# Unsupervised Real-World Super-Resolution: A Domain Adaptation Perspective

Wei Wang, Haochen Zhang, Zehuan Yuan, Changhu Wang

2021 ICCV OCTOBER 11-17 VIRTUAL

## INTRODUCTION

### ➢Motivation

- Most existing convolution neural network based super-resolution (SR) methods generate their paired training dataset by simplly interpolating high-resolution (HR) images to their low-resolution (LR) version or artificially generating “real” LR counterparts from real HR images using image-to-image translation.
- However, it is still difficult to train an ideal degraded LR image generator to perfectly mimic real images and real-world SR remains a challenging problem so far. In this paper we reconsider the unpaired real-world SR from a feature-level domain adaptation perspective.

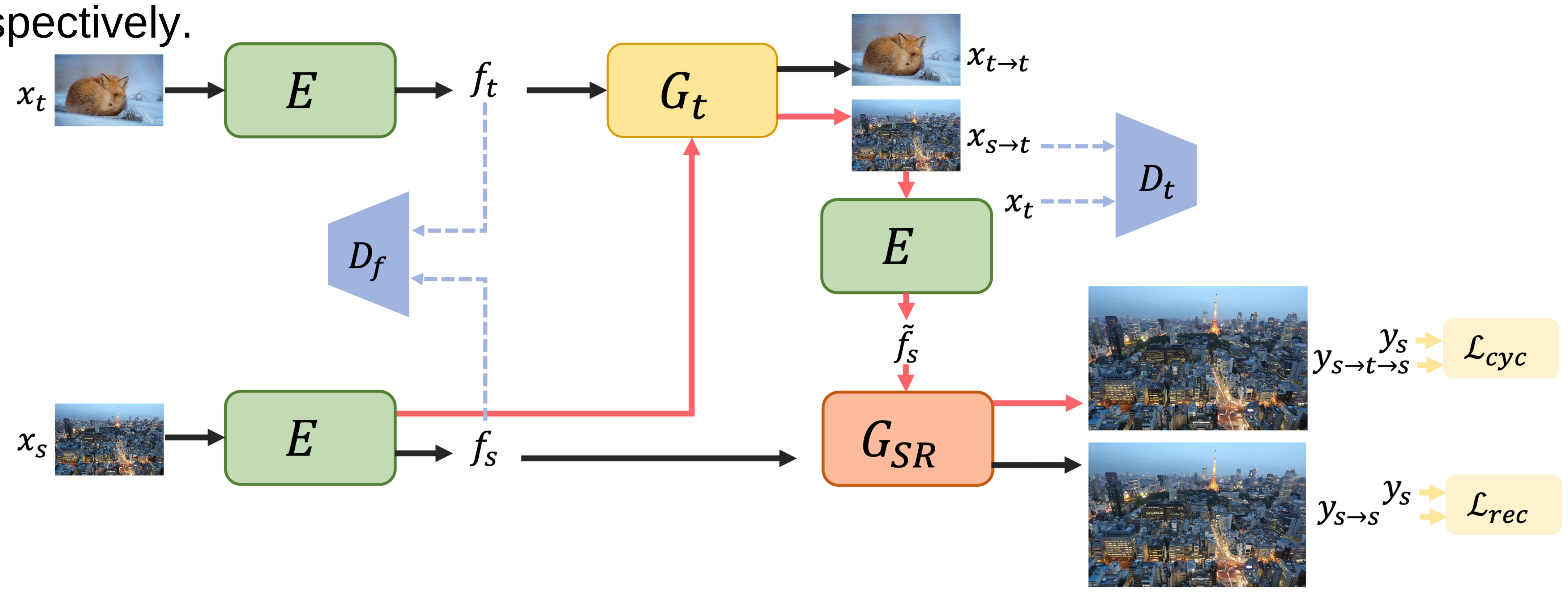
### ➢Contribution

- We propose a novel unpaired SR training framework based on feature distribution alignment.
- We introduce several losses to not only align feature space better but also preserve image details for the SR task.
- Extensive experiments on three challenging datasets show that our proposed method has advantages over the existing unpaired SR training solutions.

