Rethinking Image Super Resolution from Long-Tailed Distribution Learning Perspective

Yuanbiao Gou¹, Peng Hu¹, Jiancheng Lv¹, Hongyuan Zhu², Xi Peng¹∗
¹ College of Computer Science, Sichuan University, China
² Institute for Infocomm Research (I²R), A*STAR, Singapore
{gouyuanbiao, penghu.ml, hongyuanzhu.cn, pengx.gm}@gmail.com; lvjiancheng@scu.edu.cn

Abstract

Existing studies have empirically observed that the resolution of the low-frequency region is easier to enhance than that of the high-frequency one. Although plentiful works have been devoted to alleviating this problem, little understanding is given to explain it. In this paper, we try to give a feasible answer from a machine learning perspective, i.e., the twin fitting problem caused by the long-tailed pixel distribution in natural images. With this explanation, we reformulate image super resolution (SR) as a long-tailed distribution learning problem and solve it by bridging the gaps of the problem between low- and high-level vision tasks. As a result, we design a long-tailed distribution learning solution, that rebalances the gradients from the pixels in the low- and high-frequency region, by introducing a static and a learnable structure prior. The learned SR model achieves better balance on the fitting of the low- and high-frequency region so that the overall performance is improved. In the experiments, we evaluate the solution on four CNN- and one Transformer-based SR models w.r.t. six datasets and three tasks, and experimental results demonstrate its superiority.

1. Introduction

Image super resolution aims to restore a high-resolution (HR) image from a low-resolution (LR) one, which is an important technique in image processing [13, 26, 27, 52] and computer vision [7, 14, 18, 45, 51]. In the past decades, plentiful SR methods have been proposed [19, 53], and applied to a wide range of real-world applications [21, 47, 49, 54].

Among existing studies, the learning-based methods that learn a mapping between LR and HR image spaces have achieved the state-of-the-art performance [17, 39, 43, 58, 59]. Nonetheless, they have empirically observed that the high-frequency regions are harder to be super-resolved than the low-frequency ones in the natural image. To alleviate that, various SR methods have been proposed following the below two paradigms, i.e., developing generalized models with larger capacities [31, 36] or specific models with high-frequency enhancements [37, 48]. The former obtains better results in both the high- and low-frequency regions via constantly enlarging the capacities, while the latter enhances the high-frequency regions through specific auxiliary sub-networks, loss functions, training strategies, etc. Although the promising results have been obtained, they involve the following three limitations. First, the large capacity models take a lot of time and computations in the training and infer-
ring, which is unavailable to mobile scenarios. Second, the specific models need ingenious designs about the architecture and training strategy, which is difficult to training and prone to artifacts. Third, they don’t dive into the problem and give a reasonable explanation, thus alleviate the problem not in the most cost-effective way.

In this paper, we dive into the problem and explain it from a machine learning perspective, i.e., the twin fitting problem caused by the long-tailed pixel distribution in the natural images. Taking the Fig. 1 as an example, the number of pixels in the low-frequency region is far more than that in the high-frequency one, i.e., the long-tailed pixel distribution. Since majority pixels in the low-frequency region dominate minority pixels in the high-frequency one, the gradients of SR model are mainly from the former instead of the latter. As a result, the SR model is optimized to mainly fit the pixels in the low-frequency region, and thus over-fitting them while underfitting those in the high-frequency region, i.e., the twin fitting problem.

Motivated by the above explanation, we reformulate SR as the long-tailed distribution learning problem. With this reformulation, the twin fitting problem could be alleviated during training in a model-agnostic way, and thus applicable to different SR models. However, although the long-tailed distribution learning problem has been extensively studied in high-level vision tasks, there are few works on it in low-level ones. Therefore, we bridge the gaps of the problem between in low- and high-level vision ones, and design a simple and effective solution to verify the feasibility of our reformulation. To be specific, we design a novel long-tailed distribution learning method for SR, termed as Focal Pixel Learning (FPL), which adaptively re-weights the loss contribution of pixels by combining two complementary structure priors. In this way, the gradients of SR model could be rebalanced, leading it to achieve better balance on the fitting of the high- and low-frequency regions.

The contributions of this work are summarized below.

- For the first time, this work dives into the observation that the high-frequency regions are harder to be super-resolved than the low-frequency ones, and gives a reasonable explanation, i.e., the long-tailed pixel distribution and it caused twin fitting problem.
- With our explanation, this work reformulates SR as a long-tailed distribution learning problem and designs a novel solution to verify its feasibility, which could be the first long-tailed distribution learning solution for SR, as far as we know.
- Extensive analyses and experiments are conducted to demonstrate the explanation, verify the reformulation, and validate the solution. The results demonstrate that our works could consistently improve the performance of SR models with different complexities.

