

# Meta-Transfer Learning for Zero-Shot Super-Resolution

Jae Woong Soh Sunwoo Cho Nam Ik Cho

Department of ECE, INMC, Seoul National University, Seoul, Korea

{soh90815, etoo33}@snu.ac.kr, nicho@snu.ac.kr

## Abstract

Convolutional neural networks (CNNs) have shown dramatic improvements in single image super-resolution (SISR) by using large-scale external samples. Despite their remarkable performance based on the external dataset, they cannot exploit internal information within a specific image. Another problem is that they are applicable only to the specific condition of data that they are supervised. For instance, the low-resolution (LR) image should be a “bicubic” downsampled noise-free image from a high-resolution (HR) one. To address both issues, zero-shot super-resolution (ZSSR) has been proposed for flexible internal learning. However, they require thousands of gradient updates, i.e., long inference time. In this paper, we present Meta-Transfer Learning for Zero-Shot Super-Resolution (MZSR), which leverages ZSSR. Precisely, it is based on finding a generic initial parameter that is suitable for internal learning. Thus, we can exploit both external and internal information, where **one single gradient update** can yield quite considerable results. (See Figure 1). With our method, the network can quickly adapt to a given image condition. In this respect, our method can be applied to a large spectrum of image conditions within a fast adaptation process.

## 1. Introduction

SISR, which is to find a plausible HR image from its counterpart LR image, is a long-standing problem in low-level vision area. Recently, the remarkable success of CNNs brought attention to the research community, and hence numerous CNN-based SISR methods have exhibited large performance leap [15, 17, 21, 47, 2, 45, 36, 20, 12, 13]. Most of the recent state-of-the-art (SotA) CNN-based methods are based on a large number of external training dataset and self-supervised settings with *known* degradation model, e.g., “bicubic” downsampling. Impressively, the recent SotA CNNs show significant PSNR gains compared to the conventional large size of models for the noise-free “bicubic” downsampling condition. However, in real-world sit-

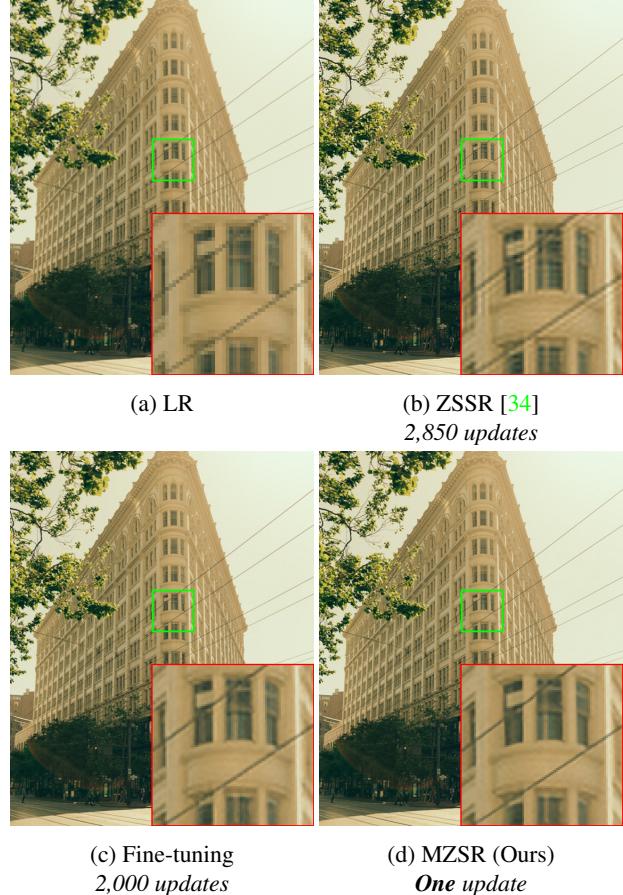


Figure 1: Super-resolved results ( $\times 2$ ) of “img050” in Urban100 [14]. The blur kernel of the LR image is an isotropic Gaussian kernel with width 2.0. Result of (c) is fine-tuned from a pre-trained model. Our MZSR outperforms other methods within just **one** single gradient descent update.

uations, when the LR image has distant statistics in downsampling kernels and noises, the recent methods produce undesirable artifacts and show inferior results due to the domain gap. Moreover, their number of parameters and memory overheads are usually too large to be used in real appli-

cations.

Besides, non-local self-similarity in scale and across multi-scale, which is the internal recurrence of information within a single image, is one of the strong natural image priors. Therefore it has long been used in image restoration tasks, including image denoising [5, 6] and super-resolution [24, 14]. Additionally, the powerful image prior of non-local property is embedded into network architecture [19, 22, 46] by implicitly learning such priors to boost the performance of the networks further. Also, some works to learn internal distribution have been proposed [34, 32, 33]. Moreover, there have been many studies to combine the advantages of external and internal information for image restoration [26, 43, 42, 41].

Recently, ZSSR [34] has been proposed for zero-shot super-resolution, which is based on the zero-shot setting to exploit the power of CNN but can be easily adapted to the test image condition. Interestingly, ZSSR learns the internal non-local structure of the test image, *i.e.*, deep internal learning. Thus it outperforms external-based CNNs in some regions where the recurrences are salient. Also, ZSSR is highly flexible that it can address any blur kernels, and thus easily adapted to the conditions of test images.

However, ZSSR has a few limitations. First, it requires thousands of backpropagation gradient updates at test time, which requires considerable time to get the result. Also, it cannot fully exploit the large-scale external dataset, and rather it depends only on internal structure and patterns, which lacks in the number of total examples. Eventually, this leads to inferior results in most of the regions with general patterns compared to the external-based methods.

On the other hand, meta-learning or learning to learn fast has recently attracted many researchers. Meta-learning aims to address a problem that artificial intelligence is hard to learn new concepts quickly with a few examples, unlike human intelligence. In this respect, meta-learning is jointly merged with few-shot learning, and many methods with this approach have been proposed [35, 39, 38, 28, 25, 8, 10, 18, 37]. Among them, Model-Agnostic Meta-Learning (MAML) [8] has shown great impact, showing SotA performance by learning the optimal initial state of the model such that the base-learner can fast adapt to a new task within a few gradient steps. MAML employs the gradient update as meta-learner, and the same author analyzed that gradient descent can approximate any learning algorithm [9]. Moreover, Sun *et al.* [37] have jointly utilized MAML with transfer learning to exploit large-scale data for few-shot learning.

Inspired by the above-stated works and ZSSR, we present Meta-Transfer Learning for Zero-Shot Super-Resolution (MZSR), which is kernel-agnostic. We found that simply employing transfer learning or fine-tuning from a pre-trained network does not yield plausible results. As ZSSR only has a meta-test step, we additionally adopt a

meta-training step to make the model adapt fast to new blur kernel scenarios. Additionally, we adopt transfer learning in advance to fully utilize external samples, further leveraging the performance. In particular, transfer learning with the help of a large-scale synthetic dataset (“bicubic” degradation setting) is first performed for the external learning of natural image priors. Then, meta-learning plays a role in learning task-level knowledge with different downsampling kernels as different tasks. At the meta-test step, simple self-supervised learning is conducted to learn image-specific information within a few gradient steps. As a result, we can exploit both external and internal information. Also, by leveraging the advantages of ZSSR, we may use a lightweight network, which is flexible to different degradation conditions of LR images. Furthermore, our method is much faster than ZSSR, *i.e.*, it quickly adapts to new tasks within a few gradient steps, while ZSSR requires thousands of updates.

