Efficient Test-Time Adaptation for Super-Resolution with Second-Order Degradation and Reconstruction

Zeshuai Deng[1*], Zhuokun Chen[1 2*], Shuaicheng Niu[5*], Thomas H. Li[6],
Bohan Zhuang[3†], Mingkui Tan[1 2 4†]

[1]South China University of Technology, [2]Pazhou Lab, [3]ZIP Lab, Monash University,
[4]Key Laboratory of Big Data and Intelligent Robot, Ministry of Education,
[5]Nanyang Technological University, [6]Peking University Shenzhen Graduate School

NeurIPS 2023

Code: https://github.com/DengZeshuai/SRTTA

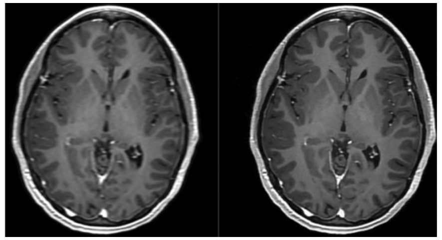
Outline

- **Background**
- **Methodology**
- **Experimental Results**
- **Conclusion**

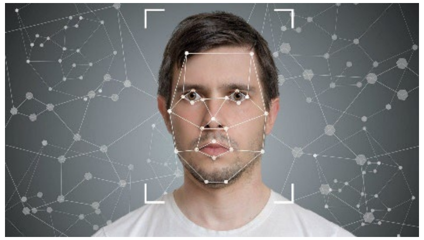
Problem Definition

Super Resolution

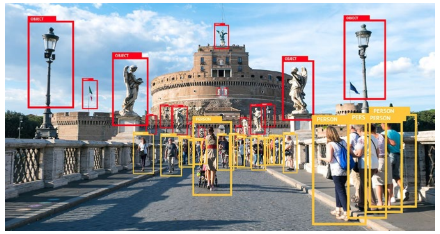
Input: Low-resolution image **Output: High-resolution image**