2. Related Works

Here, we briefly review the related works of image super resolution and long-tailed distribution learning.

2.1. Image Super Resolution

Although a lot of SR models have been proposed [11, 16, 33, 42], they are advanced along two directions. One is to develop generalized models with larger capacities. For example, SRCNN [9] introduced the convolution neural network into SR for the first time, and outperformed the traditional methods. RDN [58] introduced the dense connections to utilize the hierarchical features from all convolutional layers. SwinIR [31] introduced the shifted windowing scheme to model the local attention and long-range dependency. ESRT [36] utilized a convolution neural network to extract deep features first and then used a Transformer to model the long-range dependency. The other direction is to develop specific models with high-frequency enhancements. For instance, PCL [48] proposed a contrastive learning framework to enhance LR images into sharp HR images. SPSR [37] introduced a gradient map SR network to guide the high-frequency region recovery in the image SR network. WDST [8] decomposed an image into high- and low-frequency sub-bands, and separately enhanced them via different subnetworks. SA [44] introduced a heuristic metric to exploit informative LR-HR patch pairs for training.

Different from them, this work dives into the observation that the high-frequency regions are harder to be super-resolved than the low-frequency regions, and explains it from the twin fitting problem caused by the long-tailed pixel distribution. Inspired by this explanation, this work reformulates SR as a long-tailed distribution learning problem, and designs a novel model-agnostic solution. This solution could endow the SR model with the better recovery capability of the high-frequency regions, without introducing extra model capacities or auxiliary strategies.

2.2. Long-tailed Distribution Learning

In real-world applications, samples typically exhibit a long-tailed distribution, where a small portion of classes have massive samples but the others are with only a few samples. With the unbalanced training data, models would be biased towards majority classes with massive samples, leading to poor performance on minority classes that have limited samples [4, 32, 46]. To address the problem, massive studies have been conducted in recent years [6, 22, 34, 55, 56], which could be categorized into class re-balancing, information augmentation and module improvement. Here, we briefly review the related category of class re-balancing, and more details could be referred to [57].

Class re-balancing is the main paradigm in the long-tailed distribution learning, which balances training samples of different classes through re-sampling, cost-sensitive
learning, and/or logit adjustment. As logit adjustment post-
hoc shifts the logits based on label frequencies, we focus
on re-sampling and cost-sensitive learning which act on
the training process. To be specific, re-sampling usually under-
samples the majority classes and/or over-samples the
minority classes, e.g., SMOTE [5] linearly interpolates sam-

dles for the minority classes, and UNSAM [41] learns a data

sampler to discard samples. Besides, cost-sensitive learning
re-balances classes by re-weighting the loss contribution
of different classes during the training. For example, CB [6]
re-weights the loss value to be inverse to the effective num-
ber of samples per class. FL [32] down-weights the loss val-
ues assigned to the majority classes and the well-classified
examples. GHM [28] re-weights the loss values of samples
based on their gradients per iteration.

Although the long-tailed distribution learning has been
extensively studied in high-level vision tasks, such as clas-
sification and detection, few studies are conducted in low-
level ones. Different from the existing works, this work first
reveals the long-tailed pixel distribution in natural images
and the twin fitting problem in SR, and then achieves SR
in a long-tailed distribution learning paradigm. To the best
of our knowledge, this work could be the first study on this
topic for SR.

3. The Proposed Solution

In this section, we first theoretically explain why the SR
models suffer from the twin fitting problem, and then elabo-
rate on how does the proposed solution learn from the long-
tailed pixel distribution and alleviates the problem.

3.1. Twin Fitting Problem in SR

SR models are generally trained through pixel-wise loss
function $\ell_p$ on a large number of natural images. Specif-
cally, for a given HR image $y$ as well as the LR counterpart
$x$, the pixel-wise loss function $\ell_p$ is

$$L = \frac{1}{I} \sum_{i=1}^{I} |f(x)_i - y_i|^p,$$  

where $f(\cdot)$ denotes the SR model, $i$ and $I$ indicate the pixel
index and number, respectively. With the formula, we could observe that $\ell_p$ treats all pixels equally, i.e., every pixel, ei-
ther in the high- or low-frequency region, equally contribute to
the overall loss value. However, as depicted in Fig. 1, the pixels in the natural images show a long-tailed distribu-
tion, i.e., the number of pixels in the low-frequency region is far more than that in the high-frequency region. For clar-
ity, let $M$ and $N$ denote the pixel number in the low- and
high-frequency region, respectively. Therefore, $\ell_p$ in Eq. (1)
could be reformulated as the sum of the loss values in the
two regions, i.e.,

$$L = \frac{1}{I} \sum_{m=1}^{M} |f(x)_m - y_m|^p + \frac{1}{I} \sum_{n=1}^{N} |f(x)_n - y_n|^p,$$

s.t. $M \gg N$ and $M + N = I$.