In summary, our overall contribution is three-fold:

- We present a novel training scheme based on meta-transfer learning, which learns an effective initial weight for fast adaptation to new tasks with the zero-shot unsupervised setting.
- By using external and internal samples, it is possible to leverage the advantages of both internal and external learning.
- Our method is fast, flexible, lightweight and unsupervised at meta-test time, hence, eventually can be applied to real-world scenarios.

## 2. Related Work

### 2.1. CNN-based Super-Resolution

SISR is based on the image degradation model as

$$\mathbf{I}_{LR}^k = (\mathbf{I}_{HR} * \mathbf{k}) \downarrow_s + \mathbf{n}, \quad (1)$$

where  $\mathbf{I}_{HR}$ ,  $\mathbf{I}_{LR}^k$ ,  $\mathbf{k}$ ,  $*$ ,  $\downarrow_s$ , and  $\mathbf{n}$  denote HR, LR image, blur kernel, convolution, decimation with scaling factor of  $s$ , and white Gaussian noise, respectively. It is notable that diverse degraded conditions can be found in real-world scenes, with various unknown  $\mathbf{k}$ ,  $\downarrow_s$ , and  $\mathbf{n}$ .

Recently, numerous CNN-based networks have been proposed to super-resolve LR image with *known* downsampling kernel [15, 17, 21, 12, 47, 2, 36, 20, 13]. They show extreme performances in “bicubic” downsampling scenarios but suffer in non-bicubic cases due to the domain gap. To cope with multiple degradation kernels, SRMD [44] has been proposed. With additional inputs of kernel and noise information, SRMD outperforms other SISR methods in non-bicubic conditions. Also, IKC [11] has been proposed

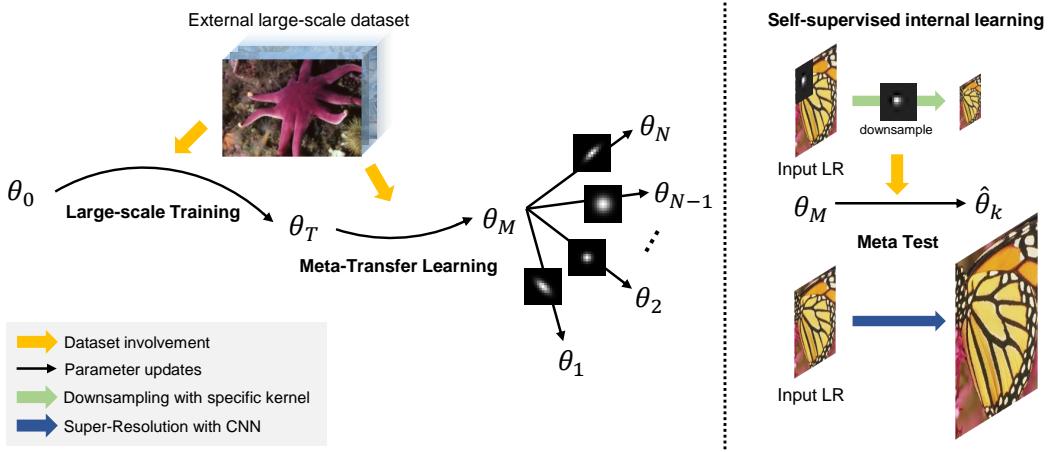


Figure 2: The overall scheme of our proposed MZSR. During meta-transfer learning, the external dataset is used, where internal learning is done during meta-test time. From random initial point  $\theta_0$ , large-scale dataset DIV2K [1] with “bicubic” degradation is exploited to obtain  $\theta_T$ . Then, meta-transfer learning learns a good representation  $\theta_M$  for super-resolution tasks with diverse blur kernel scenarios. The figure shows  $N$  tasks for simplicity. In the meta-test phase, self-supervision within a test image is exploited to train the model with corresponding blur kernel.

for blind super-resolution. On the other hand, ZSSR [34] has been proposed to learn image specific internal structure with CNN, and has shown that it can be applied to real-world scenes due to its flexibility.

## 2.2. Meta-Learning

In recent years, diverse meta-learning algorithms have been proposed. They can be categorized into three groups. The first group is *metric based methods* [35, 38, 39], which is to learn metric space in which learning is efficient within a few samples. The second group is *memory network-based methods* [31, 28, 25], where the network learns across task knowledges and well generalizes to unseen tasks. The last group is *optimization based methods*, where gradient descent plays a role as a meta-learner optimization [10, 18, 9, 8]. Among them, MAML [8] has shown a great impact on the research community, and several variants have been proposed [27, 37, 3, 30]. MAML inherently requires second-order derivative terms, and the first-order algorithm has also been proposed in [27]. Also, to cope with the instability of MAML training, MAML++ [3] has been proposed. Moreover, MAML within embedded space has been proposed [30]. In this paper, we employ MAML scheme for fast adaptation of zero-shot super-resolution.

## 3. Preliminary

We introduce self-supervised zero-shot super-resolution and meta-learning schemes with notations, following related works [34, 8].

**Zero-Shot Super-Resolution** ZSSR [34] is totally unsupervised or self-supervised. Two phases of training and test are both held in runtime. In training phase, the test image  $\mathbf{I}_{LR}$  is downsampled with desired kernel to generate “LR son” denoted as  $\mathbf{I}_{son}$ , and  $\mathbf{I}_{LR}$  becomes the HR supervision, “HR father.” Then, the CNN is trained with the LR-HR pairs generated by a single image. The training solely depends on the test image, thus learns specific internal information to given image statistics. In the test phase, the trained CNN then works as a feedforward network, and the test input image is fed to the CNN to get the super-resolved image  $\mathbf{I}_{SR}$ .

**Meta-Learning** Meta-learning has two phases: meta-training and meta-test. We consider a model  $f_\theta(\cdot)$ , which is parameterized by  $\theta$ , that maps inputs  $\mathbf{x}$  to outputs  $\mathbf{y}$ . The goal of meta-training is to make the model to be able to adapt to a large number of different tasks. A task  $\mathcal{T}_i$  is sampled from a task distribution  $p(\mathcal{T})$  for meta-training. Within a task, training samples are used to optimize the base-learner with a task-specific loss  $\mathcal{L}_{\mathcal{T}_i}$  and test samples are used to optimize the meta-learner. In meta-test phase, the model  $f_\theta(\cdot)$  quickly adapts to a new task  $\mathcal{T}_{new}$  with the help of meta-learner. MAML [8] employs a simple gradient descent algorithm as the meta-learner and seeks to find an initial transferable point where a few gradient updates lead to a fast adaptation of the model to a new task.

In our case, the input  $\mathbf{x}$  and the output  $\mathbf{y}$  are  $\mathbf{I}_{LR}^k$  and  $\mathbf{I}_{SR}$ . Also, diverse blur kernels constitute the task distribution, where each task corresponds to the super-resolution of

an image degraded by a specific blur kernel.