Medical Analysis



Face Recognition

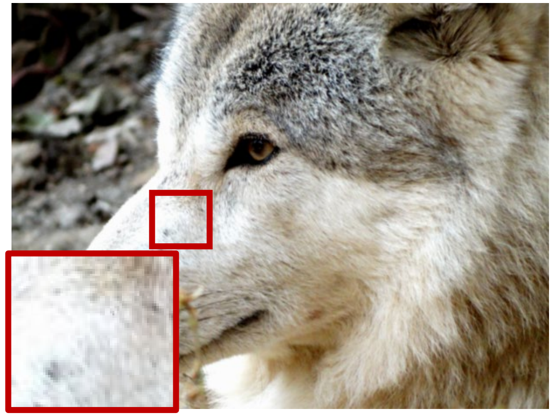


Object Recognition

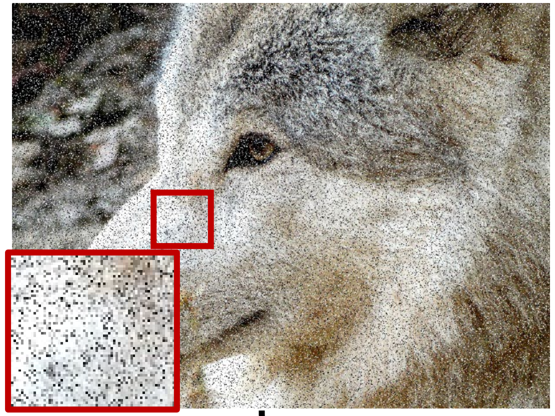
Video Restoration

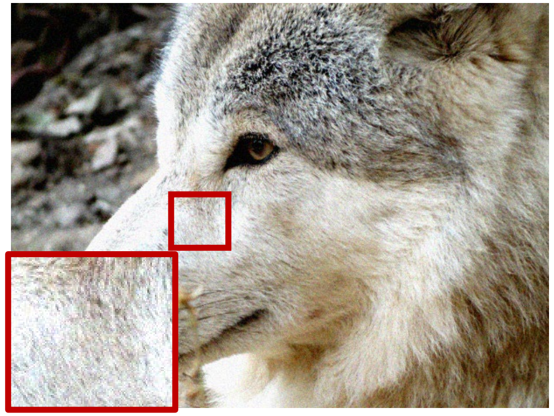
……

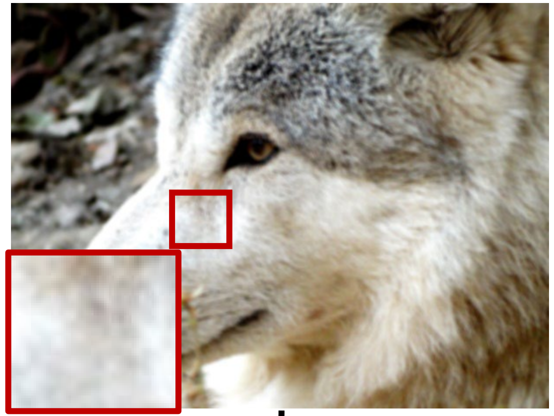
Dynamically changing domain shift



Training







Testing

Domain shift vs. Degradation Shift

Domain Shift

- Domain shift refers to the change in data distribution between training and testing

$$\mathcal{D}_{training} \neq \mathcal{D}_{testing} \tag{1}$$

Degradation Shift

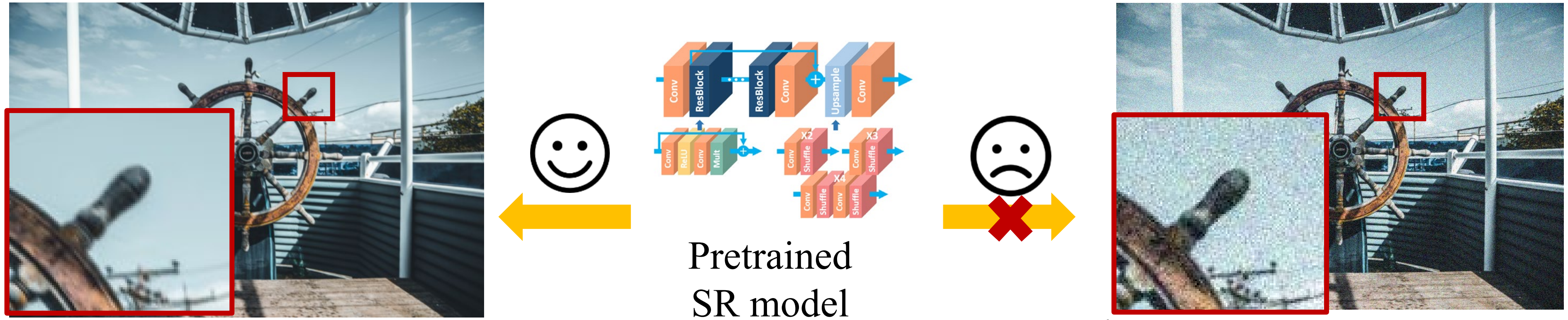
- The degradation process of real-world test images can be modeled by a classical degradation model $D(\cdot)$. It can be defined by:

$$\mathbf{x} = \mathbf{D}(\mathbf{y}) = [(\mathbf{y} \otimes \mathbf{k}) \downarrow_s + \mathbf{n}]_{JPEG_q} \tag{2}$$

