# Learning Controllable Degradation for Real-World Super-Resolution via Constrained Flows

**Seobin Park** [*1] **Dongjin Kim** [*2] **Sungyong Baik** [3] **Tae Hyun Kim** [2]

## Abstract

Recent deep-learning-based super-resolution (SR) methods have been successful in recovering high-resolution (HR) images from their low-resolution (LR) counterparts, albeit on the synthetic and simple degradation setting: bicubic downscaling. On the other hand, super-resolution on real-world images demands the capability to handle complex downscaling mechanism which produces different artifacts (*e.g.,* noise, blur, color distortion) upon downscaling factors. To account for complex downscaling mechanism in real-world LR images, there have been a few efforts in constructing datasets consisting of LR images with real-world downsampling degradation. However, making such datasets entails a tremendous amount of time and effort, thereby resorting to very few number of downscaling factors (*e.g.,* $\times 2, \times 3, \times 4$). To remedy the issue, we propose to generate realistic SR datasets for unseen degradation levels by exploring the latent space of real LR images and thereby producing more diverse yet realistic LR images with complex real-world artifacts. Our quantitative and qualitative experiments demonstrate the accuracy of the generated LR images, and we show that the various conventional SR networks trained with our newly generated SR datasets can produce much better HR images.

## 1. Introduction

Recent studies on deep learning methodologies and large train datasets remarkably improved the performance of several image restoration tasks, such as image super-resolution (SR) (Dong et al., 2015; Zhang et al., 2018a) and image denoising (Zhang et al., 2017). The datasets used in these restoration tasks consist of pairs of low- and high-quality images, and they are used to learn a complex mapping between the low-quality image and the corresponding high-quality image. In general, these datasets are produced by assuming simple degradation models, such as bicubic downsampling (Agustsson & Timofte, 2017) and additive Gaussian noise (Zhang et al., 2017). However, deep SR networks trained with these synthetic datasets struggle with mapping the real-world low-resolution (LR) photos to its high-resolution (HR) counterparts due to discrepancies between the distributions of synthetic train images and real-world test images (*i.e.,* domain misalignment) (Cai et al., 2019). As a result, several studies have been conducted to generate new SR datasets that enable the SR networks to handle real-world LR images rather than synthetic LR images created by bicubic downsampling.

Recently, a few works have been proposed to simulate complex real-world degradation, attempting to produce more realistic SR datasets (Xiao et al., 2020; Ji et al., 2020; Wolf et al., 2021). In particular, RealSR dataset proposed in (Cai et al., 2019) includes the image pairs of real LR and HR photos of the same scene taken by changing the focal lengths of a DSLR camera. For a given scene, an image captured with a longer focal length (*e.g.,* 105mm) serves as an HR image, while an image captured with a shorter focal length (*e.g.,* 35mm) serves as an LR image. Then, an LR-HR image pair of the given scene (*e.g.,* $\times 3$ degradation level or scale factor) is created by using a pixel-wise alignment process with image registration algorithm (Cai et al., 2019). The dataset construction process is, however, time-consuming and burdensome, which makes it almost infeasible to create LR-HR pairs by continuously changing the degradation level. Thus, RealSR dataset is only composed of images with very few number of SR degradation levels (*e.g.,* $\times 2, \times 3, \times 4$), which severely restricts the practical usage in training the SR networks for real-world scenarios.

Therefore, we propose to develop a method that can synthesize realistic LR images that are not available in conventional real-world SR datasets by controlling the degradation

[*]Equal contribution [1]Department of Electrical and Computer Engineering, The University of Texas at Austin, Austin, Texas, USA [2]Department of Computer Science, Hanyang University, Seoul, Korea [3]Department of Data Science, Hanyang University, Seoul, Korea. Correspondence to: Tae Hyun Kim <taehyunkim@hanyang.ac.kr>.

*Proceedings of the $40^{th}$ International Conference on Machine Learning*, Honolulu, Hawaii, USA. PMLR 202, 2023. 