## 4. Method

The overall scheme of our proposed MZSR is shown in Figure 2. As shown, our method consists of three steps: large-scale training, meta-transfer learning, and meta-test.

### 4.1. Large-scale Training

This step is similar to the large-scale ImageNet [7] pre-training for object recognition. In our case, we adopt DIV2K [1] which is a high-quality dataset  $\mathcal{D}_{HR}$ . Using *known* “bicubic” degradation, we first synthesized large number of paired dataset  $(\mathbf{I}_{HR}, \mathbf{I}_{LR}^{bic})$ , denoted as  $\mathcal{D}$ . Then, we trained the network to learn super-resolution of “bicubic” degradation model by minimizing the loss,

$$\mathcal{L}^{\mathcal{D}}(\theta) = \mathbb{E}_{\mathcal{D} \sim (\mathbf{I}_{HR}, \mathbf{I}_{LR}^{bic})} [||\mathbf{I}_{HR} - f_{\theta}(\mathbf{I}_{LR}^{bic})||_1], \quad (2)$$

which is the pixel-wise L1 loss [21, 34] between prediction and the ground-truth.

The large-scale training has contributions within two respects. First, as super-resolution tasks share similar properties, it is possible to learn efficient representations that implicitly represent natural image priors of high-resolution images, thus making the network ease to be learned. Second, as MAML [8] is known to show some unstable training, we ease the training phase of meta-learning with the help of well pre-trained feature representations.

### 4.2. Meta-Transfer Learning

Since ZSSR is trained with the gradient descent algorithm, it is possible to introduce an *optimization-based* meta-training step with the help of gradient descent algorithm, which is proven to be a universal learning algorithm [9].

In this step, we seek to find a sensitive and transferable initial point of the parameter space where a few gradient updates lead to large performance improvements. Inspired by MAML, our algorithm mostly follows MAML but with several modifications.

Unlike MAML, we adopt different settings for meta-training and meta-test. In particular, we use the external dataset for meta-training, whereas internal learning is adopted for meta-test. This is because we intend our meta-learner to more focus on the kernel-agnostic property with the help of a large-scale external dataset.

We synthesize dataset for meta-transfer learning, denoted as  $\mathcal{D}_{meta}$ .  $\mathcal{D}_{meta}$  consists of pairs,  $(\mathbf{I}_{HR}, I_{LR}^k)$ , with diverse kernel settings. Specifically, we used isotropic and anisotropic Gaussian kernels for the blur kernels. We consider a kernel distribution  $p(\mathbf{k})$ , where each kernel is determined by a covariance matrix  $\Sigma$ . It is chosen to have a random angle  $\Theta \sim U[0, \pi]$ , and two random eigenvalues

$\lambda_1 \sim U[1, 2.5s]$ ,  $\lambda_2 \sim U[1, \lambda_1]$  where  $s$  denotes the scaling factor. Precisely, the covariance matrix is expressed as

$$\Sigma = \begin{bmatrix} \cos(\Theta) & -\sin(\Theta) \\ \sin(\Theta) & \cos(\Theta) \end{bmatrix} \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} \begin{bmatrix} \cos(\Theta) & \sin(\Theta) \\ -\sin(\Theta) & \cos(\Theta) \end{bmatrix}. \quad (3)$$

Eventually, we train our meta-learner based on  $\mathcal{D}_{meta}$ . We may divide  $\mathcal{D}_{meta}$  into two groups:  $\mathcal{D}_{tr}$  for task-level training, and  $\mathcal{D}_{te}$  for task-level test.

In our method, adaptation to a new task  $\mathcal{T}_i$  with respect to the parameters  $\theta$  is one or more gradient descent updates. For one gradient update, new adapted parameters  $\theta_i$  is then

$$\theta_i = \theta - \alpha \nabla_{\theta} \mathcal{L}_{\mathcal{T}_i}^{tr}(\theta), \quad (4)$$

where  $\alpha$  is the task-level learning rate. The model parameters  $\theta$  are optimized to achieve minimal test error of  $\mathcal{D}_{meta}$  with respect to  $\theta_i$ . Concretely, the meta-objective is

$$\arg \min_{\theta} \sum_{\mathcal{T}_i \sim p(\mathcal{T})} \mathcal{L}_{\mathcal{T}_i}^{te}(\theta_i) \quad (5)$$

$$= \arg \min_{\theta} \sum_{\mathcal{T}_i \sim p(\mathcal{T})} \mathcal{L}_{\mathcal{T}_i}^{te}(\theta - \alpha \nabla_{\theta} \mathcal{L}_{\mathcal{T}_i}^{tr}(\theta)). \quad (6)$$

Meta-transfer optimization is performed using Eq. 6, which is to learn the knowledge across task. Any gradient-based optimization can be used for meta-transfer training. For stochastic gradient descents, the parameter update rule is expressed as

$$\theta \leftarrow \theta - \beta \nabla_{\theta} \sum_{\mathcal{T}_i \sim p(\mathcal{T})} \mathcal{L}_{\mathcal{T}_i}^{te}(\theta_i), \quad (7)$$

where  $\beta$  is the meta-learning rate.

### 4.3. Meta-Test

The meta-test step is exactly the zero-shot super-resolution. As evidence in [34], this step enables our model to learn internal information within a single image. With a given LR image, we downsample it with corresponding downsampling kernel (kernel estimation algorithms [24, 29] can be adopted for blind scenario) to generate  $\mathbf{I}_{son}$  and perform a few gradient updates with respect to the model parameter using a single pair of “LR son” and a given image. Then, we feed a given LR image to the model to get a super-resolved image.

### 4.4. Algorithm

Algorithm 1 demonstrates the process of our meta-transfer training procedures of Section 4.1 and 4.2. Lines 3-7 is the large-scale training stage. Lines 11-14 is the inner loop of meta-transfer learning where the base-learner is updated to task-specific loss. Lines 15-16 presents the meta-learner optimization.

---

**Algorithm 1:** Meta-Transfer Learning

---

**Input:** High-resolution dataset  $\mathcal{D}_{HR}$  and blur kernel distribution  $p(\mathbf{k})$   
**Input:**  $\alpha, \beta$ : learning rates  
**Output:** Model parameter  $\theta_M$

- 1 Randomly initialize  $\theta$
- 2 Synthesize paired dataset  $\mathcal{D}$  by bicubicly downsample  $\mathcal{D}_{HR}$
- 3 **while** not done **do**
- 4   Sample LR-HR batch from  $\mathcal{D}$
- 5   Evaluate  $\mathcal{L}^D$  by Eq. 2
- 6   Update  $\theta$  with respect to  $\mathcal{L}^D$
- 7 **end**
- 8 Generate task distribution  $p(\mathcal{T})$  with  $\mathcal{D}_{HR}$  and  $p(\mathbf{k})$
- 9 **while** not done **do**
- 10   Sample task batch  $\mathcal{T}_i \sim p(\mathcal{T})$
- 11   **for** all  $\mathcal{T}_i$  **do**
- 12     Evaluate training loss ( $\mathcal{D}_{tr}$ ):  $\mathcal{L}_{\mathcal{T}_i}^{tr}(\theta)$
- 13     Compute adapted parameters with gradient descent:  $\theta_i = \theta - \alpha \nabla_{\theta} \mathcal{L}_{\mathcal{T}_i}^{tr}(\theta)$
- 14   **end**
- 15   Update  $\theta$  with respect to average test loss ( $\mathcal{D}_{te}$ ):  
$$\theta \leftarrow \theta - \beta \nabla_{\theta} \sum_{\mathcal{T}_i \sim p(\mathcal{T})} \mathcal{L}_{\mathcal{T}_i}^{te}(\theta_i)$$
- 17 **end**

