Super-Resolution Neural Operator

Min Wei, Xuesong Zhang

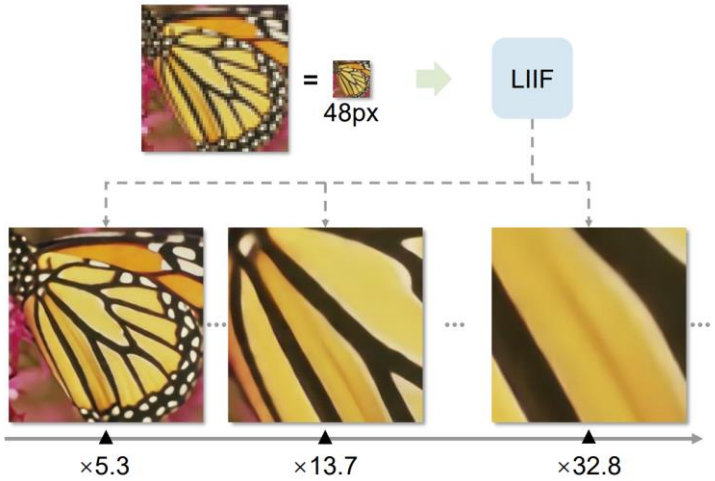
Beijing University of Posts and Telecommunications

THU-AM-169

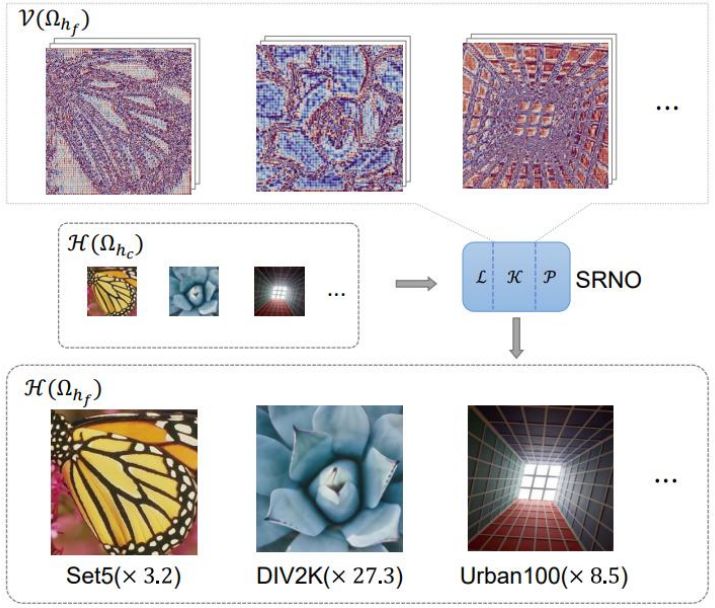
Introduction

- Treating images as implicit functions.
- Learning a implicit function space shared by different instances.
- Neural Operator learn mappings between implicit function spaces.

From:



To:



Super-Resolution Neural Operator, Min Wei, Xuesong Zhang.

What is Neural Operator (NO) ?

- Neural Operator (NO) was proposed for discretization invariant solutions of PDEs via operator learning by the following abstract form:

$$(\mathbf{L}_a u)(x) = f(x), \quad x \in D,$$
$$u(x) = 0, \quad x \in \partial D,$$

- NO seeks a feasible operator $\mathcal{G}: \mathcal{A} \to \mathcal{U}, a \mapsto u$, directly mapping the coefficient to the solution within an acceptable tolerance.
- And it can be formulated as follows:

$$z_0(x) = \mathcal{L}(x, a(x)),$$
$$z_{t+1}(x) = \sigma(W_t z_t(x) + (\mathcal{K}_t(z_t; \Phi))(x)),$$
$$u(x) = \mathcal{P}(z_T(x)),$$

Super-Resolution Neural Operator, Min Wei, Xuesong Zhang.