Obviously, the majority pixels in the low-frequency re-

gion could easily dominate the minority pixels in the high-
frequency region on the contribution of overall loss value.
Therefore, the gradients of SR model are mainly from the
low-frequency regions and be biased toward fitting them.
As a result, SR model would overfit the low-frequency re-
regions while underfitting the high-frequency ones. Here, we
term this competing fitting issue as the twin fitting problem, which is particularly severe in the limited capacity models.

3.2. Long-tailed Distribution Learning for SR

As the twin fitting problem arises from the long-tailed
pixel distribution, it is highly expected to solve it from the
long-tailed distribution learning perspective, i.e., recasting
SR as a long-tailed distribution learning problem. Although
plentiful methods have been proposed to solve it in high-
level vision tasks, there are few studies in low-level vision
tasks due to the following two obstacles. First, no semantic
class label to indicate a pixel belonging to the high- or low-
frequency region since it depends on the surroundings. Sec-
ond, no specific boundary between high- and low-frequency
regions since the frequencies are distributed continuously.
In other words, it is daunting to conduct long-tailed distribu-
tion learning for low-level vision tasks due to the absence of
discrete and semantic labels for pixels. Therefore, to solve
the twin fitting problem through long-tailed pixel distribu-
tion learning, we should overcome the two obstacles.

3.2.1 Two Structure Priors

To overcome the first obstacle, we introduce two structure
priors to jointly indicate the pixel distribution about the low-
and high-frequency region in a heuristic way.

The structure prior $y_{sp}$ comes from the observation il-

dustrated in Fig. 1, i.e., BI usually achieves better results in the
low-than high-frequency regions. Namely, pixels with large absolute differences between HR image $y$ and BI image $f_{BI}(x)$ have a high probability of being in the high-
frequency region. Based on the observation, we formulate a static structure prior as below,

$$y_{sp} = |y - f_{BI}(x)|,$$

in which the pixels with small values are more likely to be in the low-frequency region, while the pixels with large values are more likely to be in the high-frequency region. Therefore, $y_{sp}$ could be viewed as the sketoch labels for the pixels in different frequency regions.
Heuristic labels to indicate the pixels (process for solving the twin fitting problem. As a result, we transform the structure priors into soft weights for re-weighting the loss contributions per pixel. The second obstacle is no specific boundaries among the pixels with large absolute differences between HR image \( y \) and SR image \( f(x) \) are more likely to be those not be and hard to be fitted in the high-frequency region. With this observation, we formulate a learnable structure prior as below,

\[
y_{lp} = |y - f(x)|, \tag{4}
\]

in which the small values indicate the pixels that are easy to be fitted in the low-frequency regions, while the large values indicate those hard to be fitted in the high-frequency ones. Hence, \( y_{lp} \) could be viewed as the subtle labels for pixels labeled through the criterion of fitting difficulty. Meanwhile, \( y_{lp} \) could endow the model with the capacity of knowing itself fitting degree, thus dynamically adjusting the training process for solving the twin fitting problem.

In summary, we introduce the two structure priors as the heuristic labels to indicate the pixels (i.e., \( y_{sp} \)) as well as those easy and hard to be fitted (i.e., \( y_{lp} \)) in the low- and high-frequency regions. As a result, the first obstacle could be overcome.

### 3.2.2 Focal Pixel Learning

The second obstacle is no specific boundaries among the different frequencies, and thus the pixels cannot be divided into either high- or low-frequency ones. To overcome that, we follow the cost-sensitive learning paradigm, which re-weights the loss contributions of different classes during the training. As a result, we transform the structure priors into soft weights for re-weighting the loss contribution per pixel. Given the structure priors \( y_{sp} \) and \( y_{lp} \), we first harmonize them so that they are in a comparable magnitude, i.e.,

\[
g(z) := \frac{z - \min(z)}{\max(z) - \min(z)} \quad \text{where } z \in \{y_{sp}, y_{lp}\}. \tag{5}
\]

With the weighting function, the structure priors could be transformed into the weights, while keeping the relative size of values, i.e., the large weights correspond to the pixels in the high-frequency region, while the small weights correspond to those in the low-frequency one. With the weighting matrices, long-tailed distribution learning for SR could be achieved in a cost-sensitive learning way.

