Solving Diffusion ODEs with Optimal Boundary Conditions for Better Image Super-Resolution

Yiyang Ma[1], Huan Yang[2], Wenhan Yang[3], Jianlong Fu[2], Jiaying Liu[1]
[1]Wangxuan Institute of Computer Technology, Peking University
[2]Microsoft Research [3]Pengcheng Laboratory

STRUCT @PKU
Spatial and Temporal Restoration, Understanding and Compression

Introduction

- **Problem:**

  Instability during inference of Diffusion-based SR models due to the randomness.

- **Existing Works:**

  Did not deal with the phenomenon.

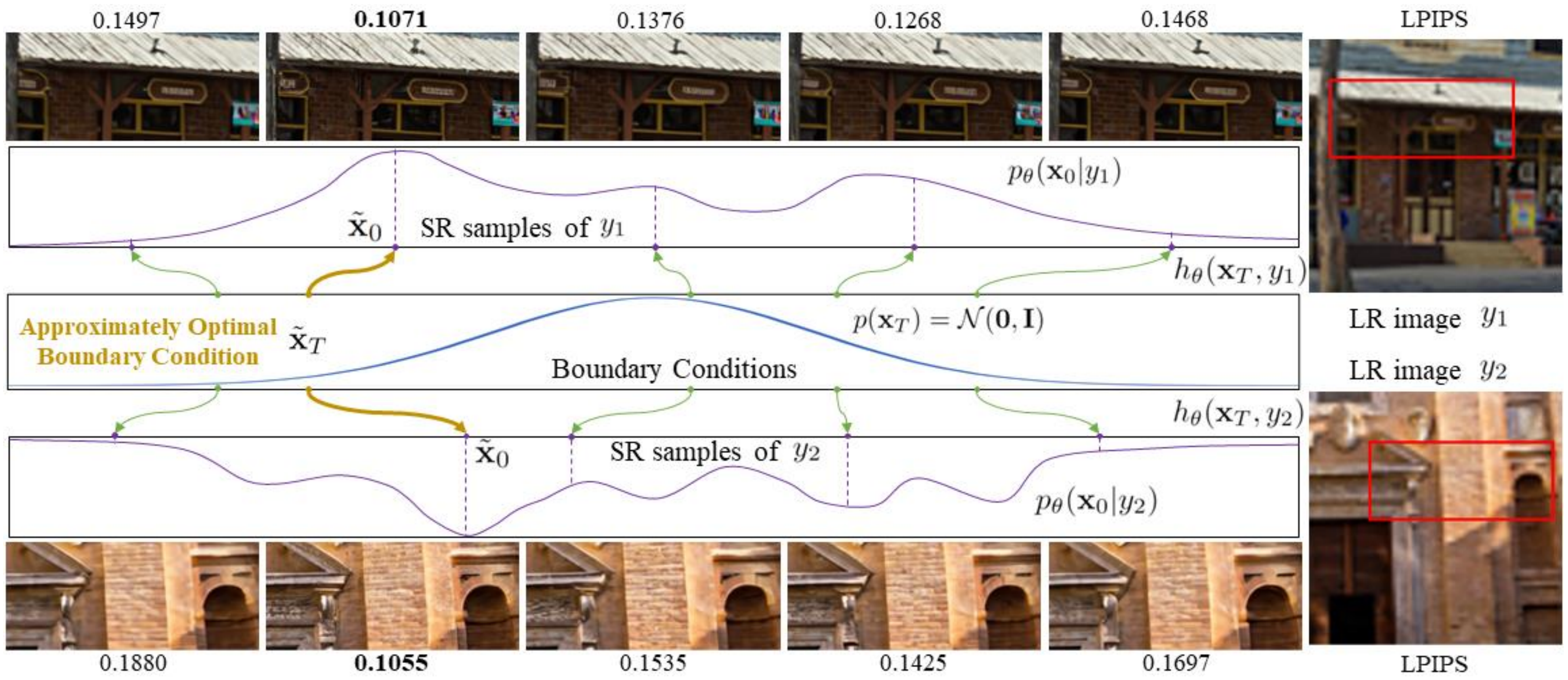
  Simply fixed the random seed during inference.

- **Our Solution**

  Analysis on the Approx. Optim. BC of Diffusion ODEs.

  Obtaining the $\tilde{\mathbf{x}}_T$ with a ref. set of image pairs.

  Applying the $\tilde{\mathbf{x}}_T$ to all the LRs, achieving better and stable results.



Approach - Can Boost All the Diffusion-based SR Models

- **Preliminary: Diffusion ODE sampler**

  The results are determined by the **boundary condition** $\mathbf{x}_T$ (*i.e.,* the Gaussian noise at the start of the sampling process).

  e.g., DDIM (*ICLR* 21'), DPM-Solver (*NIPS* 22').