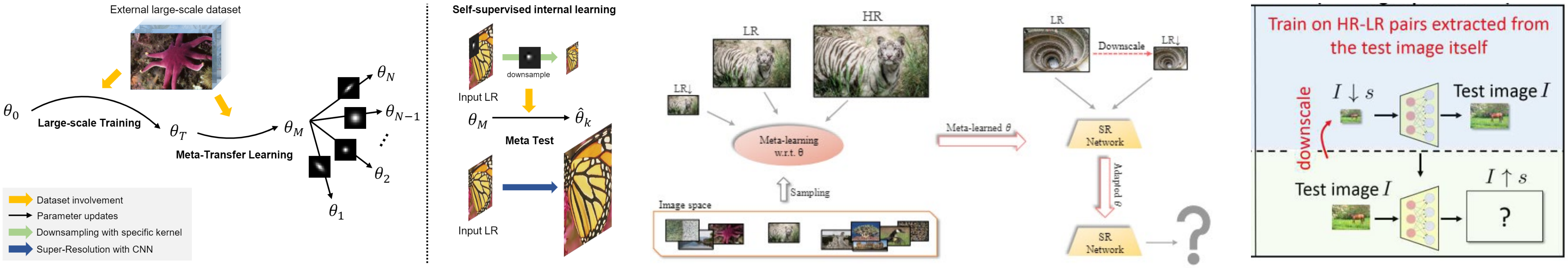
- where $\otimes$ denotes the convolution operation, $\downarrow_s$ denotes the downsampling with a scale factor of s, and $JPEGq$ denotes the JPEG compression with the quality factor q

Motivations

- Real-world images may exhibit **various degradation types** due to diverse imaging sensors and multiple Internet transmissions, **limiting the performance of pre-trained SR models**



- It is hard to **quickly adapt to dynamically changing domain (degradation shift)**



MZSR **MLSR** **ZSSR**

work bad / **time-consuming**

Rethinking

- Existing SR methods suffer from two key limitations: **low efficiency** and **narrow focus on a single degradation type**