Why NO in SR ?

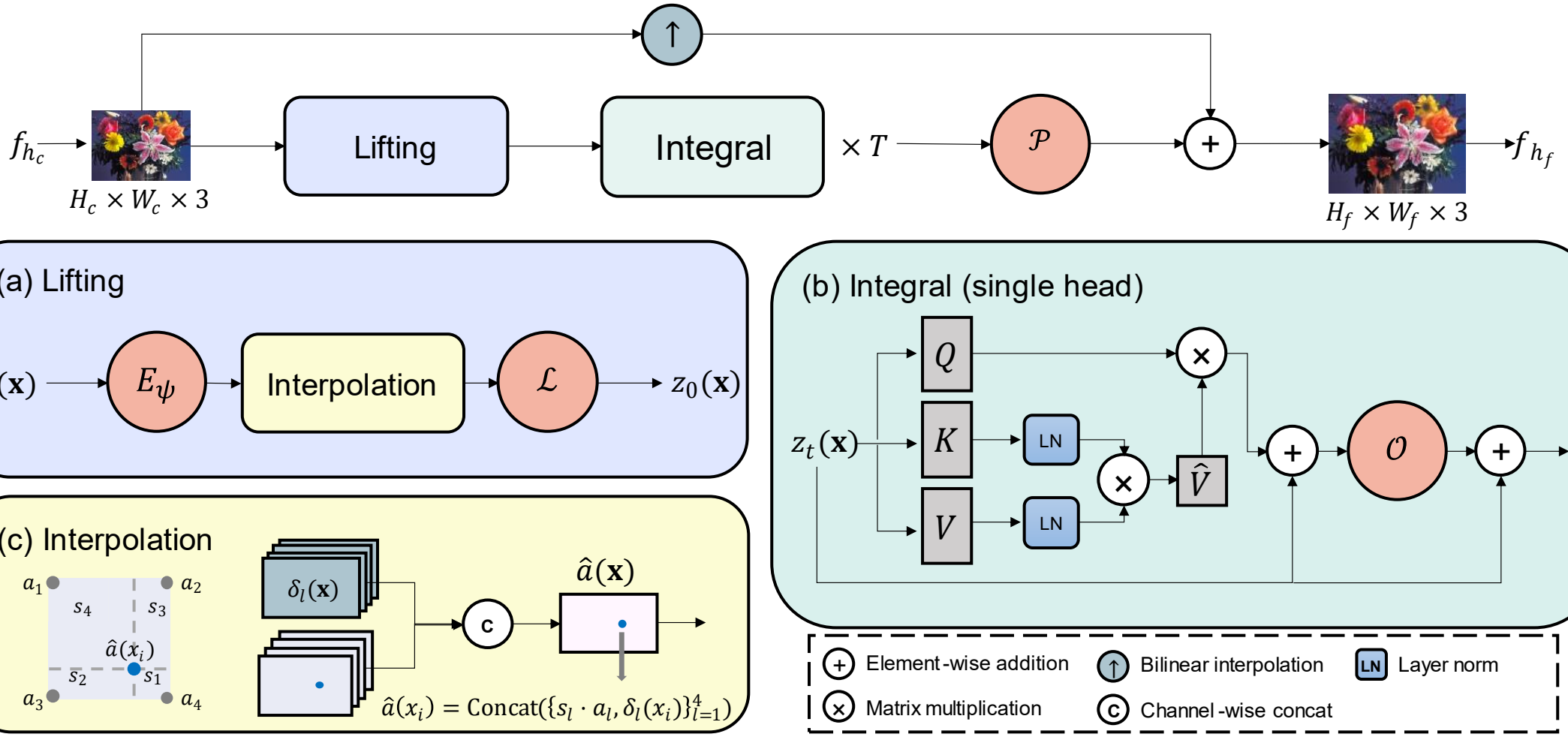
Current limitations in recent implicit function works:

- MLP's *spectral bias* results in limited performance when decoding high-frequency components.
- A local shared function is hard to capture *global correlations*.