As illustrated in Fig. 1, SR models usually perform remarkably better in the high-frequency regions while slightly better in the low-frequency ones than BI. Therefore, we introduce BI into the solution and endow it with a novel connotation. Specifically, BI is essentially a re-sampling strategy for long-tailed pixel distribution learning, which under-samples the pixels in the low-frequency region, i.e.,

\[
\hat{y} = y - f_{BI}(x). \tag{6}
\]

In other words, the number of pixels in the low-frequency region that need to be fitted is decreased and thus allowing SR models to fit the residual pixels that on a less extremely long-tailed distribution.

Combining the two strategies, we formulate a long-tailed distribution learning solution for SR, termed as Focal Pixel Learning (FPL), i.e.,

\[
\mathcal{L}_i = W(y_{sp}; \alpha_{sp}, \gamma_{sp}) \times W(y_{lp}; \alpha_{lp}, \gamma_{lp}) \times |f(x) - \hat{y}|^p, \tag{7}
\]

and the overall loss value is \( \mathcal{L} = \frac{1}{I} \sum_{i=1}^{I} \mathcal{L}_i \). Intuitively, FPL introduces BI to under-sample the majority pixels, while introducing two structure priors and a weighting function to re-balance the pixel-wise contribution in Eq. (2). As a result, SR models focus more on learning from the pixels and those hard to be fitted in the high-frequency regions. Note that as \( y_{lp} \) involves the current fitting degree, the HR image \( y \) in Eq. (4) should be substituted with \( \hat{y} \). Besides, the gradients are blocked in the weighting function.

### 4. Experiments

In this section, we devote to verifying the feasibility of our rethinkings, i.e., achieving SR in a long-tailed distribution learning paradigm, so that the twin fitting problem could be alleviated. In the following, we will detail the experimental settings first, and then assess FPL on four CNN- and one Transformer-based methods about three SR tasks and six SR datasets. Finally, we will conduct some analysis experiments to demonstrate the effectiveness of FPL.

---

**Figure 2. Overview of the proposed FPL derived from Eq. (7).** For a given SR model \( f(\cdot) \), FPL endows it with the capacity of learning from the long-tailed pixel distribution through adaptive re-sampling and re-weighting.
Due to the space limitation, we present more experiments in supplementary material.

4.1. Experimental Settings

We use DIV2K [1] as the training dataset which contains 800 images of 2K resolution. Following ClassSR [24], we densely crop 159M LR and HR image pairs with the sizes of i) $32 \times 32$ and $128 \times 128$ for $4 \times$ SR, and ii) $32 \times 32$ and $64 \times 64$ for $2 \times$ SR, respectively. For evaluations, six widely used datasets are employed, i.e., Set5 [3], Set14 [50], BSD100 [38], Manga109 [12], Urban100 [20], and Test2K [24]. To measure the performance, two metrics of PSNR and SSIM in RGB color space are used.

As FPL is compatible with existing SR models, we introduce the models of FSRCNN [10] (tiny, 25K/468M$^1$), CARN [2] (small, 295K/1.16G), SRResNet [25] (middle, 1.5M/4.56G) and MSRN [29] (large, 6.1M/13.4G), which are representative CNN models at different complexities, and SwinIR [31] (lightweight, 1.2M/3.4G) which is a Transformer model. All models are implemented in PyTorch [40], and the experiments are conducted on Ubuntu 18.04 with GeForce RTX TITAN GPUs.

We train CARN, SRResNet, and MSRN with the iterations of 1000K, FSRCNN and SwinIR with that of 500K, based on the batch size 16 and the patch size 32 for LR images. Meanwhile, the images are augmented by flipping and rotation during the training. The initial learning rate is set to 1e-3 for FSRCNN and CARN, and 2e-4 for SRResNet, MSRN, and SwinIR. To optimize the models, we adopt the Adam [23] optimizer with the default settings, as well as the cosine annealing learning strategy [35] with the minimum learning rate of 1e-7. For a better reproducibility, we do not exhaustively tune the models on FPL. Instead, we fix $\alpha_{sp} = 0.5$, $\gamma_{sp} = 1$, $\alpha_{lp} = 1$, and $\gamma_{lp} = 1$ based on FSRCNN throughout the experiments, regardless the differences in models, datasets, SR tasks, etc.