## PROPOSED METHOD

### ➢Framework

$E$ means three copies of a same encoder and $G_t$, $G_{SR}$ represent two different decoders. $x_t$ and $x_s$ are input LR images from target and source domain respectively, and $f_t$, $f_s$ are corresponding feature maps. As the arrows in different colors imply, $x_{t\to t}$ is an image restored from feature $f_t$ which has the same contents and degradation with $x_t$. $x_{s\to t}$ is generated from feature $f_s$ with contents of $x_s$ but degradation of $x_t$. Then $x_{s\to t}$ is fed into encoder $E$ to extract feature $\tilde{f}_s$. Finally, the super-resolved images $y_{s\to t\to s}$ and $y_{s\to s}$ is generated from feature maps $\tilde{f}_s$ and $f_s$ respectively.




The framework can be divided into two main parts: **(a) feature distribution alignment** **(b) feature domain regularization**

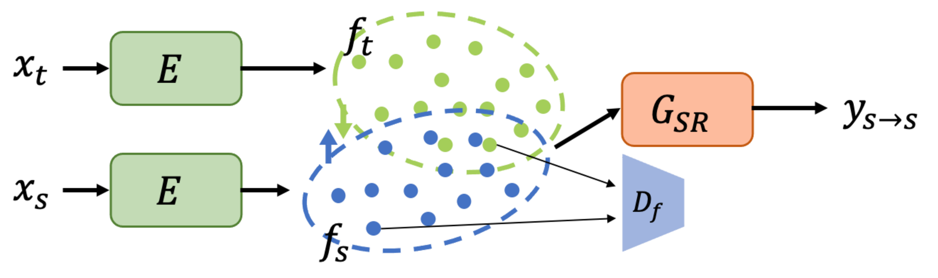
- SR Reconstruction loss

$$\mathcal{L}_{rec}(E,G_{SR}) = \|y_s - y_{s\to s}\|_1 + \alpha\mathcal{L}_{fea}(y_s, y_{s\to s}) + \beta\mathcal{L}_{adv}(y_s, y_{s\to s})$$

- Feature alignment loss

$$\min_{\theta_E}\mathcal{L}_{align}(E) = \mathbb{E}_{x_t\sim\mathcal{T}(x)}\left[(D_f(E(x_t)) - 0.5)^2\right] + \mathbb{E}_{x_s\sim\mathcal{S}(x)}\left[(D_f(E(x_s)) - 0.5)^2\right]$$

$$\min_{\theta_{Df}}\mathcal{L}_{align}(D_f) = \mathbb{E}_{x_t\sim\mathcal{T}(x)}\left[(D_f(E(x_t)) - 0)^2\right] + \mathbb{E}_{x_s\sim\mathcal{S}(x)}\left[(D_f(E(x_s)) - 1)^2\right]$$




(a) Feature Distribution Alignment

- Target LR restoration loss

$$\mathcal{L}_{res}(E,G_t) = \|x_t - x_{t\to t}\|_1$$

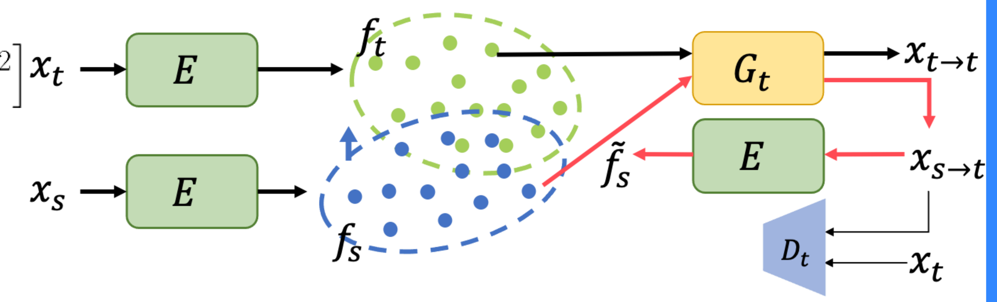
- Target degradation style loss

$$\min_{\theta_E,\theta_{G_t}}\mathcal{L}_{sty}(E,G_t) = \mathbb{E}_{f_s\sim\mathcal{S}(f)}\left[(D_t(G_t(f_s)) - 1)^2\right]$$

$$\min_{\theta_{D_t}}\mathcal{L}_{sty}(D_t) = \mathbb{E}_{f_s\sim\mathcal{S}(f)}\left[(D_t(G_t(f_s)) - 0)^2\right] + \mathbb{E}_{x_t\sim\mathcal{T}(x)}\left[(D_t(x_t) - 1)^2\right]$$