1. **How to quickly adapt to unknown domain during test-time?**

2. **How to design a generalized test-time learning framework?**

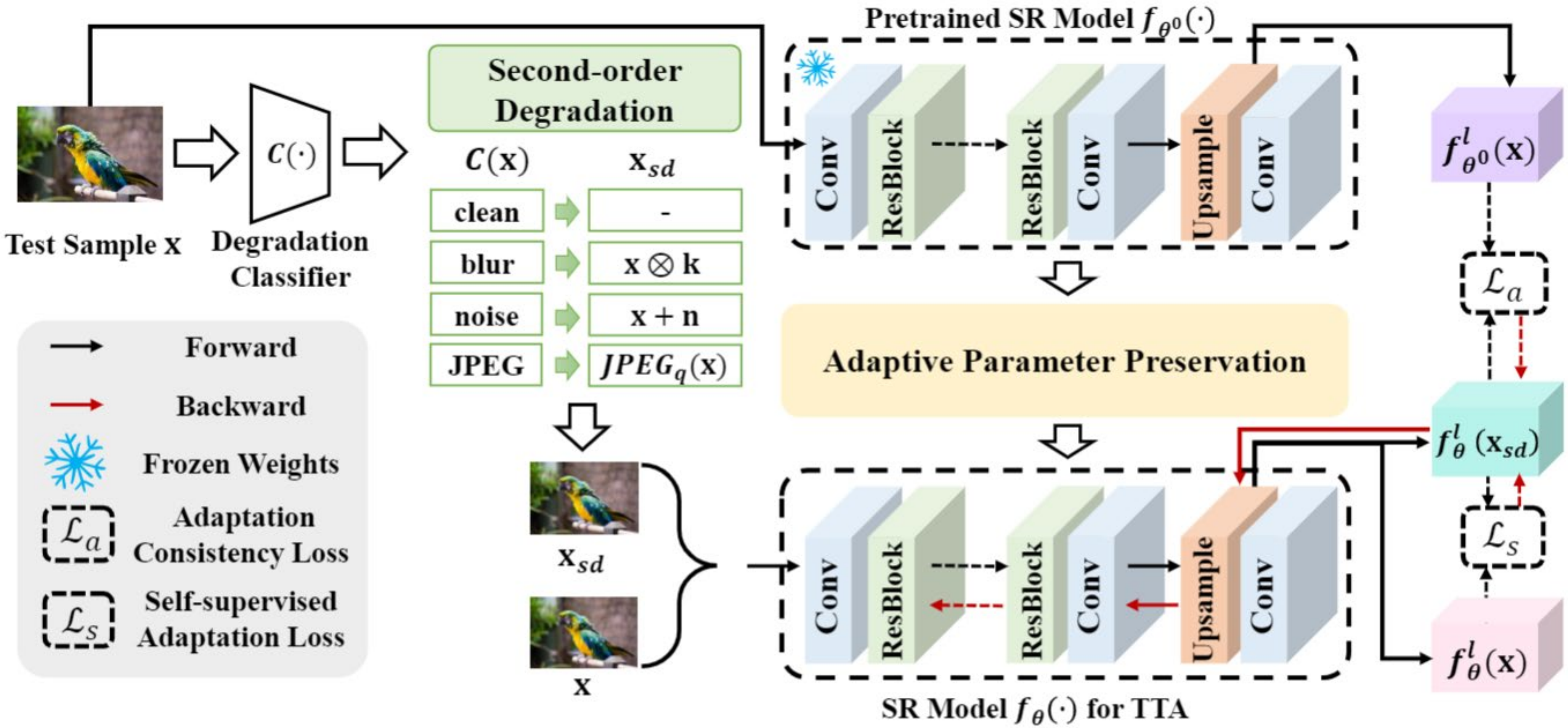
Outline

- Background
- **Methodology**
- Experimental Results
- Conclusion

SRTTA Framework

Our method

- We propose a super-resolution test-time adaptation framework (SRTTA) to adapt a trained super-resolution model to target domains with unknown degradation



SRTTA Pipeline

Algorithm 1: The pipeline of the Super-Resolution Test-Time Adaptation.

Input: Real-world test images $\{\mathbf{x}_t\}_{t=1}^{T}$, adaptation iteration steps S for each image, learning rate η, batch size N, preservation ratio ρ.

Load the pretrained SR models $f_{\theta^0}(\cdot)$ and the degradation classifier $C(\cdot)$.

Select and freeze the important parameters using Eqn. (9) with ρ.

Our APP strategy

for $\mathbf{x}_t$ *in* $\{\mathbf{x}_t\}_{t=1}^{T}$ **do**

  for s *in* $\{1, 2, ..., S\}$ **do**

    Construct paired data $\{\mathbf{x}_{sd}^{i}, \mathbf{x}_t\}_{i=1}^{N}$ based on $C(\mathbf{x}_t)$ using Eqn. (3);

    Adapt the SR model using Eqn. (6) with η;

  end

end

Test-time adaptation

Output: The adapted SR model f_{θ}, the predictions $\{\hat{\mathbf{y}}_t = f_{\theta}(\mathbf{x}_t)\}_{t=1}^{T}$ for all $\mathbf{x}_t$ in $\{\mathbf{x}_t\}_{t=1}^{T}$.

- We use a pre-trained degradation classifier to predict the degradation type $C(x)$ of the test image
- We construct a set of paired data using (**Second-order Degradation scheme**) and adapt the SR model with our **Second-Order Reconstruction loss**
- We directly use the frozen pre-trained SR model when test samples are clean images

Motivation of Second-order Degradation

How to quickly identify the type of degradation?

- Existing methods **narrow focus** on blur degradation
- We use a **pre-trained degradation classifier** to quickly recognize more degradation types

How to quickly construct paired data to adapt the SR model the target domain?

- Existing methods often precisely estimate the parameters of the degradation to construct the paired data, which is time-consuming
- We randomly generate parameters of degradations, avoiding estimating degradations

Second-Order Degradation Scheme

Second-Order Degradation

- Construct a set of second-order degraded images x_{sd} using Eqn. (3) according to the prediction results of the pre-trained degradation classifier

$$\begin{gathered}\mathbf{x}_{sd} = D(\mathbf{x}, C(\mathbf{x})) = D_j(D_b(\mathbf{x}, c_b) + D_n(c_n), c_j), \\ D_b(\mathbf{x}, c_b) = c_b(\mathbf{x} \otimes \mathbf{k}) + (1 - c_b)\mathbf{x},\;\; D_n(c_n) = c_n \mathbf{n}, \\ D_j(\mathbf{x}, c_j) = c_j JPEG_q(\mathbf{x}) + (1 - c_j)\mathbf{x},\end{gathered} \tag{3}$$