4.2. Comparison Experiments

We first extend the models with our FPL to obtain the performance improvements. From the table, one could see that FPL boosts FSRCNN with the gains from 0.16dB/0.0012 to 0.58dB/0.0089, and CARN with that from 0.14dB/0.0008 to 0.49dB/0.0063 in PSNR/SSIM on test datasets. Meanwhile, SRResNet+FPL outperforms MSRN (6.1M/13.4G, the large size model) by 0.07dB/0.0033 in PSNR/SSIM. Analogously, MSRN+FPL outperforms SwinIR by 0.05dB/0.0043 on the Urban100. There are two reasons that affect the performance gains. First, larger capacity models fit the pixels better both in the low- and high-frequency regions, and thus less suffer from the long-tailed distribution. Second, higher resolution images have a higher proportion of low-frequency regions, and thus less benefit from alleviating the twin fitting problem.

Comparisons on $2 \times$ SR task. The quantitative results are shown in Tab. 2, which demonstrate that the FPL improves the performance of the SR models in most cases. From the table, one could see that FPL boosts FSRCNN with the gains from 0.16dB/0.0012 to 0.58dB/0.0089, and CARN with that from 0.14dB/0.0008 to 0.49dB/0.0063 in PSNR/SSIM on test datasets. Meanwhile, SRResNet+FPL obtains 0.14dB~0.57dB gains in PSNR, while MSRN+FPL and SwinIR+FPL achieve PSNR gains at most 0.11dB and 0.09dB, respectively.

Comparisons on $4 \times$ SR with multiple degradations. We introduce multiple degradations including downsampling, blurring, and noise into $4 \times$ SR task, i.e.,

$$x = (y \downarrow_s) \otimes k + n,$$  

where $\downarrow_s$ is the $s$ scale bicubic downsample, $k$ is the blur kernel, and $n$ is the additive white Gaussian noise. For better reproducibility, we fix the blur kernel $k$ with the kernel size of $3 \times 3$ and the kernel standard deviation of 5, and the noise $n$ with the noise level of 5. The quantitative results are shown in Tab. 3, from which one could see that FPL still improves the performance of the SR models, e.g., FSRCNN, CARN, SRResNet, and MSRN achieve PSNR/SSIM gains at most 0.29dB/0.0116, 0.22dB/0.0089, 0.09dB/0.0049, and 0.05dB/0.0043, respectively.

4.3. Analysis Experiments

Effectiveness on State-of-the-Art Lightweight Model. The comparison experiments have shown the effects of FPL on different capacity models. As a supplementary, we conduct the evaluations on BSBN [30], which is the first place in the model complexity track of NTIRE 2022 Efficient SR Challenge. The results are shown in Tab. 4, which shows the attractive performance improvements of BSBN from FPL.

Ablation Studies and Parameter Analyses. To investigate the parameters in the weighting functions, we change one of them while remaining the others unchanged. Specifically, we first change $\gamma_{sp}$ and $\gamma_{lp}$ with 0, 1, 2, where $\gamma_{sp} = 0$...
Table 1. Quantitative results on 4 × SR task. “+FPL” indicates the corresponding model trained with FPL on the same settings, and “Gains” denotes the performance improvement from it.

<table>
<thead>
<tr>
<th>Method</th>
<th>Set5</th>
<th>Set14</th>
<th>BSD100</th>
<th>Manga109</th>
<th>Urban100</th>
<th>Test2K</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PSNR</td>
<td>SSIM</td>
<td>PSNR</td>
<td>SSIM</td>
<td>PSNR</td>
<td>SSIM</td>
</tr>
<tr>
<td>FSRCNN</td>
<td>28.71</td>
<td>0.8500</td>
<td>25.84</td>
<td>0.7389</td>
<td>25.58</td>
<td>0.7122</td>
</tr>
<tr>
<td>+FPL</td>
<td>29.00</td>
<td>0.8565</td>
<td>26.01</td>
<td>0.7446</td>
<td>25.70</td>
<td>0.7178</td>
</tr>
<tr>
<td>Gains</td>
<td>0.29</td>
<td>0.0065</td>
<td>0.17</td>
<td>0.0057</td>
<td>0.12</td>
<td>0.0056</td>
</tr>
<tr>
<td>CARN</td>
<td>29.94</td>
<td>0.8737</td>
<td>26.61</td>
<td>0.7605</td>
<td>25.58</td>
<td>0.7178</td>
</tr>
<tr>
<td>+FPL</td>
<td>30.11</td>
<td>0.8765</td>
<td>26.69</td>
<td>0.7641</td>
<td>25.73</td>
<td>0.7238</td>
</tr>
<tr>
<td>Gains</td>
<td>0.17</td>
<td>0.0028</td>
<td>0.08</td>
<td>0.0036</td>
<td>0.09</td>
<td>0.0042</td>
</tr>
<tr>
<td>SRResNet</td>
<td>30.13</td>
<td>0.8771</td>
<td>26.84</td>
<td>0.7672</td>
<td>26.25</td>
<td>0.7361</td>
</tr>
<tr>
<td>+FPL</td>
<td>30.27</td>
<td>0.8795</td>
<td>26.85</td>
<td>0.7677</td>
<td>26.27</td>
<td>0.7376</td>
</tr>
<tr>
<td>Gains</td>
<td>0.14</td>
<td>0.0024</td>
<td>0.10</td>
<td>0.0029</td>
<td>0.05</td>
<td>0.0034</td>
</tr>
<tr>
<td>SwinIR</td>
<td>30.34</td>
<td>0.8795</td>
<td>26.87</td>
<td>0.7670</td>
<td>26.27</td>
<td>0.7365</td>
</tr>
<tr>
<td>+FPL</td>
<td>30.39</td>
<td>0.8805</td>
<td>26.91</td>
<td>0.7688</td>
<td>26.29</td>
<td>0.7393</td>
</tr>
<tr>
<td>Gains</td>
<td>0.05</td>
<td>0.0010</td>
<td>0.04</td>
<td>0.0018</td>
<td>0.02</td>
<td>0.0028</td>
</tr>
</tbody>
</table>