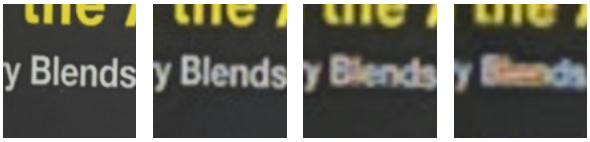


*Figure 1.* HR and corresponding LR images with different degradation levels ($\times 2$, $\times 3$ and $\times 4$) from RealSR (Cai et al., 2019) train dataset.

level continuously. While some recent approaches (Xiao et al., 2020; Wolf et al., 2021; Ji et al., 2020) are proposed to synthesize new datasets for a specific target degradation model, they cannot generate datasets beyond that target domain (*e.g.,* unseen/arbitrary degradation levels). However, generating LR images with unseen/arbitrary degradation levels is challenging as we observe that the distortion changes in a complex and non-linear manner as the degradation level changes (see the complex changes in color and blur according to the degradation level in Figure 1).

In this work, we start from an observation that LR images share some common aspects across different levels of degradation (e.g., blue-and-orange color distortions in images with degradation levels of $\times 2$, $\times 3$ and $\times 4$ in Figure 1). One naïve approach for generating LR images with continuous degradation levels would be to simply interpolate two images with different degradation levels in the pixel space. However, this approach causes non-realistic artifacts when two highly different distortions are merged, and the resulting images are not diverse (*i.e.,* discriminative approach). To generate the LR images maintaining degradation fidelity with continuous manner, we propose to traverse along the latent space of given LR images to create LR images for unseen degradation levels. Specifically, we employ conditional normalizing flows (NF) (Lugmayr et al., 2020), and owing to its ability of calculating exact likelihood, we can easily embed the degradation information to the latent space and manipulate the latent space to ensure that plausible and diverse LR images are generated according to degradation levels. To organize the constrained latent space, our framework utilizes two losses: a newly introduced LR-consistency loss and information bottleneck (IB) loss (Tishby et al., 2000). We present a new LR-consistency loss to enforce the consistency between the generated LR images and real images in terms of the low-frequency structures. IB loss is used to ensure that different latent variables correspond to different degradation levels, generating diverse LR images according to degradation level.

To our best knowledge, our approach is the first attempt to generate realistic LR images by changing the degradation level continuously given an HR image. We empirically validate the effectiveness of our method in generating LR images with arbitrary and continuous-valued degradation levels while capturing complex degradation in real-world. The experimental results demonstrate that our proposed dataset can be used to train many conventional SR networks to handle LR images with arbitrary degradation levels.

## 2. Related Works

• **Single image super-resolution (SISR)** The SISR task aims to enhance the resolution of low-resolution (LR) images by recovering high-frequency details. Recent deep learning-based methods have demonstrated remarkable performance (Dong et al., 2015; Kim et al., 2016; Zhang et al., 2018b;a; Liang et al., 2021), compared to traditional methods (Chang et al., 2004; Glasner et al., 2009). These methods require a large amount of training data, thus employing simple bicubic downsampling to synthesize a sufficiently large dataset. However, SR networks fail to work on images that are out of the distribution of training images (El Helou et al., 2020). Because the real-world degradations do not follow bicubic downsampling, there exist discrepancies between real-world LR and synthetic training LR images, leading to failures of SR networks in real-world scenarios (Cai et al., 2019).

• **Real-world SISR datasets** To bridge the gap, several works have been proposed to capture real LR-HR pairs of images using physical equipment like beam-splitter (Qu et al., 2016) and cameras (Köhler et al., 2019; Zhang et al., 2019; Cai et al., 2019), as well as traditional degradation models such as Gaussian noise, kernel, and bicubic interpolation (Zhang et al., 2021; Wang et al., 2021). In particular, RealSR (Cai et al., 2019) dataset was collected by taking images of the same scene with different focal lengths and by aligning them through optimization. However, constructing these datasets is highly laborious, posing challenges for building a large-scale dataset. As a result, new approaches have been recently developed to synthetically generate much larger datasets for the real-world SR problem. DML (Xiao et al., 2020) modeled real-world degradation, where the degradation kernel is assumed to be non-uniform and spatially varying. Impressionism (Ji et al., 2020) and DeFlow (Wolf et al., 2021) proposed to learn more complex degradation models using generative methods. BSRGAN (Zhang et al., 2021) and Real-ESRGAN (Wang et al., 2021) learned SR networks that are robust against various real-world degradations by synthesizing the dataset with a random combination of multiple degradations. Comparing to the prior arts, our study aims to tackle the challenge of generating LR images with continuous degradation levels facilitating real-world degradation information.