---

---

**Algorithm 2:** Meta-Test

---

**Input:** LR test image  $\mathbf{I}_{LR}$ , meta-transfer trained model parameter  $\theta_M$ , number of gradient updates  $n$  and learning rate  $\alpha$   
**Output:** Super-resolved image  $\mathbf{I}_{SR}$

- 1 Initialize model parameter  $\theta$  with  $\theta_M$
- 2 Generate LR son  $\mathbf{I}_{son}$  by downsampling  $\mathbf{I}_{LR}$  with corresponding blur kernel.
- 3 **for**  $n$  steps **do**
- 4   Evaluate loss  $\mathcal{L}(\theta) = \|\mathbf{I}_{LR} - f_{\theta}(\mathbf{I}_{son})\|_1$
- 5   Update  $\theta \leftarrow \theta - \alpha \nabla_{\theta} \mathcal{L}(\theta)$
- 6 **end**
- 7 **return**  $\mathbf{I}_{SR} = f_{\theta}(\mathbf{I}_{LR})$

---

Algorithm 2 presents the meta-test step, which is the zero-shot super-resolution. A few gradient updates ( $n$ ) are performed while meta-test, and the super-resolved image is obtained with final updated parameters.

## 5. Experiments

### 5.1. Training Details

For the CNN, we adopt a simple 8-layer CNN architecture with residual learning following ZSSR [34]. Its number of parameters is 225 K. For meta-transfer training, we use

DIV2K [1] for the high-quality dataset and we set  $\alpha = 0.01$  and  $\beta = 0.0001$  for entire training. For the inner loop, we conducted 5 gradient updates, *i.e.* 5 unrolling steps, to obtain adapted parameters. We extracted training patches with a size of  $64 \times 64$ . To cope with gradient vanishing or exploding problems due to the unrolling process of base learners, we utilize the weighted sum of losses from each step, *i.e.*, providing supervision of additional losses to each unrolling step [3]. At the initial point, we evenly weigh the losses and decayed the weights except for the last unrolling step. In the end, the weighted loss converges to our final training task loss. We employ ADAM [16] optimizer as our meta-optimizer. As the subsampling process ( $\downarrow_s$ ) can be the *direct* method [34] or the *bicubic* subsampling [44, 11], we trained two models for different subsampling methods: *direct* and *bicubic*.

### 5.2. Evaluations on “Bicubic” Downsampling

We evaluate our method with several recent SotA SISR methods, including supervised and unsupervised methods on famous benchmarks: Set5 [4], BSD100 [23], and Urban100 [14]. We measure PSNR and SSIM [40] in Y-channel of YCbCr colorspace.

The overall results are shown in Table 1. CARN [2] and RCAN [45], which are trained for “bicubic” downsampling condition, show extremely overwhelming performances. Since the training scenario and the test scenario exactly match each other, supervision on external samples could boost the performance of CNN. On the other hands, ZSSR [34] and our methods show improvements against bicubic interpolation but not as good as the supervised ones, because both methods are trained within the unsupervised or self-supervised regime. Our methods show comparable results to ZSSR within only *one* single gradient descent update.

### 5.3. Evaluations on Various Blur Kernels

In this section, we demonstrate the results on various blur kernel conditions. We assume four scenarios: severe aliasing, isotropic Gaussian, unisotropic Gaussian, and isotropic Gaussisan followed by *bicubic* subsampling. Precisely, the methods are

- $g_{0,2}^d$ : isotropic Gaussian blur kernel with width  $\lambda = 0.2$  followed by *direct* subsampling.
- $g_{2,0}^d$ : isotropic Gaussian blur kernel with width  $\lambda = 2.0$  followed by *direct* subsampling.
- $g_{ani}^d$ : anisotropic Gaussian with widths  $\lambda_1 = 4.0$  and  $\lambda_2 = 1.0$  with  $\Theta = -0.5$  from Eq. 3, followed by *direct* subsampling.
- $g_{1,3}^b$ : isotropic Gaussian blur kernel with width  $\lambda = 1.3$  followed by *bicubic* subsampling.

	Supervised			Unsupervised		
Dataset	Bicubic	CARN [2]	RCAN [45]	ZSSR [34]	MZSR (1)	MZSR (10)
Set5	33.64/0.9293	37.76/0.9590	38.18/0.9604	36.93/0.9554	36.77/0.9549	37.25/0.9567
BSD100	29.55/0.8427	32.09/0.8978	32.38/0.9018	31.43/0.8901	31.33/0.8910	31.64/0.8928
Urban100	26.87/0.8398	31.92/0.9256	33.30/0.9376	29.34/0.8941	30.01/0.9054	30.41/0.9092

Table 1: The average PSNR/SSIM results on “bicubic” downsampling scenario with  $\times 2$  on benchmarks. The numbers in parenthesis in our methods stand for the number of gradient updates.

	Supervised				Unsupervised		
Kernel	Dataset	Bicubic	RCAN [45]	IKC [11]	ZSSR [34]	MZSR (1)	MZSR (10)
$g_{0,2}^d$	Set5	30.24/0.8976	28.40/0.8618	29.09/0.8786	34.29/0.9373	33.14/0.9277	33.74/0.9301
	BSD100	27.45/0.7992	25.16/0.7602	26.23/0.7808	29.35/0.8465	28.74/0.8389	29.03/0.8415
	Urban100	24.70/0.7958	21.68/0.7323	23.66/0.7806	28.13/0.8788	26.24/0.8394	26.60/0.8439
$g_{2,0}^d$	Set5	28.73/0.8449	29.15/0.8601	29.05/0.8896	34.90/0.9397	35.20/0.9398	36.05/0.9439
	BSD100	26.51/0.7157	26.89/0.7394	27.46/0.8156	30.57/0.8712	30.58/0.8627	31.09/0.8739
	Urban100	23.70/0.7109	24.14/0.7384	25.17/0.8169	27.86/0.8582	28.23/0.8657	29.19/0.8838
$g_{ani}^d$	Set5	28.15/0.8265	28.42/0.8379	28.74/0.8565	33.96/0.9307	34.05/0.9271	34.78/0.9323
	BSD100	26.00/0.6891	26.22/0.7062	26.44/0.7310	29.72/0.8479	28.82/0.8013	29.54/0.8297
	Urban100	23.13/0.6796	23.35/0.6982	23.62/0.7239	27.03/0.8335	26.51/0.8126	27.34/0.8369
$g_{1,3}^b$	Set5	30.54/0.8773	31.54/0.8992	33.88/0.9357	35.24/0.9434	35.18/0.9430	36.64/0.9498
	BSD100	27.49/0.7546	28.27/0.7904	30.95/0.8860	30.74/0.8743	29.02/0.8544	31.25/0.8818
	Urban100	24.74/0.7527	25.65/0.7946	29.47/0.8956	28.30/0.8693	28.27/0.8771	29.83/0.8965

Table 2: The average PSNR/SSIM results on various kernels with  $\times 2$  on benchmarks. The numbers in parenthesis in our methods stand for the number of gradient updates. The best results are highlighted in red and the second best are in blue.