Table 2. Quantitative results on 2 × SR task. “+FPL” indicates the corresponding model trained with FPL on the same settings, and “Gains” denotes the performance improvement from it.

<table>
<thead>
<tr>
<th>Method</th>
<th>Set5</th>
<th>Set14</th>
<th>BSD100</th>
<th>Manga109</th>
<th>Urban100</th>
<th>Test2K</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PSNR</td>
<td>SSIM</td>
<td>PSNR</td>
<td>SSIM</td>
<td>PSNR</td>
<td>SSIM</td>
</tr>
<tr>
<td>FSRCNN</td>
<td>34.74</td>
<td>0.9456</td>
<td>30.56</td>
<td>0.8944</td>
<td>30.01</td>
<td>0.8909</td>
</tr>
<tr>
<td>+FPL</td>
<td>34.97</td>
<td>0.9468</td>
<td>30.74</td>
<td>0.8967</td>
<td>30.21</td>
<td>0.8948</td>
</tr>
<tr>
<td>Gains</td>
<td>0.23</td>
<td>0.0012</td>
<td>0.18</td>
<td>0.0023</td>
<td>0.20</td>
<td>0.0039</td>
</tr>
<tr>
<td>CARN</td>
<td>35.30</td>
<td>0.9489</td>
<td>31.14</td>
<td>0.9007</td>
<td>30.47</td>
<td>0.8977</td>
</tr>
<tr>
<td>+FPL</td>
<td>35.48</td>
<td>0.9497</td>
<td>31.30</td>
<td>0.9031</td>
<td>30.61</td>
<td>0.9006</td>
</tr>
<tr>
<td>Gains</td>
<td>0.18</td>
<td>0.0008</td>
<td>0.16</td>
<td>0.0024</td>
<td>0.14</td>
<td>0.0029</td>
</tr>
<tr>
<td>SRResNet</td>
<td>35.48</td>
<td>0.9496</td>
<td>31.26</td>
<td>0.9018</td>
<td>30.51</td>
<td>0.8994</td>
</tr>
<tr>
<td>+FPL</td>
<td>35.62</td>
<td>0.9503</td>
<td>31.48</td>
<td>0.9041</td>
<td>30.74</td>
<td>0.9021</td>
</tr>
<tr>
<td>Gains</td>
<td>0.14</td>
<td>0.0007</td>
<td>0.22</td>
<td>0.0023</td>
<td>0.23</td>
<td>0.0027</td>
</tr>
<tr>
<td>MSRN</td>
<td>35.66</td>
<td>0.9506</td>
<td>31.55</td>
<td>0.9049</td>
<td>30.75</td>
<td>0.9012</td>
</tr>
<tr>
<td>+FPL</td>
<td>35.67</td>
<td>0.9507</td>
<td>31.62</td>
<td>0.9065</td>
<td>30.75</td>
<td>0.9023</td>
</tr>
<tr>
<td>Gains</td>
<td>0.01</td>
<td>0.0001</td>
<td>0.07</td>
<td>0.0016</td>
<td>0.00</td>
<td>0.0011</td>
</tr>
<tr>
<td>SwinIR</td>
<td>35.70</td>
<td>0.9507</td>
<td>31.55</td>
<td>0.9050</td>
<td>30.75</td>
<td>0.9010</td>
</tr>
<tr>
<td>+FPL</td>
<td>35.73</td>
<td>0.9509</td>
<td>31.59</td>
<td>0.9059</td>
<td>30.75</td>
<td>0.9022</td>
</tr>
<tr>
<td>Gains</td>
<td>0.03</td>
<td>0.0002</td>
<td>0.04</td>
<td>0.0009</td>
<td>0.00</td>
<td>0.0012</td>
</tr>
</tbody>
</table>