• **Arbitrary scale super-resolution** To improve the generalizability and applicability, several SISR networks that can handle arbitrary scales have been introduced (Hu et al., 2019; Chen et al., 2021; Yang et al., 2021; Lee & Jin, 2022).

Meta-SR (Hu et al., 2019) proposed a meta-network that can predict weights of upscale filters for a given arbitrary scale factor. Interpreting images as continuous fields of pixels. LIIF (Chen et al., 2021) adopts implicit neural representation, which enables the representation of images with continuous, arbitrary scales. However, to train these networks to handle arbitrary degradation levels, they need LR images generated with simple bicubic downsampling with different scaling factors. Thus, these networks can be vulnerable to real-world degradation due to domain misalignment problem (Cai et al., 2019).

• **Normalizing flow (NF)** Owing to its capability in density estimation, NF has recently gained a particular amount of attention in computer vision tasks requiring data distribution modeling (Kingma & Dhariwal, 2018; Lugmayr et al., 2020; Wolf et al., 2021; Abdelhamed et al., 2019; Wang et al., 2022; Ardizzone et al., 2020). In SR tasks, SRflow (Lugmayr et al., 2020) model the conditional distribution of HR images given LR images to tackle an ill-posed problem that a single LR image can be restored into multiple HR images. Deflow (Wolf et al., 2021) further models the degradation distributions on the unpaired setting of noisy and clean images based on SRflow model architecture. However, these methods can only handle a specifically targeted SR degradation level (*e.g.,* $\times 2$) and thus cannot handle real-world LR images with unseen/arbitrary degradation levels.

• **Latent space interpolation** Several works on image manipulation and generations (Karras et al., 2020; Patashnik et al., 2021; Liu et al., 2021; Berthelot* et al., 2019; Chen et al., 2019; Shen et al., 2020; Zhu et al., 2020) have attempted to generate new images with intended semantic information by manipulating the latent variables. For instance, new face images can be generated by controlling high-level attributes, such as ages, hairstyles, and emotion (Chen et al., 2019; Shen et al., 2020; Zhu et al., 2020). While these methods focus on manipulating mostly high-level and domain-specific attributes, our proposed method aims to accurately control low-level image attributes like blur and noise using a constrained latent variable interpolation.

## 3. Proposed Method: InterFlow

In Figure 1, we can see different artifacts such as noise, blur, and color distortion as degradation level changes. Thus, we aim to develop a new generative model which allows us to control the degradation level arbitrary and generate LR images, including more realistic artifacts. In this section, we explain our LR image generation process for unseen degradation levels by leveraging given real LR images in the latent space that we dub $InterFlow$.

To model the degradation information, we use RealSR (Cai et al., 2019) dataset. The dataset approximates the DLSR imaging system based on thin lens equation with an additional assumption that the distance between lens and object is sufficiently far apart (*e.g.,* >2m) (Cai et al., 2019). Based on this approximation, the image size can be zoomed linearly to the focal length. For instance, the image taken with a longer focal length can represent enlarged objects with finer textures. Therefore, we define degradation level $s$ as a ratio of different focal lengths as follows:

$$s = \frac{f_2}{f_1},\ f_2 \geq f_1 \tag{1}$$

where $f_2$ is fixed to the longest possible focal length (*e.g.,* 105mm). Images with different degradation levels include different amounts of noise, blur, and artifacts. We denote the HR image as $I_{HR}$ and the corresponding LR image as $I_s$, which are identical visual scenes captured with different focal lengths $f_2$ and $f_1$, respectively. Any lens distortion, exposure, and pixel misalignments of $I_{HR}$ and $I_s$ are corrected by image pair registration method originally used in RealSR (Cai et al., 2019) during which $I_s$ is scaled up using bicubic upsampling to have the same resolution as $I_{HR}$. For convenience, we will denote the aligned LR image as $I_s$. Note that $s$ is indicated as scale factor in RealSR (Cai et al., 2019), but since the paired HR and LR images of RealSR are pixel-wise aligned, we will refer to $s$ as degradation level in the following sections to avoid terminology confusion.