NO provides a mathematical framework that can direct neural network design to maintain the continuum in the spatial domain.

NO's kernel integrals can explicitly capture the global relationship constraining the underlying solution function.

Super-Resolution Neural Operator (SRNO)



Super-Resolution Neural Operator (SRNO)

- **Lifting:** concatenate the weighted features to reduce the blocky artifacts.

$$z_0(x) = \mathcal{L}(\mathbf{c}, \{s_l \cdot E_\psi(a(\hat{x}_l)), \delta_l(x)\}_{l=1}^4),$$

- **Kernel integral:** employ the Galerkin-type attention operator with a linear complexity $O(n_{h_f} d_z^2)$.

$$(\mathcal{K}(z))(x) = \int_\Omega K(z(x), z(y)) z(y) \mathrm{d}y$$
$$\approx \sum_{i=1}^{n_{h_f}} K(z(x), z_i) z_i, \quad \forall x \in \Omega_{h_f},$$

where

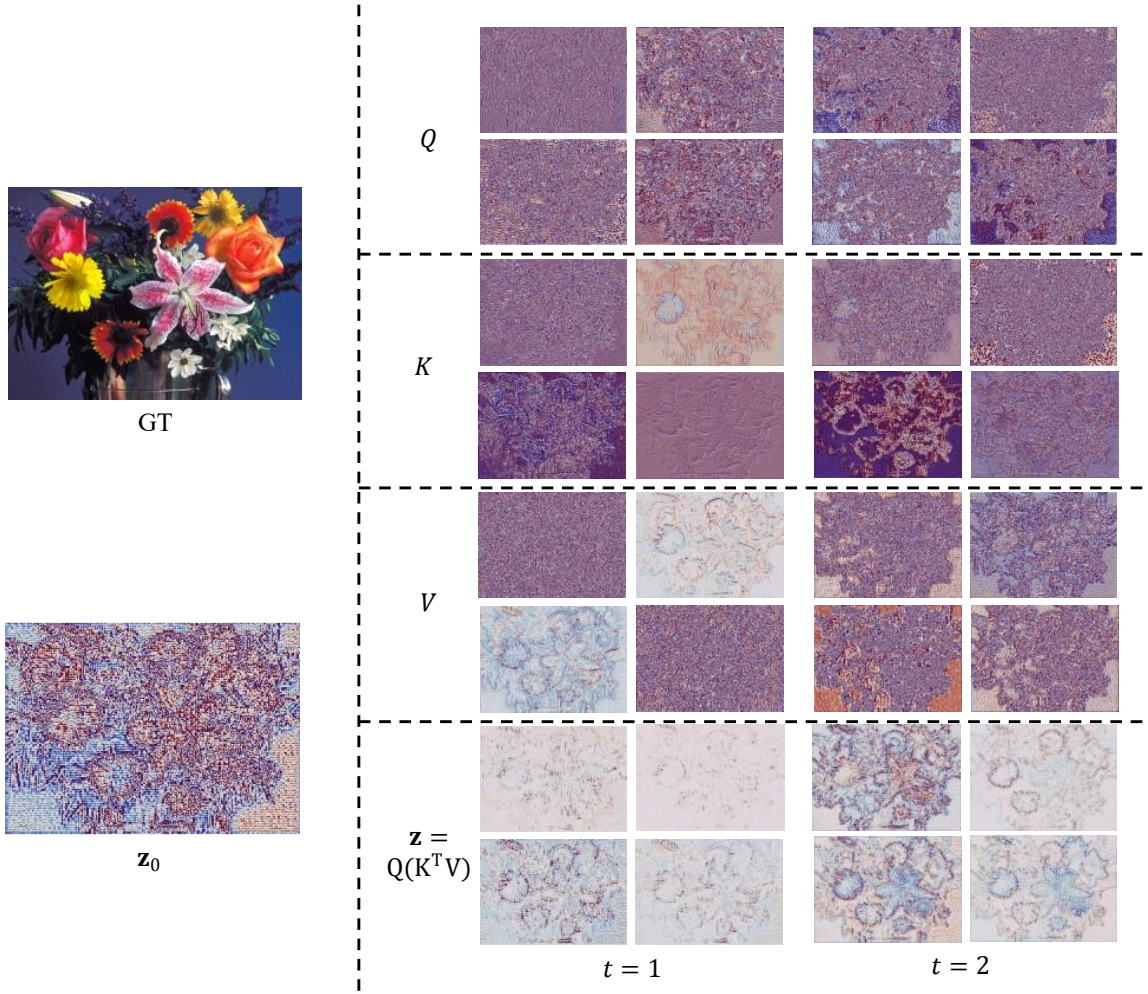
$$K(z(x), z_i) = \frac{\exp\left(\frac{\langle W_q z(x), W_k z_i \rangle}{\sqrt{d_z}}\right)}{\sum_{j=1}^{n_{h_f}} \exp\left(\frac{\langle W_q z_j, W_k z_i \rangle}{\sqrt{d_z}}\right)} W_v.$$