or $\gamma_{lp} = 0$ denotes to remove one of the weighting function, and $\gamma_{sp} = 0$, $\gamma_{lp} = 0$ disables both of the weighting functions, while BI is remained for under-sampling. The results are shown in Fig. 4, from which one could see that the indispensable roles of them. Note that as the BI under-sampling revises the pixel distribution to a less extremely long-tailed one, it obtains better performance gains than the priors which balance the fitting degree of the model for improvements. But generally, the best performance is obtained by integrating them together. Second, as $\alpha_{sp}$ and $\alpha_{lp}$ should be greater than zero, we change them with 0.1, 0.5, 1, 2. As shown in Fig. 4, there is no very obvious performance changes among different parameters’ settings. Overall, although the parameters affect the performance, FPL could achieve the consistent improvements.

Weighting matrix visualizations. We visualize weighting matrices from the two structure priors and their element-wise product for observing the attentions on HR images. As shown in Fig. 5, the weighting matrix (b) from the static structure prior has large values in the high-frequency regions, while that (c) from the learnable structure prior has large values on the hard pixels in high-frequency regions.
Table 3. Quantitative results on 4× SR task with multiple degradations. “+FPL” indicates the corresponding model trained with FPL on the same settings, and “Gains” denotes the performance improvement from it.

<table>
<thead>
<tr>
<th>Method</th>
<th>Set5 PSNR</th>
<th>Set5 SSIM</th>
<th>Set14 PSNR</th>
<th>Set14 SSIM</th>
<th>BSD100 PSNR</th>
<th>BSD100 SSIM</th>
<th>Manga109 PSNR</th>
<th>Manga109 SSIM</th>
<th>Urban100 PSNR</th>
<th>Urban100 SSIM</th>
<th>Test2K PSNR</th>
<th>Test2K SSIM</th>
</tr>
</thead>
<tbody>
<tr>
<td>FSRCNN</td>
<td>24.73</td>
<td>0.6859</td>
<td>23.13</td>
<td>0.5809</td>
<td>23.45</td>
<td>0.5598</td>
<td>22.08</td>
<td>0.6861</td>
<td>20.77</td>
<td>0.5640</td>
<td>23.55</td>
<td>0.6129</td>
</tr>
<tr>
<td>+FPL</td>
<td>24.91</td>
<td>0.6957</td>
<td>23.23</td>
<td>0.5869</td>
<td>23.54</td>
<td>0.5658</td>
<td>22.37</td>
<td>0.6977</td>
<td>20.89</td>
<td>0.5737</td>
<td>23.57</td>
<td>0.6140</td>
</tr>
<tr>
<td>Gains</td>
<td>0.18</td>
<td>0.0098</td>
<td>0.10</td>
<td>0.0060</td>
<td>0.09</td>
<td>0.0060</td>
<td>0.29</td>
<td>0.0116</td>
<td>0.12</td>
<td>0.0097</td>
<td>0.02</td>
<td>0.0011</td>
</tr>
<tr>
<td>CARN</td>
<td>25.69</td>
<td>0.7330</td>
<td>23.91</td>
<td>0.6188</td>
<td>23.90</td>
<td>0.5865</td>
<td>23.84</td>
<td>0.7588</td>
<td>21.60</td>
<td>0.6210</td>
<td>23.64</td>
<td>0.6309</td>
</tr>
<tr>
<td>+FPL</td>
<td>25.80</td>
<td>0.7381</td>
<td>23.99</td>
<td>0.6236</td>
<td>23.96</td>
<td>0.5912</td>
<td>24.06</td>
<td>0.7668</td>
<td>21.72</td>
<td>0.6299</td>
<td>23.63</td>
<td>0.6338</td>
</tr>
<tr>
<td>Gains</td>
<td>0.11</td>
<td>0.0051</td>
<td>0.08</td>
<td>0.0048</td>
<td>0.06</td>
<td>0.0047</td>
<td>0.22</td>
<td>0.0080</td>
<td>0.12</td>
<td>0.0089</td>
<td>0.02</td>
<td>0.0011</td>
</tr>
<tr>
<td>SRResNet</td>
<td>25.86</td>
<td>0.7403</td>
<td>24.08</td>
<td>0.6256</td>
<td>23.99</td>
<td>0.5918</td>
<td>24.29</td>
<td>0.7722</td>
<td>21.89</td>
<td>0.6382</td>
<td>23.61</td>
<td>0.6320</td>
</tr>
<tr>
<td>+FPL</td>
<td>25.95</td>
<td>0.7431</td>
<td>24.08</td>
<td>0.6276</td>
<td>24.02</td>
<td>0.5952</td>
<td>24.38</td>
<td>0.7760</td>
<td>21.92</td>
<td>0.6431</td>
<td>23.52</td>
<td>0.6337</td>
</tr>
<tr>
<td>Gains</td>
<td>0.09</td>
<td>0.0028</td>
<td>0.00</td>
<td>0.0020</td>
<td>0.03</td>
<td>0.0034</td>
<td>0.09</td>
<td>0.0038</td>
<td>0.03</td>
<td>0.0049</td>
<td>-0.09</td>
<td>0.0017</td>
</tr>
<tr>
<td>MSRN</td>
<td>26.01</td>
<td>0.7459</td>
<td>24.09</td>
<td>0.6266</td>
<td>24.02</td>
<td>0.5932</td>
<td>24.40</td>
<td>0.7757</td>
<td>21.95</td>
<td>0.6412</td>
<td>23.59</td>
<td>0.6324</td>
</tr>
<tr>
<td>+FPL</td>
<td>26.05</td>
<td>0.7480</td>
<td>24.10</td>
<td>0.6291</td>
<td>24.03</td>
<td>0.5958</td>
<td>24.45</td>
<td>0.7782</td>
<td>21.97</td>
<td>0.6455</td>
<td>23.48</td>
<td>0.6331</td>
</tr>
<tr>
<td>Gains</td>
<td>0.04</td>
<td>0.0021</td>
<td>0.01</td>
<td>0.0025</td>
<td>0.01</td>
<td>0.0026</td>
<td>0.05</td>
<td>0.0025</td>
<td>0.02</td>
<td>0.0043</td>
<td>-0.11</td>
<td>0.0007</td>
</tr>
</tbody>
</table>