- Feature identity loss

$$\mathcal{L}_{idt}(E,G_t) = \left\|f_s - \tilde{f}_s\right\|_1$$




(b) Feature Domain Regularization

- Cycle loss

$$\mathcal{L}_{cyc}(E,G_t,G_{SR}) = \|y_s - y_{s\to t\to s}\|_1 + \alpha\mathcal{L}_{fea}(y_s, y_{s\to t\to s}) + \beta\mathcal{L}_{adv}(y_s, y_{s\to t\to s})$$

- Full objective

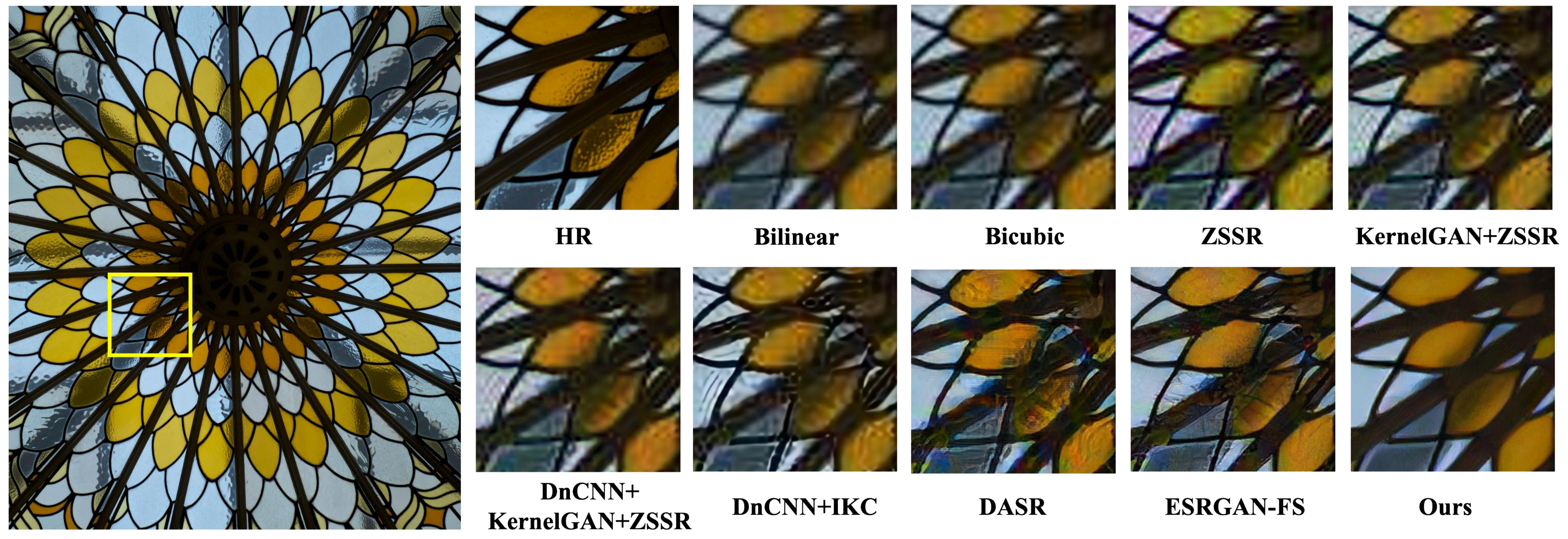
$$\mathcal{L}_{train} = \lambda_{align}\mathcal{L}_{align}(E) + \lambda_{rec}\mathcal{L}_{rec}(E,G_{SR}) + \lambda_{res}\mathcal{L}_{res}(E,G_t) + \lambda_{sty}\mathcal{L}_{sty}(E,G_t) + \lambda_{idt}\mathcal{L}_{idt}(E,G_t) + \lambda_{cyc}\mathcal{L}_{cyc}(E,G_t,G_{SR})$$