  Thus, the SR images can be a function of BC and LR image:

$$\mathbf{x}_0 = h_\theta(\mathbf{x}_T, \mathbf{y})$$

- **Find an Approx. Optim. BC $\tilde{\mathbf{x}}_T$**

  We employ perceptual distance between SR and HR as an approximated implementation of the likelihood.

  We build a reference set and calculate the $\tilde{\mathbf{x}}_T$ on it:

$$\tilde{\mathbf{x}}_T \approx \mathop{\arg\min}_{\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})} \sum_{i=1}^{R} M(h_\theta(\mathbf{x}_T, \mathbf{y}_i), \mathbf{z}_i)$$

- **Optimal Boundary Condition $\mathbf{x}_T^*$**

  The BC that can generate the SR image which is the most close to the HR image:

$$\mathbf{x}_0^* = h_\theta(\mathbf{x}_T^*, \mathbf{y}), \text{ where } \mathbf{x}_0^* = \mathop{\arg\max}_{\mathbf{x}_0} p_\theta(\mathbf{x}_0|\mathbf{y})$$

  We prove that the $\mathbf{x}_T^*$ is approximately consistent to different LRs:

$$\mathbf{x}_T^* = \mathop{\arg\max}_{\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})} p_\theta(h_\theta(\mathbf{x}_T, \mathbf{y})) \approx \mathop{\arg\max}_{\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})} p_\theta(h_\theta(\mathbf{x}_T, \mathbf{y}_i)), \forall \mathbf{y}_i \in \mathcal{C}$$

- **Apply the $\tilde{\mathbf{x}}_T$ to Different LRs**

  As we have declared before, the $\tilde{\mathbf{x}}_T$ can be directly applied in the sampling process of different LRs:

$$\tilde{\mathbf{x}}_0 = h_\theta(\tilde{\mathbf{x}}_T, \mathbf{y})$$

  achieving stable and better SR performances.

  The method does not limited the SR model itself, which means it **can be employed in all the diffusion-based SR models.**

