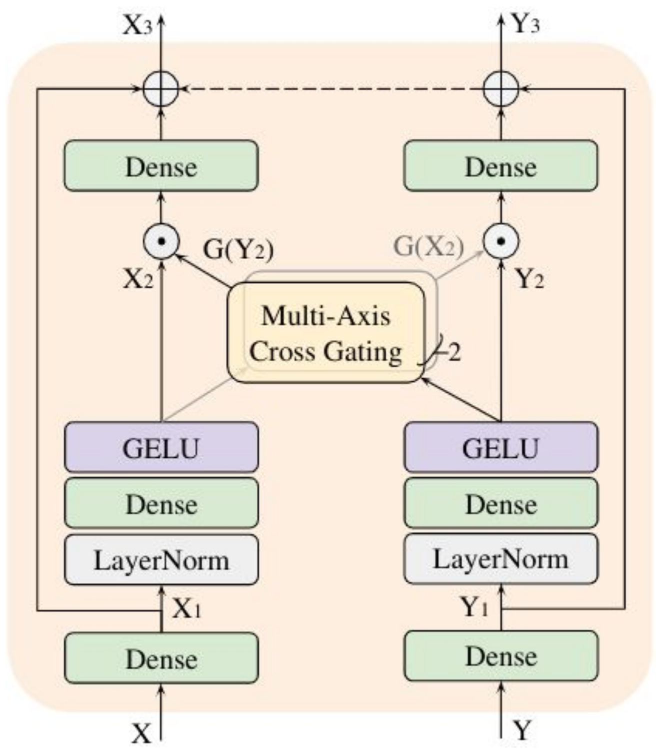
- A global **UNet**-like architecture, with multi-stage stacks
- **Every** block enjoys **global-local spatial interaction**
- 'Fully-convolutional', i.e., can be trained on small patches and **directly applied on any high resolution** (w/o causing patch-boundary effects)
- scales linearly w.r.t. input image size, unlike other MLP models

Core Module 1: Multi-Axis Gated MLP Nlock



- Contains local branch (2nd Axis) and global branch (1st Axis)
- Apply gMLP on one axis each time in either one branch
- Global and 'fully-convolutional' with linear complexity
- Standalone module that can be plugged-in in many networks

Core Module 2: Cross-Gating MLP Block

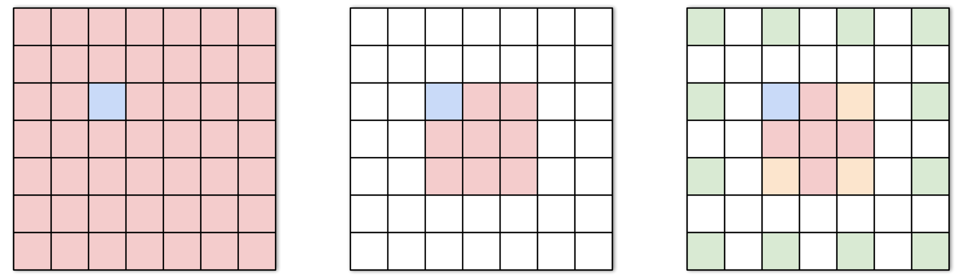


- Same design to core module 1, but extending to interact two features
- G(.) function obtains multi-axis gating signals only, and gating is applied reciprocally with two features:

$$\hat{\mathbf{X}} = \mathbf{X}_2 \odot \mathrm{G}(\mathbf{Y}_2), \quad \hat{\mathbf{Y}} = \mathbf{Y}_2 \odot \mathrm{G}(\mathbf{X}_2)$$

- Can be used as a conditioning layer or fusion module
- Also global and 'fully-convolutional' with linear complexity

Model	Complexity	Fully-conv	Global
MLP-Mixer [36]	$\mathcal{O}(N^2)$	✗	✓
gMLP [19]	$\mathcal{O}(N^2)$	✗	✓
Swin-Mixer [22]	$\mathcal{O}(N)$	✓	✗
MAXIM (ours)	$\mathcal{O}(N)$	✓	✓