## EXPERIMENTAL RESULTS

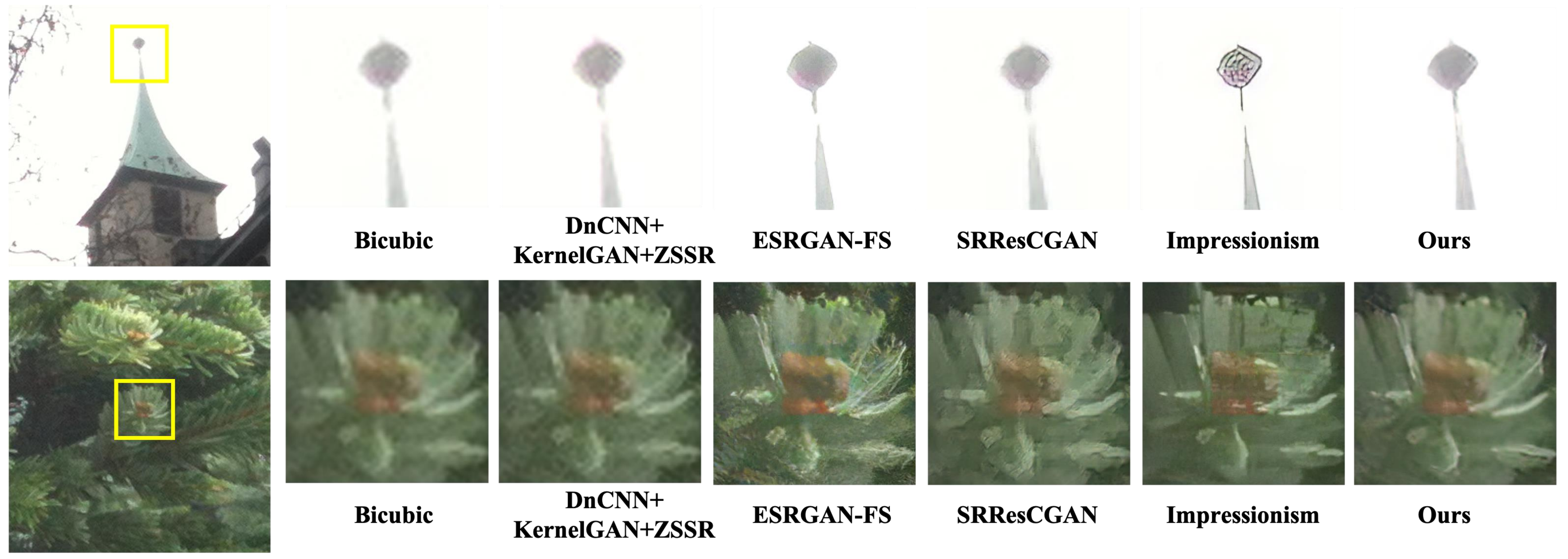
### ➢Quantitative Comparisons

| AIM 2019 Method | LPIPS↓ | PSNR↑ | SSIM↑ | NTIRE 2020 Method | LPIPS↓ | PSNR↑ | SSIM↑ |
|---|---|---|---|---|---|---|---|
| †Bicubic | 0.673 | 22.36 | 0.614 | †Bicubic | 0.632 | 25.52 | 0.671 |
| †MadDeamon(Winner) | 0.403 | 21.00 | 0.504 | †Impressionism(Winner) | 0.227 | 24.83 | 0.672 |
| ZSSR[27] | 0.639 | 22.21 | 0.603 | ZSSR[27] | 0.620 | 24.93 | 0.642 |
| KernelGAN[1]+ZSSR[27] | 0.613 | 22.40 | 0.611 | KernelGAN[1]+ZSSR[27] | 0.598 | 25.34 | 0.661 |
| DnCNN[41]+K.[1]+Z.[27] | 0.607 | 22.40 | 0.614 | DnCNN[41]+K.[1]+Z.[27] | 0.438 | 25.84 | 0.722 |
| DnCNN[41]+IKC[10] | 0.614 | 22.26 | 0.596 | DnCNN[41]+IKC[10] | 0.384 | 26.50 | 0.748 |
| *Maeda *et al.* [21] | 0.454 | 22.88 | 0.661 | SRResCGAN[35] | 0.335 | 25.05 | 0.676 |
| DASR[38] | 0.346 | 21.79 | 0.577 | | | | |
| Ours | 0.340 | 22.60 | 0.622 | Ours | 0.252 | 25.40 | 0.707 |

**➢Visual Performance:** Our method could generate high-frequency details with the least artifacts on synthetic and real data.




Visual results on synthetic image




Visual results on real phone image