v.s.

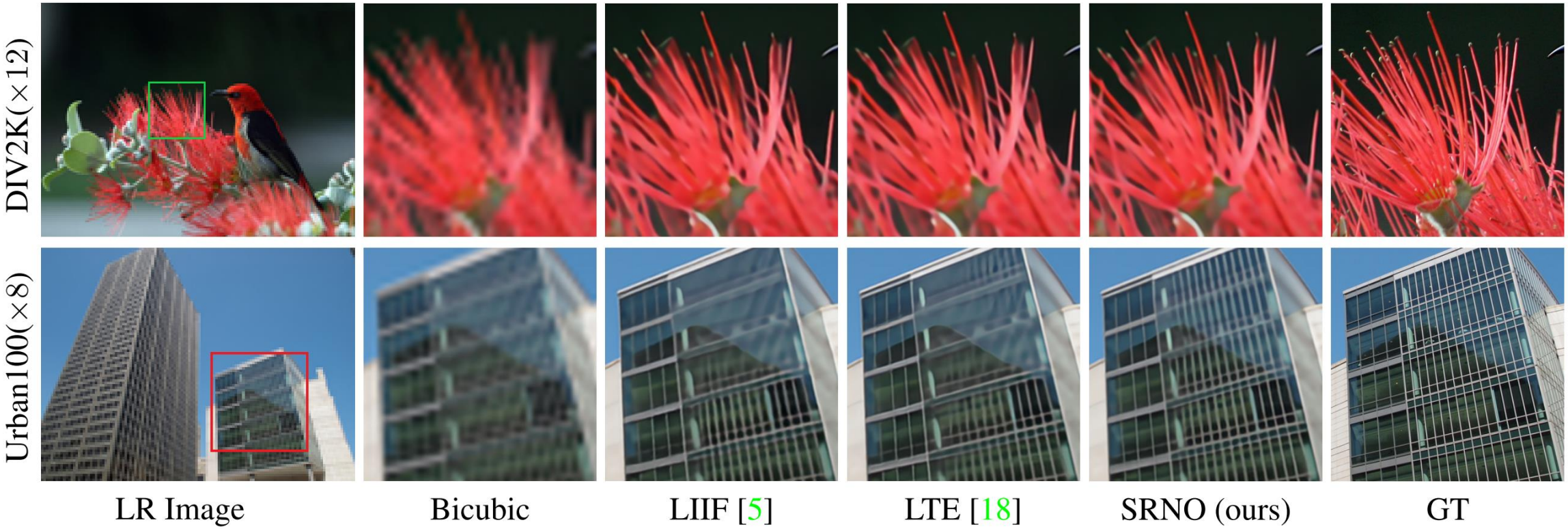
$$((\mathcal{K}(z))(x))_j = \sum_{l=1}^{d_z} \langle k_l, v_j \rangle q_l(x)$$
$$\approx \sum_{l=1}^{d_z} \left(\int_\Omega k_l(y) v_j(y) \mathrm{d}y \right) q_l(x), \quad \forall x \in \Omega_{h_f}.$$

Super-Resolution Neural Operator, Min Wei, Xuesong Zhang.