- $\boldsymbol{k}$ denotes a **random** blur kernel, $\boldsymbol{n}$ denotes a **random** noise map, q denotes a **random** quality factor
- Prediction results of the degradation classifier are denoted by c_b, c_n and $c_j \in \{0, 1\}$

Adaptation with Second-Order Reconstruction

Self-supervised adaptation

- Adapt the pre-trained model to remove the degradation using Eqn. (4)

$$\mathcal{L}_s(\mathbf{x}, \mathbf{x}_{sd}) = \sqrt{(f^l_\theta(\mathbf{x}) - f^l_\theta(\mathbf{x}_{sd}))^2 + \epsilon} \tag{4}$$

- $f^l_\theta(\cdot)$ denotes the output features of the l_{th} layer of the pre-trained SR model

Consistency maximization

- Keep the model consistent across adaptation using Eqn. (5)

$$\mathcal{L}_a(\mathbf{x}, \mathbf{x}_{sd}) = \sqrt{(f^l_{\theta^0}(\mathbf{x}) - f^l_\theta(\mathbf{x}_{sd}))^2 + \epsilon} \tag{5}$$

- $f^l_{\theta^0}(\cdot)$ denotes the output features of the l_{th} layer of the pre-trained SR model

Second-order reconstruction

- Keep the model consistent across adaptation using Eqn. (6)

$$\mathcal{L} = \mathcal{L}_s(\mathbf{x}, \mathbf{x}_{sd}) + \alpha \mathcal{L}_a(\mathbf{x}, \mathbf{x}_{sd}) \tag{6}$$

Adaptive Parameter Preservation for Anti-Forgetting

Diagonal Fisher information matrix

- Evaluating the importance of each parameters using Eqn. (7) and Eqn. (8)

$$\omega(\theta_i^0) = \frac{1}{|\mathcal{D}_c|} \sum_{\mathbf{x}_c \in \mathcal{D}_c} \left(\frac{\partial \mathcal{L}_c(\mathbf{x}_c)}{\partial \theta_i^0}\right)^2 \tag{7}$$

$$\mathcal{L}_c(\mathbf{x}_c) = \sqrt{(\bar{\mathbf{y}} - f_{\theta^0}(\mathbf{x}_c))^2 + \epsilon}, \;\; s.t. \;\; \bar{\mathbf{y}} = \frac{1}{8} \sum_{i=1}^{8} \mathbf{R}_i(f_{\theta^0}(\mathbf{A}_i(\mathbf{x}_c))) \tag{8}$$

- D_C denotes a set of clean images, $A_i \in \{A_j\}_{j=1}^{8}$ denotes an augmentation operation, R_i denotes the inverse operation of A_i

Important Parameter Selection

- Select the important parameters using Eqn. (9)

$$\mathcal{S} = \{\theta_i^0 | \omega(\theta_i^0) > \tau_\rho, \theta_i^0 \in \theta^0\} \tag{9}$$