Figure 3. Qualitative and quantitative (PSNR/SSIM) results on 4× SR task.

As a result, their element-wise product (d) integrates the two weighting matrices to down-weight the pixels in the low-frequency regions, while up-weighting the pixels in the high-frequency regions, in which the hard pixels are with larger weight values than the easy ones. For example, the edges of the “hat” are high-frequency regions according to the weighting matrix (b), and relatively easy to be super-resolved in terms of the weighting matrix (c), thus they are
Table 4. Quantitative results of BSRN [30] on $4 \times$ SR task.

<table>
<thead>
<tr>
<th>Method</th>
<th>Set5</th>
<th>Urban100</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PSNR</td>
<td>SSIM</td>
</tr>
<tr>
<td>BSRN</td>
<td>32.35</td>
<td>0.8966</td>
</tr>
<tr>
<td>+FPL</td>
<td>32.50</td>
<td>0.8972</td>
</tr>
<tr>
<td>Gains</td>
<td>0.15</td>
<td>0.0006</td>
</tr>
</tbody>
</table>

Figure 4. Ablation studies and parameter analyses on Manga109. The left column is the results of changing $\gamma_{sp}$ and $\gamma_{lp}$, and the right column is the results of changing $\alpha_{sp}$ and $\alpha_{lp}$, where PSNR/SSIM gain denotes the performance improvements over FSRCNN.

Figure 5. The weighting matrices from the two structure priors and their element-wise product, which show significant attention on the pixels in the high-frequency regions.

Figure 6. Attribution results with diffusion index (DI) w.r.t. FSRCNN (top) and FSRCNN+FPL (bottom). From left to right respectively are HR images, receptive fields, attribution results, attribution pixels, and SR results. A higher DI represents that more pixels significantly contribute to the SR result of the given patch.

5. Conclusion

In this work, we propose a novel understanding, i.e., the twin fitting problem arises from the long-tailed pixel distribution, to explain why the high-frequency regions are harder to be super-resolved than the low-frequency ones. Inspired by the explanation, we first reformulate SR as a long-tailed distribution learning problem, and design a simple and effective solution by introducing two structure priors and an under-sampling strategy. Such a reformulation and solution enjoy the advantages of high interpretability and model agnostic, and the extensive experiments have demonstrated the superiority of them.

Acknowledgments

This work was supported in part by NSFC under Grant U21B2040, 62176171, and U19A2078; in part by Sichuan Science and Technology Planning Project under Grant 2022YFQ0014. A*STAR AME Programmatic Funding A18A2b0046, RobotHTPO Seed Fund under Project C211518008 and EDB Space Technology Development Grant under Project S22-19016-STDP.