Dynamic basis update

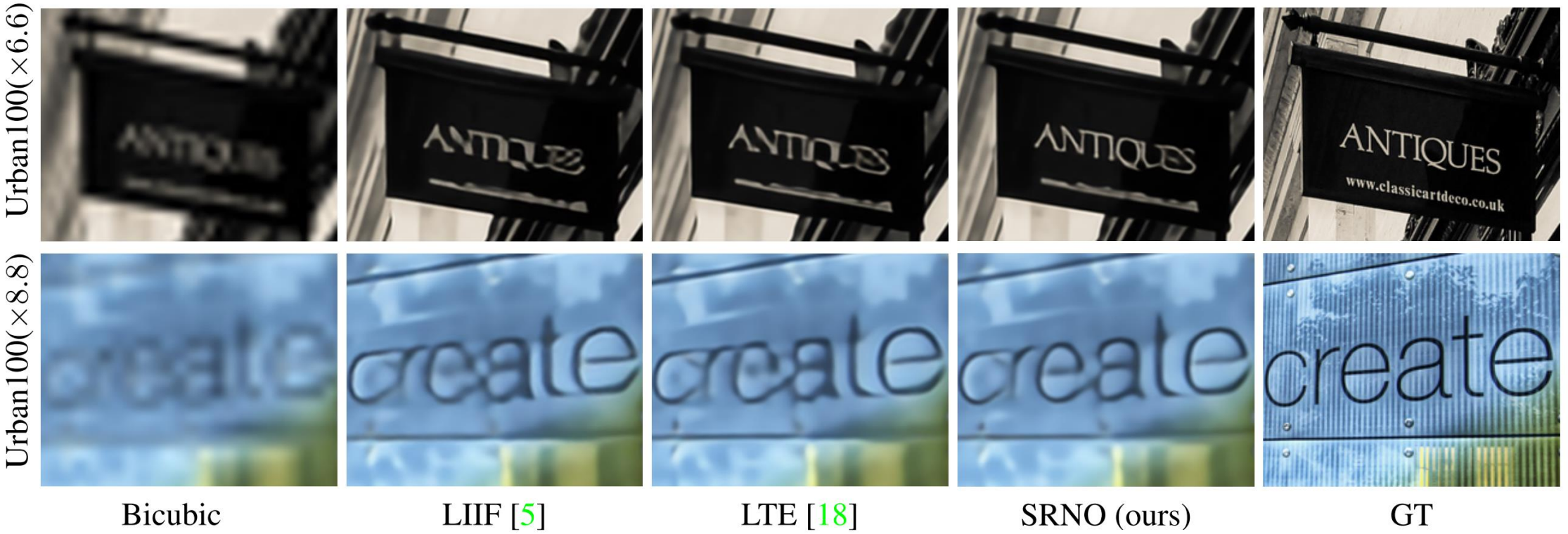


Qualitative result