The results are shown in Table 2. As the SotA method RCAN [45] is trained on “bicubic” scenario, it shows inferior performance due to domain discrepancy and lack of flexibility.

For the case of aliasing ( $g_{0,2}^d$ ), RCAN results are even worse than a simple bicubic interpolation method due to inconsistency between training and test condition. IKC<sup>1</sup> [11] is trained for *bicubic* subsampling, it never sees aliased images during training. Thus, it also shows a severe performance drop. On the other hand, ZSSR<sup>2</sup> [34] shows quite improved results due to its flexibility. However, it requires thousands of gradient updates, which require a large amount of time. Also, it starts from a random initial point and thus does not guarantee the same results for multiple tests. As shown in Table 2, our methods are comparable to others even with one single gradient update. Interestingly, our MZSR never sees the kernel with  $\lambda = 0.2$ , but the CNN quickly adapts to specific image condition. In other words, compared to other methods, our method is more robust to extrapolation.

<sup>1</sup>We reimplemented the code and retrained with DIV2K dataset.

<sup>2</sup>We used the official code but without gradual configuration.

For other cases, which are isotropic and anisotropic Gaussian, our methods outperform others with a significantly large gap. In these cases, other methods have performance gains compared to bicubic interpolation, but the differences are minor. Similar tendencies of aliasing cases can be found in all other scenarios. Interestingly, RCAN [45] shows slightly improved results compared to bicubic interpolation. Also, as the condition between training and test is consistent, IKC [11] shows comparable results. Our methods also show remarkable performance in the case of *bicubic* subsampling condition. From the extensive experimental results, we believe that our MZSR is a fast, flexible, and accurate method for super-resolution.

#### 5.4. Real Image Super-Resolution

To show the effectiveness of the proposed MZSR, we also conduct experiments on real images. Since there are no ground-truth images for real images, we only present the visual comparisons. Due to the page limit, all the comparisons on real images are presented in *supplementary material*.

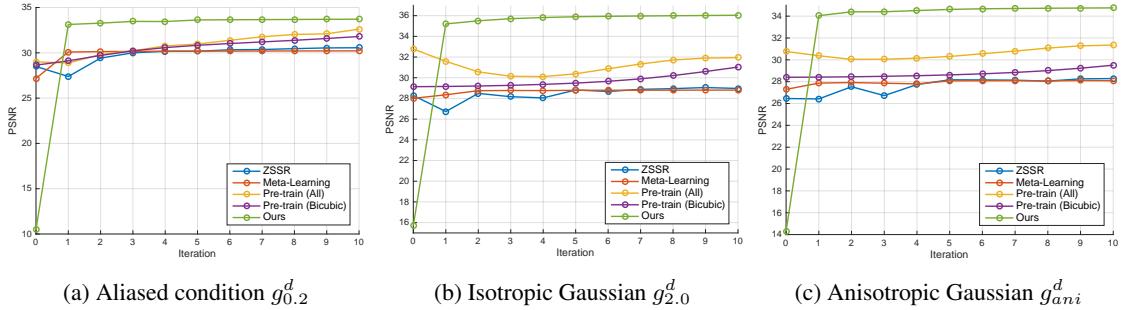


Figure 3: The average PSNR on Set5 vs. number of gradient update iterations. “Meta-Learning” is trained without initialization of pre-trained model. “Pre-train (All)” and “Pre-train (Bicubic)” are fine-tuned from pre-trained models for all kernels (blind model) and bicubic downsampling model, respectively. All methods except ours are optimized using ADAM [16] while our method is optimized with gradient descent.

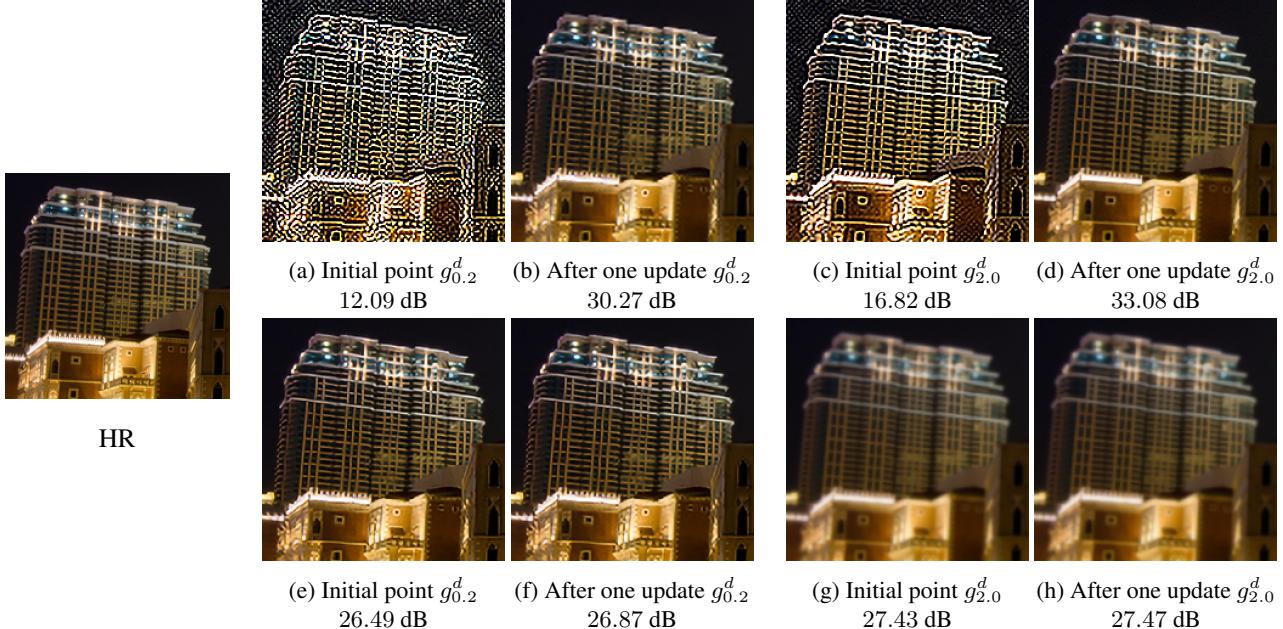


Figure 4: Visualization of the initial point and after one iteration of each method. Upper row images are from MZSR, and lower ones are from the pre-trained network on “bicubic” degradation.

## 6. Discussion

### 6.1. Number of Gradient Updates

For ablation investigation, we train several models with different configurations. We assess the average PSNR results on Set5, which are shown in Figure 3. Interestingly, the initial point of our method shows the worst performance, but in one iteration, our method quickly adapts to the image condition and shows the best performance among the compared methods. Other methods sometimes show a slow increase in performance. In other words, they are not as

flexible as ours in adapting to new image conditions.