Estimating the degradation function for an arbitrary $s$ is very difficult, since the real degradation process is highly complex and non-linear (Figure 1 and Figure 4) and real $I_s$ is not available for all $s$. We circumvent this problem by leveraging existing real LR images with limited degradation levels (*e.g.,* $\times 2$ and $\times 4$ LR images in RealSR) in the latent space. We provide more details in the following sections.

### 3.1. Learning controllable degradation level

To deal with LR images with different degradation levels, we first define the latent variable $z_s$ of an LR image $I_s$, and we assume that the latent $z_s$ follows a normal distribution whose mean and variance are $\mu_s$ and $\sigma_s^2$, respectively:

$$z_s \sim \mathcal{N}(\mu_s, \sigma_s^2), \tag{2}$$

where $\mu_s$ and $\sigma_s$ are trainable parameters in our work, and they allow us to generate more distinct LR images for each degradation level $s$.

In this work, we transform the LR image $I_s$ to the latent $z_s$ using conditional NF (Rezende & Mohamed, 2015; Lugmayr et al., 2020), where the relation between $I_s$ and $z_s$ given $I_{HR}$ is formulated with an invertible neural network $f$ with parameter $\theta$:

$$\begin{cases} z_s = f_\theta(I_s; I_{HR}) \\ I_s = f_\theta^{-1}(z_s; I_{HR}), \end{cases} \tag{3}$$

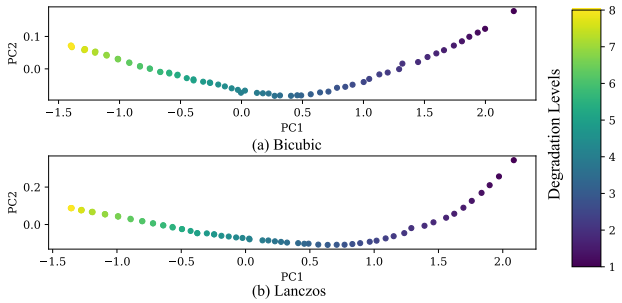


*Figure 2.* Visualization in a low dimensional space. PCA is used for dimension reduction of 80 trained means ($\{\mu_s\}_{s\in S}$) and they are projected onto 2D space. (a) NF is trained with bicubic kernel. (b) NF is trained with Lanczos kernel.

and the conditional probability density function for $I_s$ with change of variables formula is given by,

$$p(I_s|I_{HR}) = |\det Df_\theta(I_s; I_{HR})| \cdot g(z_s; \mu_s, \sigma_s^2), \quad (4)$$

where $D$ computes the Jacobian of the flow network $f_\theta$, and $g(.; \mu_s, \sigma_s^2)$ denotes the probability density function of the normal distribution with mean $\mu_s$ and variance $\sigma_s^2$. By using this conditional density function, we can train our neural network $f_\theta$ by minimizing the following negative log-likelihood (NLL) and learn controllable degradation as:

$$\begin{aligned}\mathcal{L}_{\text{flow}}(\theta) &= -\sum_{s\in S} \log p(I_s|I_{HR}) \\ &= -\sum_{s\in S} \log|\det Df_\theta(I_s; I_{HR})| \\ &\quad + \log g(f_\theta(I_s; I_{HR}); \mu_s, \sigma_s^2),\end{aligned} \quad (5)$$

where $S$ is the set of discrete degradation levels of LR images available in SR dataset (*e.g.,* $S = \{2, 3, 4\}$ for RealSR). Once $f_\theta$ is trained, given any HR image $I_{HR}$, we can synthesize a new LR image for a specific degradation level (*i.e.,* $s \in S$) by feeding a random sample from the normal distribution $\mathcal{N}(\mu_s, \sigma_s^2)$ to the inverse function $f_\theta^{-1}$ in (3).

### 3.2. Exploring latent space

To generate LR images for unseen degradation levels, we first explore the latent space of NF, based on our key observation that latents of NF change smoothly in a low dimensional space when the degradation level changes gradually. We show our finding in Figure 2. To be specific, we collect LR-HR image pairs for $S = \{1.1, 1.2, \ldots, 8.0\}$ by down- and up-scaling HR images, in turn, using bicubic kernel, then train NF with the collected image pairs by minimizing NLL in (5). Then, using PCA, we project 80 mean values (*i.e.,* $\{\mu_s\}_{s\in S}$) in a high-dimensional space onto 2D and observe that the projected mean values are lined up in the low dimensional space in as shown Figure 2(a). In particular, these projected mean values are changing smoothly as the degradation level increases. We also see similar results even when LR-HR image pairs are generated with Lanczos kernel Figure 2(b).