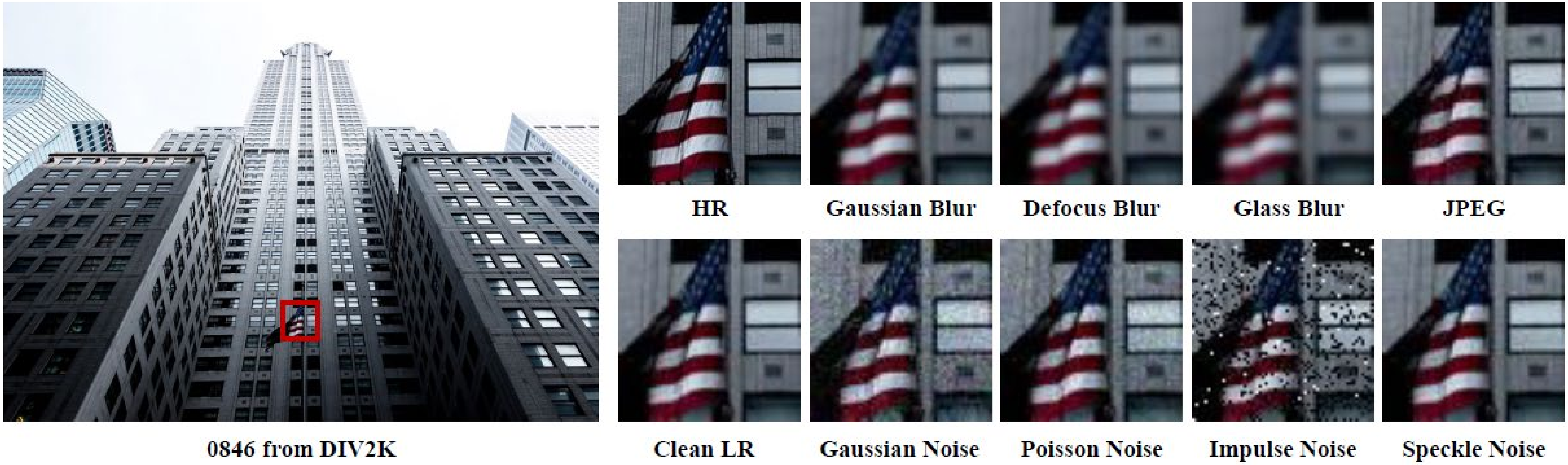
- τ_ρ denotes the first ρ-ratio largest value obtained by ranking the value $\omega(\theta_i^0)$, ρ is a hyperparameter to control the ratio of parameters to be frozen
- We only select the set of significant parameters S **once**

Outline

- Background
- Methodology
- **Experimental Results**
- Conclusion

Synthesized Dataset

- A new dataset named DIV2K-C consists of **eight** kinds of test images, which includes different degradations



- Another synthesized dataset named DIV2K-MC consists of **four** kinds of test images, which are synthesized with different **mixed** degradations, including BlurNoise, BlurJPEG, NoiseJPEG and BlurNoiseJPEG

Comparison with SOTA on DIV2K-C

Table A: We report the PSNR/SSIM results of all corruption fields in DIV2K-C for 2× SR.