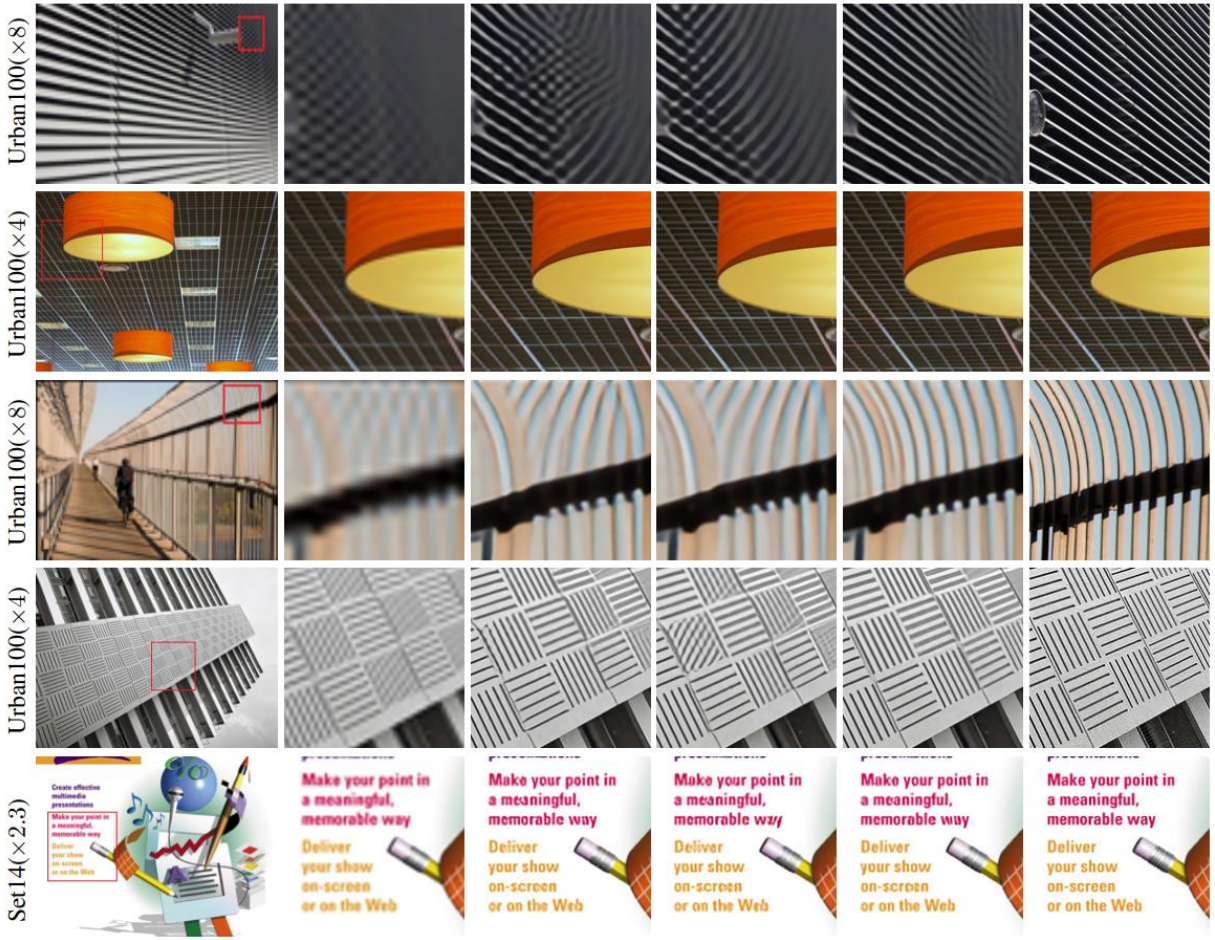


Super-Resolution Neural Operator, Min Wei, Xuesong Zhang.