From this key observation, we assume that a latent for unseen degradation (*e.g.,* $z_{1.76}$) is predictable with nearby available latents (*e.g.,* $z_{1.7}$ and $z_{1.8}$), allowing us to explore the latent space of NF to generate new LR images for arbitrary degradation level.

### 3.3. Exploiting latent space

In Figure 1, we see that real-world distortions, such as blur and color shift, change highly non-linearly but smoothly across degradation levels. Therefore, we assume that real-world LR images also change smoothly in a latent space as in Figure 2, thereby exploiting the latent space of NF to generate new LR images for unseen degradation information.

Specifically, we first predict a latent of an LR image for a specific degradation level $s' \notin S$, given $I_{HR}$, by leveraging existing latents of real LR images in the train-dataset as:

$$z_{s'} = a \cdot z_{\lfloor s' \rfloor} + (1 - a) \cdot z_{\lceil s' \rceil}, \quad (6)$$

where $\lfloor s' \rfloor$ denotes the largest degradation level in the set $S$ less than $s'$, $\lceil s' \rceil$ means the smallest one in $S$ greater than $s'$ ($\lfloor s' \rfloor < s' < \lceil s' \rceil$), and $a$ is the blending weight ($0 < a < 1$). Now, we can synthesize a new latent $z_{s'}$ by interpolating two nearby latents from the dataset with $a$, where we can expect more accurate latent prediction result when these two nearby latents are close (*i.e.,* $z_{\lfloor s' \rfloor} \approx z_{\lceil s' \rceil}$). For example, when pixel-wise aligned $I_{HR}$, $I_2$ and $I_3$ images in RealSR are given, we can produce any latent $z_{s'}$ where $2 < s' < 3$ by blending the latents of $I_2$ and $I_3$. Moreover, we can naturally extend our formulation in (6) to exploit more existing latents larger than two if available (*e.g.,* polynomial interpolation rather than bilinear one).

Then, we can synthesize a new LR image $I_{s'}$ for unseen degradation level $s'$ by taking the inverse of the our flow network with given $z_{s'}$ and $I_{HR}$ as follows:

$$I_{s'} = f_\theta^{-1}(z_{s'}; I_{HR}). \quad (7)$$

### 3.4. Constrained Flow

In this work, we aim to generate more realistic LR images corresponding to the given HR image $I_{HR}$. However, the synthetic LR images from $f_\theta^{-1}$ optimized solely using NLL in (5) have some limitations. For instance, pixel-wise alignment with $I_{HR}$ is not guaranteed, and they cannot capture subtle artifacts included in real LR images, such as tiny noise, as shown in Figure 5 (rightmost). To alleviate this problem, we give more constraints on the flow network and further improve the quality of the generated LR images.

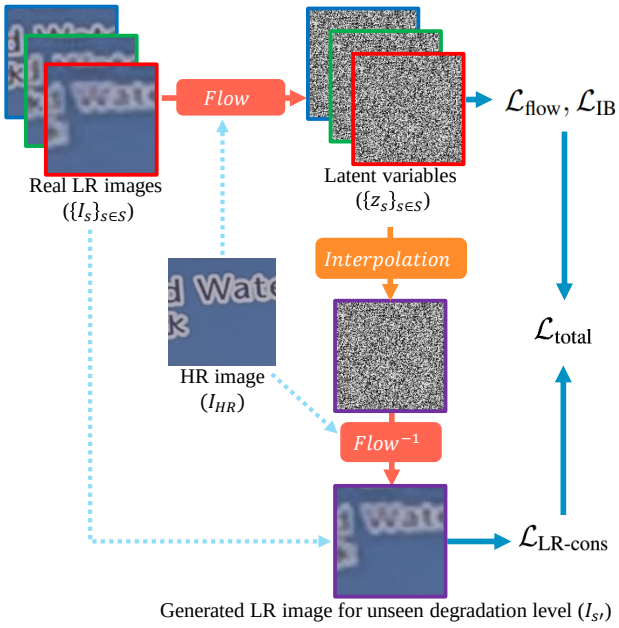


*Figure 3.* Overview of InterFlow.