Methods	GaussianBlur	DefocusBlur	GlassBlur	GaussianNoise	PossionNoise	ImpulseNoise	SpeckleNoise	JPEG	Mean	GPU Time
	PSNR/SSIM	PSNR/SSIM	PSNR/SSIM	PSNR/SSIM	PSNR/SSIM	PSNR/SSIM	PSNR/SSIM	PSNR/SSIM	PSNR/SSIM	(seconds/image)
Bicubic	28.04/0.803	24.10/**0.784**	26.31/0.745	25.35/0.554	23.33/0.496	15.28/0.324	28.65/0.774	28.28/0.806	24.92/0.661	-
SwinIR [33]	30.40/0.838	25.52/0.673	27.82/0.773	25.35/0.510	22.36/0.428	15.34/0.242	30.45/0.774	30.74/0.846	26.00/0.636	13.08
IPT [9]	28.93/0.820	24.08/0.640	26.39/0.749	22.96/0.439	20.08/0.369	13.06/0.241	28.27/0.728	28.36/0.804	24.02/0.599	55.36
HAT [10]	29.00/0.821	24.08/0.640	26.40/0.749	22.31/0.417	19.33/0.349	11.91/0.192	28.02/0.722	28.25/0.802	23.66/0.587	25.01
DAN [24]	**34.32/0.916**	25.58/0.673	31.77/0.872	26.36/0.558	23.28/0.461	11.46/0.203	30.64/0.777	31.08/0.857	26.81/0.665	3.10
DCLS-SR [38]	33.93/0.914	25.55/0.671	**31.98/0.872**	25.45/0.521	21.59/0.415	8.12/0.112	30.66/0.784	30.86/0.848	26.02/0.642	1.45
ZSSR [48]	29.91/0.831	25.54/0.674	27.79/0.771	26.79/0.590	24.24/0.509	**19.14/0.375**	30.95/0.813	31.01/0.853	26.92/0.677	117.65
KernalGAN [2]+ZSSR	30.18/0.859	**25.87**/0.679	29.01/0.808	21.45/0.436	19.32/0.366	17.93/0.354	25.07/0.686	26.11/0.774	24.37/0.620	231.41
MZSR [11]	30.14/0.838	25.54/0.670	28.03/0.777	25.94/0.543	23.48/0.472	17.05/0.314	30.00/0.771	30.49/0.845	26.33/0.654	3.34
DualSR [14]	29.00/0.854	24.40/0.640	28.18/0.805	22.30/0.509	20.11/0.436	17.22/0.376	24.99/0.738	24.74/0.751	23.87/0.639	210.85
DDNM [56]	28.46/0.808	24.09/0.636	26.39/0.744	24.37/0.497	21.92/0.432	13.98/0.310	28.60/0.753	28.26/0.802	24.51/0.623	2,288.55
EDSR [35]	30.28/0.837	25.52/0.673	27.82/0.773	25.87/0.536	22.96/0.449	15.87/0.269	30.52/0.778	30.83/0.847	26.21/0.645	-
TTA-C	30.21/0.835	25.50/0.673	27.79/0.772	26.37/0.559	23.57/0.473	16.40/0.298	30.25/0.783	30.91/0.849	26.38/0.655	13.59
SRTTA (ours)	31.07/0.869	25.86/0.674	29.01/0.815	**29.65**/0.762	26.69/0.637	16.15/0.284	**32.33/0.873**	**31.30/0.857**	**27.76**/0.721	5.38
SRTTA-lifelong (ours)	31.07/0.869	25.83/0.674	29.18/0.819	29.48/**0.797**	**27.10/0.673**	16.27/0.273	31.71/0.864	31.22/0.853	27.73/**0.728**	5.38

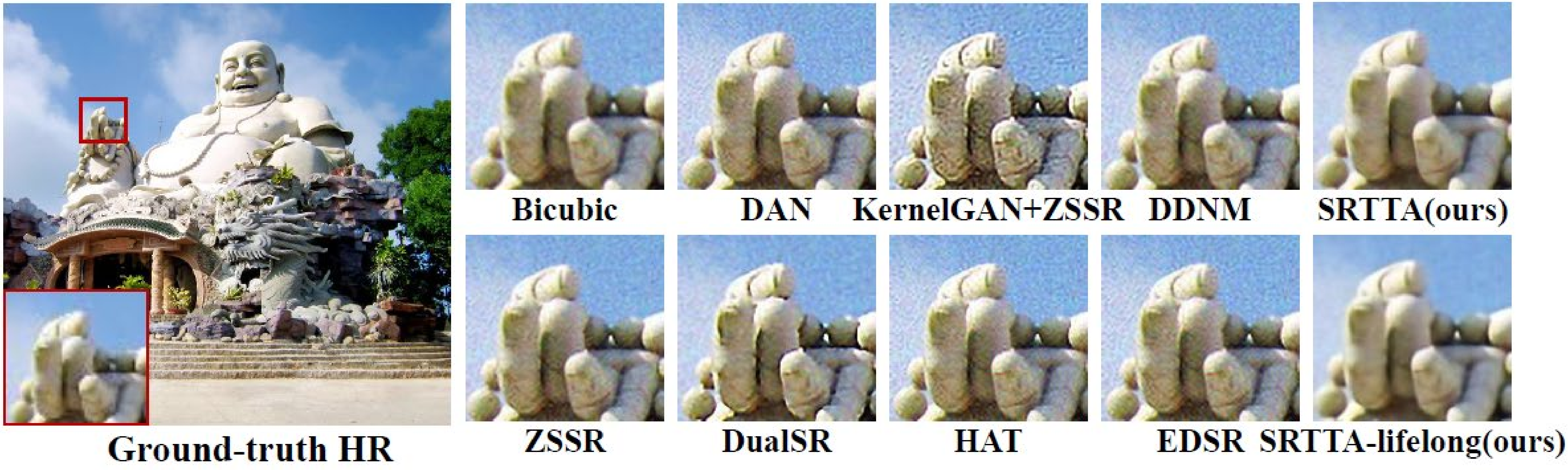
- SRTTA achieves the **best performance** in terms of PSNR and SSIM on average
- SRTTA achieves a better **tradeoff** between **performance** and **efficiency**