Mixer/gMLP Swin-Mixer MAXIM (ours)

Numerical results

Evaluated on **5 low-level** tasks, **SoTA** on **15 out of 20** datasets

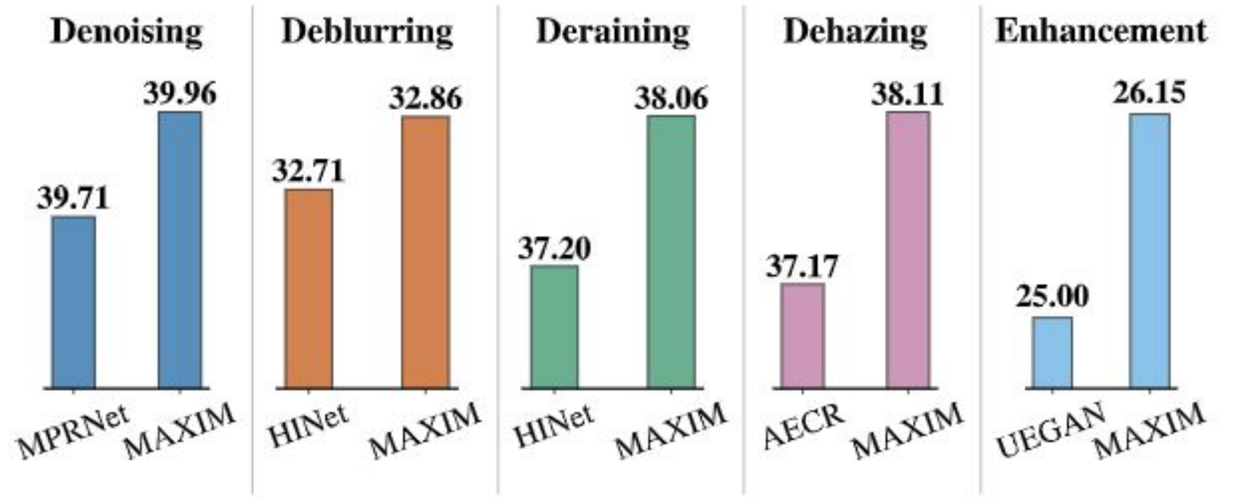


Figure 1. Our proposed MAXIM model significantly advances state-of-the-art performance on five image processing tasks in terms of PSNR: 1) Denoising (+**0.24** dB on SIDD [1]), 2) Deblurring (+**0.15** dB on GoPro [57]) 3) Deraining (+**0.86** dB on Rain100L [95]), 4) Dehazing (+**0.94** dB on RESIDE [43]), and 5) Retouching (Enhancement) (+**1.15** dB on FiveK [6]).

- State of the Art: Deblurring on HIDE (trained on GOPRO)
- Ranked #8: Deblurring on GoPro
- State of the Art: Deblurring on RealBlur-J
- Ranked #2: Deblurring on RealBlur-R
- Ranked #4: Deblurring on RealBlur-J (trained on GoPro)
- Ranked #5: Deblurring on RealBlur-R (trained on GoPro)
- Ranked #3: Low-Light Image Enhancement on LOL
- State of the Art: Photo Retouching on MIT-Adobe 5k
- Ranked #2: Single Image Deraining on Rain100H
- Ranked #3: Single Image Deraining on Rain100L
- Ranked #2: Single Image Deraining on Test100
- Ranked #3: Single Image Deraining on Test2800
- Ranked #5: Single Image Deraining on Test1200
- Ranked #4: Image Denoising on SIDD
- Ranked #5: Image Denoising on DND

Visual results → check more @arxiv.org/abs/2201.02973

Check out Web demo! @replicate Paper Code Tweet Zhihu