Experiments

- **We evaluate our approach on 3 different diffusion-based SR models.**

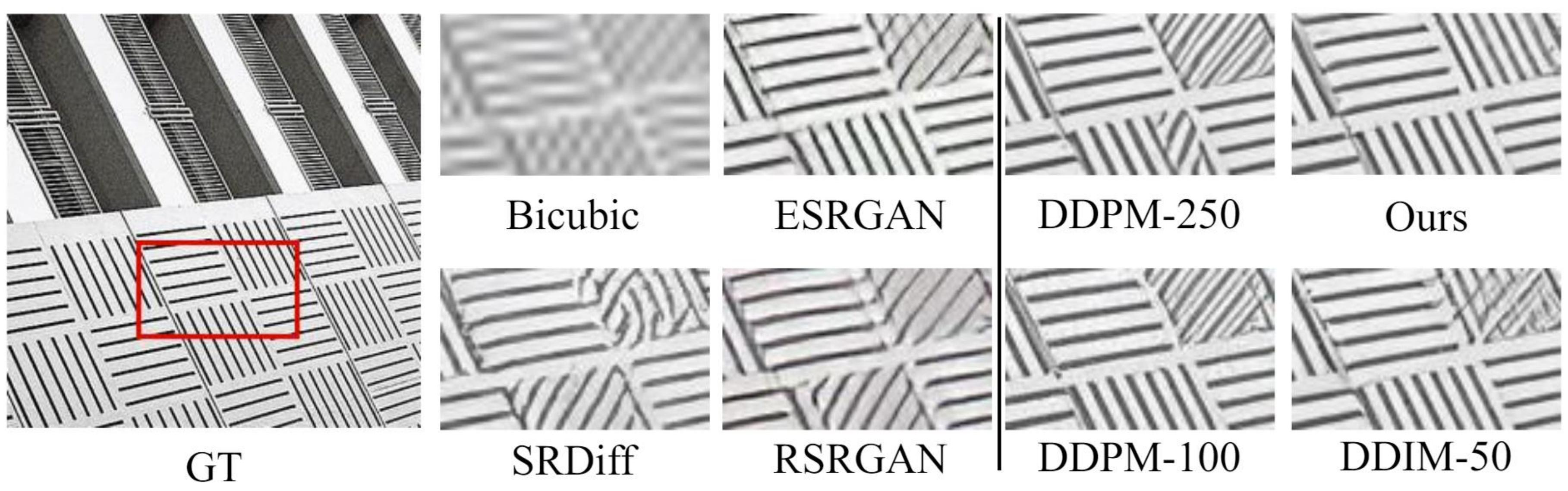
- **On Real World-SR**

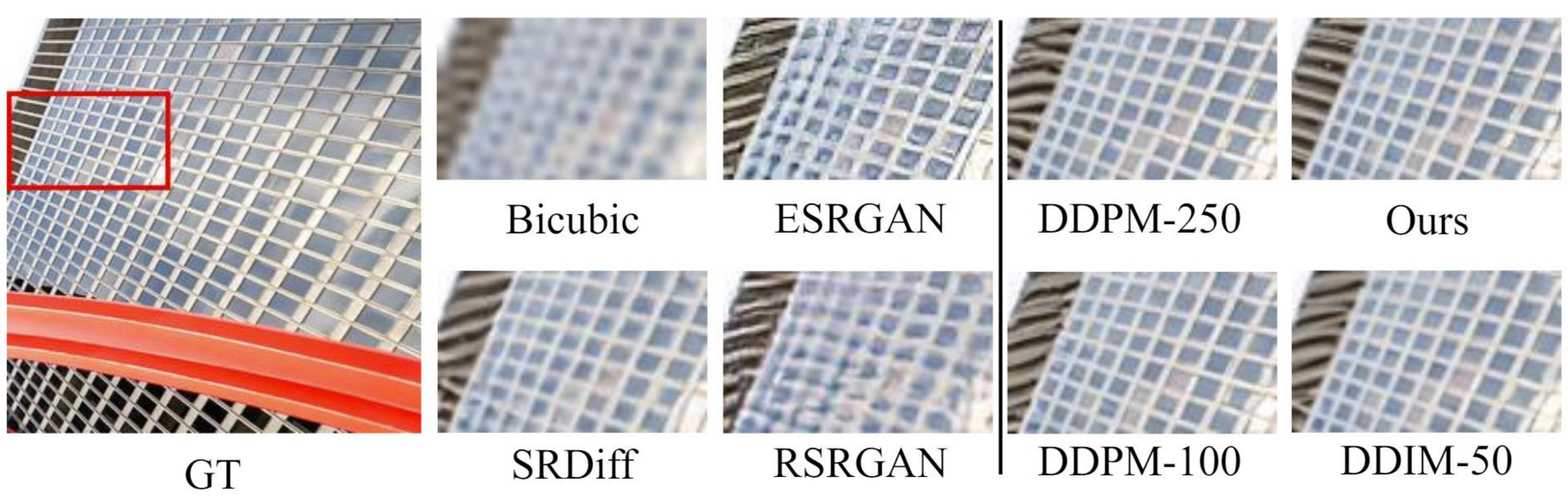
Model (& sampling method)		DIV2k-test DISTS ↓	DIV2k-test LPIPS ↓	DIV2k-test PSNR ↑	RealSR DISTS ↓	RealSR LPIPS ↓	RealSR PSNR ↑
RealSR		0.3051	0.5148	22.52	0.2532	0.3673	26.30
BSRGAN		0.2253	0.3416	22.13	0.2057	0.2582	25.52
DASR		0.2340	0.3444	22.02	0.2113	0.3014	26.32
Real-ESRGAN		0.2108	0.3109	22.36	0.2020	0.2511	25.12
KDSR-GAN		0.2022	0.2840	22.92	0.2006	0.2425	26.09
StableSR	DDPM-200	0.2010	0.3189	19.42	0.2210	0.3065	21.37
	DDIM-50	0.2217	0.3629	18.82	0.2336	0.3536	21.24
	DDIM-50 + $\tilde{\mathbf{x}}_T$	0.2046	0.3169	19.55	0.2164	0.2999	22.13
DiffIR	D-4	0.1773	0.2360	22.94	0.2076	0.2604	25.33
	D-4 + $\tilde{\mathbf{x}}_T$	0.1772	0.2357	22.95	0.1993	0.2419	25.82

- **On Bicubic-SR**

Model (& sampling method)		DIV2k-test LPIPS ↓	DIV2k-test PSNR ↑	Urban100 LPIPS ↓	Urban100 PSNR ↑	BSD100 LPIPS ↓	BSD100 PSNR ↑
ESRGAN		0.1082	28.18	0.1226	23.04	0.1579	23.65
RankSRGAN		0.1171	27.98	0.1403	23.16	0.1714	23.80
SRDiff		0.1286	28.96	0.1391	23.88	0.2046	24.17
SR3	DDPM-1000	0.1075	28.75	0.1165	24.33	0.1555	23.86
	DDPM-250	0.1142	28.95	0.1181	24.41	0.1621	24.00
	DDPM-100	0.1257	29.16	0.1232	24.51	0.1703	24.15
	DPMS-20	0.1653	27.25	0.1413	23.46	0.2037	22.79
	DDIM-50	0.1483	28.55	0.1333	24.16	0.1823	23.75
	DDIM-100	0.1571	28.16	0.1335	24.05	0.1950	23.55
	DPMS-20 + $\tilde{\mathbf{x}}_T$	0.1210	27.45	0.1179	23.57	0.1687	22.81
	DDIM-50 + $\tilde{\mathbf{x}}_T$	0.1053	28.65	0.1164	24.26	0.1552	23.99
	DDIM-100 + $\tilde{\mathbf{x}}_T$	0.1032	28.48	0.1136	24.12	0.1505	23.67

- **Visual Results**





Project Website: https://realpasu.github.io/SolvingDiffODE4SR_Website/

Team Website: http://struct.wict.pku.edu.cn/