• **Enforcing LR-consistency** Although distributions of the generated LR images by our InterFlow trained using the NLL loss in (5) can be similar to those of real LR images, these synthetic LR images are not accurately aligned with the corresponding HR images and even include slight deviations in color as well as alignment.

To solve this problem, we propose a new loss function $\mathcal{L}_{\text{LR-cons}}(\theta)$ to enforce consistency between the generated LR image $I_{s'}$ and corresponding real LR images in terms of the low-frequency structure:

$$\mathcal{L}_{\text{LR-cons}}(\theta) = ||I_{s'}\downarrow - (a \cdot I_{\lfloor s' \rfloor} + (1-a) \cdot I_{\lceil s' \rceil})\downarrow||, \quad (8)$$

where $I_{\lfloor s' \rfloor}$ and $I_{\lceil s' \rceil}$ are two pixel-wise aligned real LR images in the train dataset, where their HR counterparts are the same, but differently degraded, and $\downarrow$ indicates the bicubic down-sampling process. Note that the same blending weight $a$ is used for the latent space fusion in (6) and also used for image space blending in (8).

Thus, we can enforce the consistency of the low-frequency information such as color, shape, and structure between the generated LR image and corresponding real LR images, rather than high-frequency detail, as we measure the similarity in down-scaled image space.

• **Enforcing latent discriminability** Conventional generative models learn to capture distribution of train datasets, whereas we utilize the learned distributions to predict a latent beyond these learned distributions. However, we see that the generated LR images $I_{s'}$ from InterFlow trained with only NLL do not show characteristic of real LR images when $s' \notin S$, and it is a natural phenomenon when the two learned normal distributions $\mathcal{N}(\mu_{\lfloor s' \rfloor}, \sigma^2_{\lfloor s' \rfloor})$ and $\mathcal{N}(\mu_{\lceil s' \rceil}, \sigma^2_{\lceil s' \rceil})$ are close enough and overlapped. This restricts the capability of NF to generate new and distinctive LR images for unseen degradation levels, so we employ the information bottleneck (IB) (Tishby et al., 2000) to give more discriminability among the learned distributions, and it yields IB loss function as,

$$\mathcal{L}_{\text{IB}} = -\frac{1}{|S|}\sum_{s\in S} \log \frac{\mathcal{N}(f_\theta(I_s; I_{HR}); \mu_s, \sigma_s^2)}{\sum_{t\in S}\mathcal{N}(f_\theta(I_t; I_{HR}); \mu_t, \sigma_t^2)}, \quad (9)$$

where $|S|$ denotes the cardinality of $S$.

Note that, unlike other approaches which employ IB for generative classification tasks (Ardizzone et al., 2020), we use IB to improve the generation capability of the generative model for unseen distributions.

• **Overall flow** Figure 3 shows the overall flow of the proposed LR image generation approach. Our proposed network is trained to minimize the following loss function:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{flow}} + \lambda_{\text{LR-cons}} \cdot \mathcal{L}_{\text{LR-cons}} + \lambda_{\text{IB}} \cdot \mathcal{L}_{\text{IB}}, \quad (10)$$

where $\lambda_{\text{LR-cons}}$ and $\lambda_{\text{IB}}$ are user-parameters to control the balance among loss terms. In our experiments, we demonstrate that we can produce plausible LR images even for unseen degradation level with the aid of proposed loss models.

## 4. Experimental Results

In this section, we elaborate on implementation details and measure the quantitative and qualitative results of the generated LR images, and show the elevation of SR performance with the aid of our synthetic LR images. We will release our source code and dataset upon acceptance.

### 4.1. Implementation details

• **InterFlow architecture** For InterFlow, we use a slightly modified version of the conditional NF introduced in SR-Flow (Lugmayr et al., 2020). However, we can use any conditional NF architecture and do not restrict our method to a specific network architecture. See our supplementary material for the detailed network configuration.

• **Experimental settings** To train and evaluate the proposed method, we use the real-world SR dataset (RealSR ver.2). Specifically, RealSR is composed of images captured by Canon and Nikon cameras. To show the generalization ability of our method, we use only Canon train-dataset to train InterFlow and then synthesize LR images given HR images in the Nikon dataset through our InterFlow.