Comparison with SOTA on DIV2K-MC

Methods	BlurNoise	BlurJPEG	NoiseJPEG	BlurNoiseJPEG	Mean
SwinIR [15]	20.91/0.311	26.83/0.748	23.86/0.523	22.77/0.450	23.59/0.508
IPT [5]	21.28/0.327	26.83/0.748	24.15/0.535	22.96/0.459	23.81/0.517
HAT [6]	23.41/0.399	28.86/0.788	25.69/0.572	24.42/0.502	25.59/0.565
DAN [12]	24.14/0.438	28.95/0.791	26.20/0.593	24.82/0.519	26.03/0.585
DCLS-SR [18]	23.84/0.420	28.93/0.790	26.37/0.599	24.92/0.523	26.02/0.583
ZSSR [20]	24.95/0.493	29.02/0.793	26.68/0.617	25.24/0.542	26.47/0.611
KernelGAN [1]+ZSSR	23.08/0.424	28.32/0.786	21.90/0.474	22.76/0.443	24.02/0.532
MZSR [7]	18.73/0.213	24.90/0.667	20.37/0.398	20.62/0.354	21.16/0.408
DualSR [8]	25.59/0.561	28.24/0.787	23.78/0.586	24.62/0.541	25.56/0.619
DDNM [24]	22.62/0.389	26.82/0.746	25.11/0.582	23.81/0.504	24.59/0.555
EDSR [16]	24.02/0.430	28.93/0.790	26.08/0.587	24.73/0.514	25.94/0.580
TTA-C	24.29/0.446	28.93/0.790	26.35/0.598	24.91/0.522	26.12/0.589
SRTTA (ours)	26.93/0.709	28.93/**0.797**	**29.13/0.784**	27.12/0.728	28.03/0.755
SRTTA-lifelong (ours)	**27.67/0.749**	**29.02**/0.793	29.70/0.810	**27.52/0.747**	**28.48/0.775**

- SRTTA achieves the **best performance** in terms of PSNR and SSIM on average on DIV2K-MC

Visual comparison on DIV2K-C



(a) Visualizations under Gaussian Noise for 2× SR

- Our SRTTA is able to reduce the effect of degradation and generate **more plausible HR images**

Visual comparison on real-world images

Visual comparison for 2× SR on real-world images

- Our SRTTA is able to generate HR images with **fewer artifacts**
- These results demonstrate that our method is able to be **applied to real-world** applications

Visual comparison on real-world images

Visual comparison for 2× SR on real-world images

- Our SRTTA is able to generate HR images with **fewer artifacts**
- These results demonstrate that our method is able to be **applied to real-world** applications

Outline

- **Background**
- **Methodology**
- **Experimental Results**
- **Conclusion**

Conclusion

Conclusion

- We propose a super-resolution test-time adaptation (SRTTA) framework to adapt any pre-trained SR models to unknown target domains during the test time
- We use a pre-trained classifier to identify the degradation type for a test image and construct the paired data using our second-order degradation scheme
- We construct new test datasets named DIV2K-C and DIV2K-MC, which contain eight common degradations, to evaluate the practicality of different SR methods

Code: https://github.com/DengZeshuai/SRTTA