We visualized the result at the initial point and after one gradient update in Figure 4. As shown, the result of the initial point of MZSR is weird, but within one iteration, it is highly improved. On the other hand, the result of a pre-trained network is more natural than MZSR, but its improvement after one gradient update is minor. Furthermore, it is shown that the performance of our method increases as the gradient descent update progresses, despite the fact that it is trained for maximum performance after five gradient steps. This result suggests that with more gradient

PSNR (dB)	$g_{0.2}^d$	$g_{2.0}^d$	$g_{ani}^d$
Multi-scale (10)	33.33(-0.41)	35.67(-0.97)	33.95(-0.83)

Table 3: Average PSNR results of multi-scale model on Set5 with  $\times 2$ . The number in parenthesis is PSNR loss compared to the single-scale model.

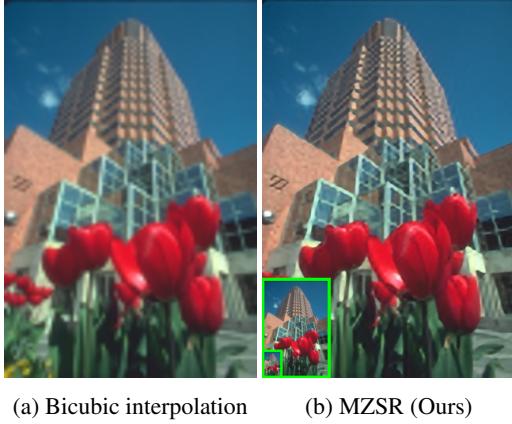


Figure 5: MZSR results on scaling factor  $\times 4$  with blur kernel  $g_{2.0}^d$ . Despite the size of LR son image is  $30 \times 20$ , MZSR learns internal information. (Green boxes at the lower left corner of MZSR image are  $\mathbf{I}_{son}$  and  $\mathbf{I}_{LR}$ )

update iterations, we might expect more of the performance improvements.

## 6.2. Multi-scale Models

We additionally trained a multi-scale model with the scaling factors  $s \in [2.0, 4.0]$ . The results on  $\times 2$  show worse results comparable to single-scale model as shown in Table 3. With multiple scaling factors, the task distribution  $p(\mathcal{T})$  becomes more complex, in which the meta-learner struggles to capture such regions that are suitable for fast adaptation.

Moreover, when meta-testing larger scaling factors, the size of  $\mathbf{I}_{son}$  becomes too small to provide enough information to the CNN. Hence, the CNN rarely utilizes information from a very small LR son image. Importantly, as our CNN learns internal information of CNN, such images with multi-scale recurrent patterns show plausible results even with large scaling factors, as shown in Figure 5.

## 6.3. Complexity

We evaluate the overall model and time complexities for several comparisons, and the results are shown in Table 4. We measure time on the environment of NVIDIA Titan XP GPU. Two fully-supervised feedforward networks for

Methods	Parameters	Time (sec)
CARN [2]	1,592 K	0.47
RCAN [45]	15,445 K	1.72
ZSSR [34]	225 K	142.72
MZSR (1)	225 K	0.13
MZSR (10)	225 K	0.36

Table 4: Comparisons of the number of parameters and time complexity for super-resolution of  $256 \times 256$  LR image with scaling factor  $\times 2$ .

“bicubic” degradation, CARN and RCAN, require a large number of parameters. Even though CARN is proposed as a lightweight network which requires one-tenth of parameters compared to RCAN, it still requires much more parameters compared to unsupervised networks. However, the time consumptions for both model are quite comparable, because only feedforward computation is involved.

On the other hand, ZSSR, which is totally unsupervised, requires much less number of parameters due to the image-specific CNN. However, it requires thousands of forward and backward pass to get a super-resolved image, *i.e.*, a large amount of time exceeding a practical extent. Our method MZSR with a single gradient update requires the shortest time among comparisons. Also, even with 10 iterations of the backward pass, our method still shows comparable time consumption against CARN.

## 7. Conclusion

In this paper, we have presented a fast, flexible, and lightweight self-supervised super-resolution method by exploiting both external and internal samples. Specifically, we adopt an optimization-based meta-learning method jointly with transfer learning to seek an initial point that is sensitive to different conditions of blur kernels. Therefore, our method can quickly adapt to specific image conditions within a few gradient updates. From our extensive experiments, we show that our MZSR outperforms other methods, including ZSSR, which requires thousands of gradient descent iterations. Furthermore, we demonstrate the effectiveness of our method with complexity evaluation. Yet, there are lots of parts that can be improved from our work such as network architecture, learning strategies, and multi-scale model, and we leave these as future works. Our code is publicly available at <https://www.github.com/JWSoh/MZSR>.

**Acknowledgements** This research was supported in part by Projects for Research and Development of Police sci-

ence and Technology under Center for Research and Development of Police science and Technology and Korean National Police Agency (PA-C000001), and in part by Samsung Electronics Co., Ltd.

## References

- [1] Eirikur Agustsson and Radu Timofte. Ntire 2017 challenge on single image super-resolution: Dataset and study. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 126–135, 2017. [3](#), [4](#), [5](#)
- [2] Namhyuk Ahn, Byungkon Kang, and Kyung-Ah Sohn. Fast, accurate, and lightweight super-resolution with cascading residual network. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 252–268, 2018. [1](#), [2](#), [5](#), [6](#), [8](#), [14](#)
- [3] Antreas Antoniou, Harrison Edwards, and Amos Storkey. How to train your maml. In *ICLR*, 2019. [3](#), [5](#)
- [4] Marco Bevilacqua, Aline Roumy, Christine Guillemot, and Marie Line Alberi-Morel. Low-complexity single-image super-resolution based on nonnegative neighbor embedding. In *BMVC*, 2012. [5](#)
- [5] Antoni Buades, Bartomeu Coll, and Jean-Michel Morel. Non-local means denoising. *Image Processing On Line*, 1:208–212, 2011. [2](#)
- [6] Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian. Image denoising by sparse 3-d transform-domain collaborative filtering. *IEEE Transactions on image processing*, 16(8):2080–2095, 2007. [2](#)
- [7] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. [4](#)
- [8] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1126–1135. JMLR.org, 2017. [2](#), [3](#), [4](#)
- [9] Chelsea Finn and Sergey Levine. Meta-learning and universality: Deep representations and gradient descent can approximate any learning algorithm. In *ICLR*, 2018. [2](#), [3](#), [4](#)
- [10] Erin Grant, Chelsea Finn, Sergey Levine, Trevor Darrell, and Thomas Griffiths. Recasting gradient-based meta-learning as hierarchical bayes. In *ICLR*, 2018. [2](#), [3](#)
- [11] Jinjin Gu, Hannan Lu, Wangmeng Zuo, and Chao Dong. Blind super-resolution with iterative kernel correction. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. [2](#), [5](#), [6](#), [13](#), [14](#)
- [12] Muhammad Haris, Gregory Shakhnarovich, and Norimichi Ukita. Deep back-projection networks for super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1664–1673, 2018. [1](#), [2](#)
- [13] Xiangyu He, Zitao Mo, Peisong Wang, Yang Liu, Mingyuan Yang, and Jian Cheng. Ode-inspired network design for single image super-resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1732–1741, 2019. [1](#), [2](#)
- [14] Jia-Bin Huang, Abhishek Singh, and Narendra Ahuja. Single image super-resolution from transformed self-exemplars. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5197–5206, 2015. [1](#), [2](#), [5](#)
- [15] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. Accurate image super-resolution using very deep convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1646–1654, 2016. [1](#), [2](#)
- [16] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. [5](#), [7](#)
- [17] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4681–4690, 2017. [1](#), [2](#)
- [18] Yoonho Lee and Seungjin Choi. Gradient-based meta-learning with learned layerwise metric and subspace. In *ICLR*, 2018. [2](#), [3](#)
- [19] Stamatios Lefkimiatis. Non-local color image denoising with convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3587–3596, 2017. [2](#)
- [20] Zhen Li, Jinglei Yang, Zheng Liu, Xiaomin Yang, Gwanggil Jeon, and Wei Wu. Feedback network for image super-resolution. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. [1](#), [2](#)
- [21] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Enhanced deep residual