Qualitative result



Qualitative result



Quantitative result

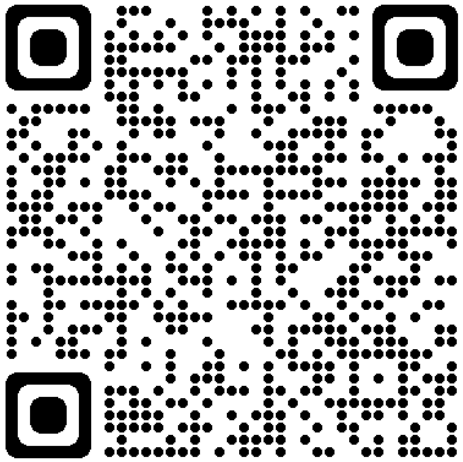
Method	In-distribution			Out-of-distribution				
	×2	×3	×4	×6	×12	×18	×24	×30
Bicubic	31.01	28.22	26.66	24.82	22.27	21.00	20.19	19.59
EDSR-baseline [21]	34.55	30.90	28.94	-	-	-	-	-
EDSR-baseline-MetaSR [5, 11]	34.64	30.93	28.92	26.61	23.55	22.03	21.06	20.37
EDSR-baseline-LIIF [5]	34.67	30.96	29.00	26.75	23.71	22.17	21.18	20.48
EDSR-baseline-LTE [18]	34.72	31.02	29.04	26.81	23.78	22.23	21.24	20.53
EDSR-baseline-SRNO (ours)	**34.85**	**31.11**	**29.16**	**26.90**	**23.84**	**22.29**	**21.27**	**20.56**
RDN-MetaSR [5, 11]	35.00	31.27	29.25	26.88	23.73	22.18	21.17	20.47
RDN-LIIF [5]	34.99	31.26	29.27	26.99	23.89	22.34	21.31	20.59
RDN-LTE [18]	35.04	31.32	29.33	27.04	23.95	22.40	21.36	20.64
RDN-SRNO (ours)	**35.16**	**31.42**	**29.42**	**27.12**	**24.03**	**22.46**	**21.41**	**20.68**

Table 1. **Quantitative comparison on DIV2K validation set (PSNR (dB))**. The best performance are bolded. EDSR-baseline trains separate models for the three in-distribution scales. The rest methods use a single model for all scales, and are trained with continuous random scales uniformly sampled in ×1–×4.

Method	In-distribution			Out-of-distribution		In-distribution			Out-of-distribution	
	×2	×3	×4	×6	×8	×2	×3	×4	×6	×8
	Set5					Set14				
RDN [41]	38.24	34.71	32.47	-	-	34.01	30.57	28.81	-	-
RDN-MetaSR [5, 11]	38.22	34.63	32.38	29.04	26.96	33.98	30.54	28.78	26.51	24.97
RDN-LIIF [5]	38.17	34.68	32.50	29.15	27.14	33.97	30.53	28.80	26.64	25.15
RDN-LTE [18]	38.23	34.72	32.61	29.32	27.26	34.09	30.58	28.88	26.71	25.16
RDN-SRNO (ours)	**38.32**	**34.84**	**32.69**	**29.38**	**27.28**	**34.27**	**30.71**	**28.97**	**26.76**	**25.26**
	B100					Urban100				
RDN [41]	32.34	29.26	27.72	-	-	32.89	28.80	26.61	-	-
RDN-MetaSR [5, 11]	32.33	29.26	27.71	25.90	24.83	32.92	28.82	26.55	23.99	22.59
RDN-LIIF [5]	32.32	29.26	27.74	25.98	24.91	32.87	28.82	26.68	24.20	22.79
RDN-LTE [18]	32.36	29.30	27.77	26.01	24.95	33.04	28.97	26.81	24.28	22.88
RDN-SRNO (ours)	**32.43**	**29.37**	**27.83**	**26.04**	**24.99**	**33.33**	**29.14**	**26.98**	**24.43**	**23.02**

Table 2. **Quantitative comparison on benchmark datasets (PSNR (dB))**. The best performances are in bold. RDN trains separate models for the three in-distribution scales. The rest methods use a single model for all scales, and are trained with continuous random scales uniformly sampled in ×1–×4.

https://github.com/2y7c3/Super-Resolution-Neural-Operator



Code is available