- networks for single image super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 136–144, 2017. 1, 2, 4
- [22] Ding Liu, Bihan Wen, Yuchen Fan, Chen Change Loy, and Thomas S Huang. Non-local recurrent network for image restoration. In *Advances in Neural Information Processing Systems*, pages 1673–1682, 2018. 2
- [23] David Martin, Charless Fowlkes, Doron Tal, and Jitendra Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, volume 2, pages 416–423. IEEE, 2001. 5
- [24] Tomer Michaeli and Michal Irani. Nonparametric blind super-resolution. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 945–952, 2013. 2, 4
- [25] Nikhil Mishra, Mostafa Rohaninejad, Xi Chen, and Pieter Abbeel. Snail: a simple neural attentive meta-learner. In *ICLR*, 2018. 2, 3
- [26] Inbar Mosseri, Maria Zontak, and Michal Irani. Combining the power of internal and external denoising. In *IEEE international conference on computational photography (ICCP)*, pages 1–9. IEEE, 2013. 2
- [27] Alex Nichol, Joshua Achiam, and John Schulman. On first-order meta-learning algorithms. *arXiv preprint arXiv:1803.02999*, 2018. 3
- [28] Boris Oreshkin, Pau Rodríguez López, and Alexandre Lacoste. Tadam: Task dependent adaptive metric for improved few-shot learning. In *Advances in Neural Information Processing Systems*, pages 721–731, 2018. 2, 3
- [29] Jinshan Pan, Deqing Sun, Hanspeter Pfister, and Ming-Hsuan Yang. Blind image deblurring using dark channel prior. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1628–1636, 2016. 4
- [30] Andrei A Rusu, Dushyant Rao, Jakub Sygnowski, Oriol Vinyals, Razvan Pascanu, Simon Osindero, and Raia Hadsell. Meta-learning with latent embedding optimization. In *ICLR*, 2019. 3
- [31] Adam Santoro, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, and Timothy Lillicrap. Meta-learning with memory-augmented neural networks. In *International conference on machine learning*, pages 1842–1850, 2016. 3
- [32] Tamar Rott Shaham, Tali Dekel, and Tomer Michaeli. Singan: Learning a generative model from a single natural image. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019. 2
- [33] Assaf Shocher, Shai Bagon, Phillip Isola, and Michal Irani. Ingan: Capturing and retargeting the “dna” of a natural image. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019. 2
- [34] Assaf Shocher, Nadav Cohen, and Michal Irani. zero-shot super-resolution using deep internal learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3118–3126, 2018. 1, 2, 3, 4, 5, 6, 8, 13, 14
- [35] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*, pages 4077–4087, 2017. 2, 3
- [36] Jae Woong Soh, Gu Yong Park, Junho Jo, and Nam Ik Cho. Natural and realistic single image super-resolution with explicit natural manifold discrimination. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8122–8131, 2019. 1, 2
- [37] Qianru Sun, Yaoyao Liu, Tat-Seng Chua, and Bernt Schiele. Meta-transfer learning for few-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 403–412, 2019. 2, 3
- [38] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1199–1208, 2018. 2, 3
- [39] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. In *Advances in neural information processing systems*, pages 3630–3638, 2016. 2, 3
- [40] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004. 5
- [41] Zhangyang Wang, Yingzhen Yang, Zhaowen Wang, Shiyu Chang, Jianchao Yang, and Thomas S Huang. Learning super-resolution jointly from external and internal examples. *IEEE Transactions on Image Processing*, 24(11):4359–4371, 2015. 2
- [42] Jun Xu, Lei Zhang, and David Zhang. External prior guided internal prior learning for real-world noisy image denoising. *IEEE Transactions on Image Processing*, 27(6):2996–3010, 2018. 2
- [43] Huanjing Yue, Xiaoyan Sun, Jingyu Yang, and Feng Wu. Image denoising by exploring external and in-

- ternal correlations. *IEEE Transactions on Image Processing*, 24(6):1967–1982, 2015. 2
- [44] Kai Zhang, Wangmeng Zuo, and Lei Zhang. Learning a single convolutional super-resolution network for multiple degradations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3262–3271, 2018. 2, 5
- [45] Yulun Zhang, Kunpeng Li, Kai Li, Lichen Wang, Bineng Zhong, and Yun Fu. Image super-resolution using very deep residual channel attention networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 286–301, 2018. 1, 5, 6, 8, 13, 14
- [46] Yulun Zhang, Kunpeng Li, Kai Li, Bineng Zhong, and Yun Fu. Residual non-local attention networks for image restoration. *ICLR*, 2019. 2
- [47] Yulun Zhang, Yapeng Tian, Yu Kong, Bineng Zhong, and Yun Fu. Residual dense network for image super-resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2472–2481, 2018. 1, 2

## Appendix

### A. Evaluation on Scaling Factor $\times 4$

To evaluate the performance on large scaling factors, we demonstrate the results on scaling factor  $\times 4$  with isotropic Gaussian kernel with width 2.0 in Table 5. As shown, our methods show comparable results to others even with one gradient update, for large scaling factors too. Also, we found that multi-scale model shows worse results than a single-scale model as evidenced in the scaling factor  $\times 2$ .

### B. Effects of Kernels on Meta-test Time

To evaluate the effects of input kernels on meta-test time, we obtained several results by feeding various kernels. The results are shown in Figure 6. It is obvious that kernel mismatch degrades the output result severely. Especially, when the input kernel largely deviates from the true kernel, the result is not very pleasing as shown in Figure 6(a) and (b). However, if the input kernel has similar shape as the true kernel then the result looks quite plausible as shown in Figure 6(c). In conclusion, the kernel estimation or knowing the true kernel is crucial for the performance gain with our method.

### C. Visualization

To show the effectiveness of our MZSR, we visualize some results including scenarios with synthetic blur kernels and real-world images. Figure 7 and 8 are the results on synthetic blur kernels. 9 is the result on a real-world image.

$g_{2,0}^d$	Supervised			Unsupervised			
Dataset	Bicubic	RCAN [45]	IKC [11]	ZSSR [34]	Multi-scale (1)	MZSR (1)	MZSR (10)
Set5	24.74/0.7321	23.92/0.7283	24.01/0.7322	27.39/0.7685	29.85/0.8601	30.20/0.8655	30.50/0.8704
BSD100	24.01/0.5998	23.16/0.5918	23.12/0.5939	25.89/0.6776	26.68/0.7136	26.73/0.7138	26.89/0.7168
Urban100	21.16/0.5811	19.52/0.5400	19.81/0.5583	23.53/0.6822	24.13/0.7251	24.36/0.7333	24.65/0.7394

Table 5: Average PSNR/SSIM results on the scaling factor  $\times 4$  on benchmarks. The numbers in parenthesis in our methods stand for the number of gradient updates. The best and the second best are highlighted in red and blue, respectively.

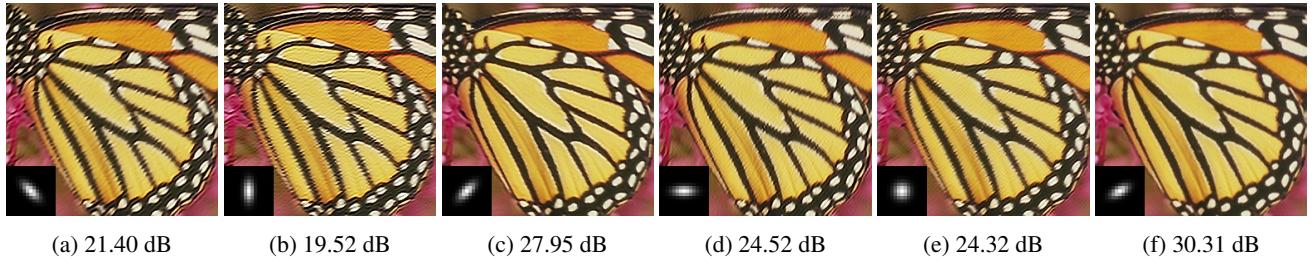


Figure 6: Comparisons when different kernels are applied on meta-test time. The last result is when the true kernel is applied.

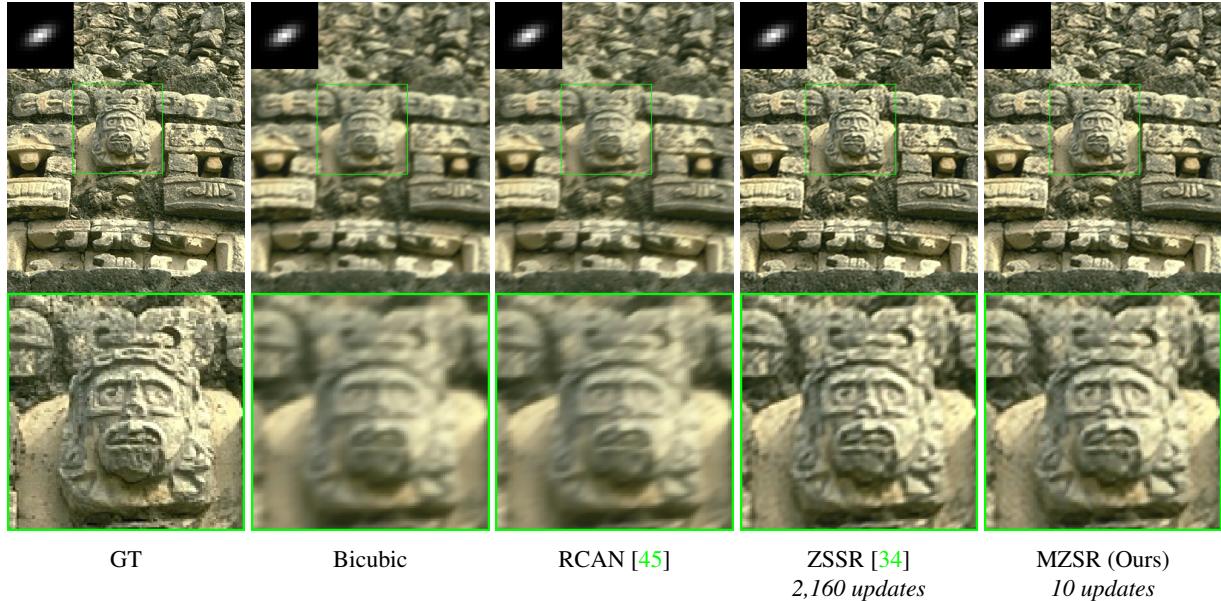


Figure 7: Visualized comparisons of super-resolution results ( $\times 2$ ) with anisotropic blur kernel  $g_{ani}^d$ .

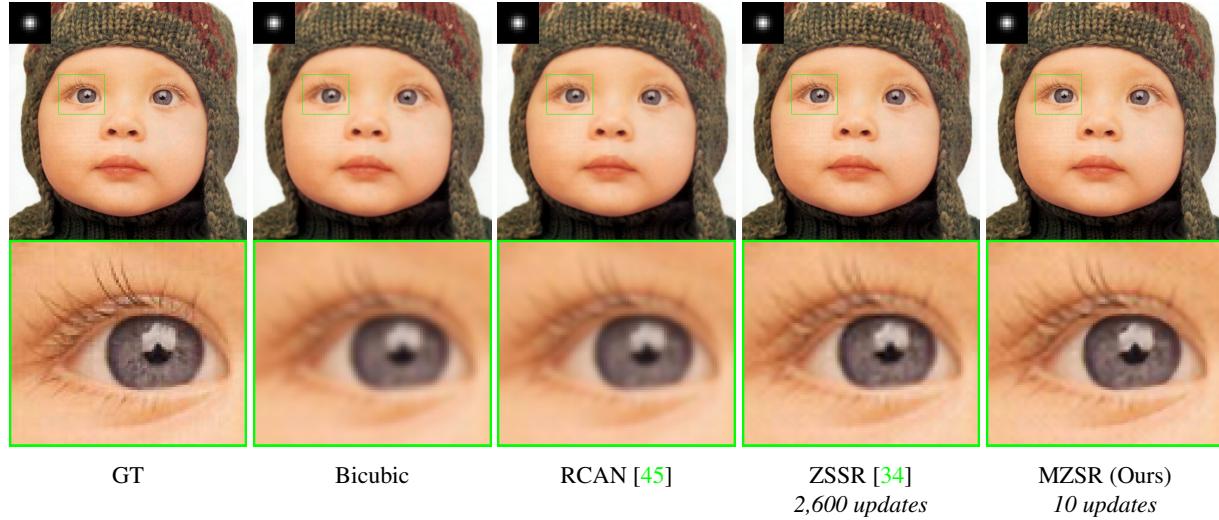


Figure 8: Visualized comparisons of super-resolution results ( $\times 2$ ) with isotropic blur kernel and bicubic subsampling  $g_{1.3}^b$ .



Figure 9: Visualized comparisons of super-resolution results ( $\times 4$ ) on real-world image.