Our InterFlow is trained by minimizing the loss in (10) using the Adam optimizer (Kingma & Ba, 2015) with 160×160 train-patches (batch size = 8) for 100k iterations. The learning rate is initially set to $10^{-4}$ and reduced by half at 50k,

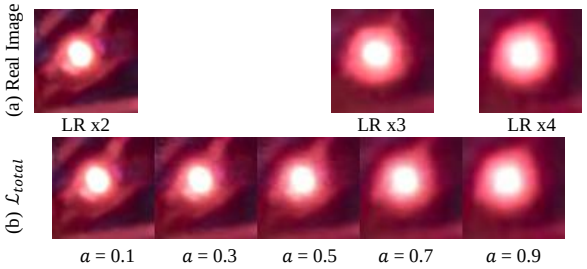


*Figure 4.* Visual comparison of LR images. (a) Real LR images in RealSR dataset. (b) Our synthetic images obtained by changing $a$.

*Figure 5.* Visual comparison of high frequency components. Our InterFlow successfully generates tiny noise in real LR images than the naïve image space interpolation approach. **From left to right**: synthetic LR image by image space interpolation, our synthetic LR image, and RealSR $\times 3$ LR image.

75k, and 90k iterations. Moreover, downscaling factor in the proposed LR-consistency loss in (8) is set to 4, and we use $\lambda_{\text{LR-cons}} = 10$ and $\lambda_{\text{IB}} = 1$ which are determined empirically.

Using our generated LR images, we train conventional SR networks; VDSR (Kim et al., 2016), RCAN (Zhang et al., 2018a), KPN (Cai et al., 2019), HAN (Niu et al., 2020), NLSN (Mei et al., 2021), and SwinIR (Liang et al., 2021). In particular, we add a squeeze operation (Kingma & Dhariwal, 2018) at the beginning of RCAN, HAN, NLSN, and SwinIR architectures to enable them to take LR input whose resolution is equal to that of output, thereby being able to handle our train-dataset. We use $k = 5$ in KPN. These six SR networks are trained by minimizing the conventional L1 loss using the Adam optimizer with $128 \times 128$ patches (batch size = 16) for 70k, 300k, 200k, 300k, 300k, and 300k iterations, respectively. The learning rate is initially set to $10^{-4}$ and halved at 50%, 75%, and 87.5% of the total training iterations.

Furthermore, for extensive experiments, we collect more test data for degradation levels $\times 2.5$ and $\times 3.5$ using the same dataset acquisition process in RealSR (Cai et al., 2019). To be specific, we use Canon 5D Mark 3 camera equipped with a lens with focal length 24 ~105 mm to take pairs of LR and HR images. Our RealSR $\times 2.5$ dataset is acquired by taking 40 pairs of images with focal lengths of 42 mm and 105 mm, and RealSR $\times 3.5$ dataset is collected by taking 40 pairs of images with the focal length of 31 mm and 105 mm. More details are available in the supplementary material.

### 4.2. LR image generation results

To evaluate the quality of the generated LR images, we train our InterFlow with the LR images in RealSR Canon trainset for $\times 2$ and $\times 4$ degradation levels (*i.e.,* $\{2, 4\} \in S$). Using the fully trained InterFlow, we generate LR images given RealSR Nikon test HR images via formulations (6)-(7). Following (Jang et al., 2021), we first measure the accuracy of added noise in our generated LR images using KL-divergence. Then, we show qualitative results and SR results to demonstrate the superiority of our generated LR images.

In Figure 6, we evaluate the KL-divergence of our synthetic LR images on $\times 3$ with Nikon $\times 3$ LR test images, which are the ground truth LR images for $\times 3$ RealSR. In this figure, we provide KL-divergence by changing the blending weight $a$ in the latent space (red line) and in the image space (green line). We observe that each method reaches a certain local minimum, which corresponds to the optimal blending weight $a$ that makes the generated dataset to resemble the ground truth $\times 3$ RealSR LR images (*i.e.,* Nikon $\times 3$ LR test images). However, we can easily see that the minimum value of KL-divergence in the LR images generated by naïve image space interpolation does not reach the minimum KL-divergence value of latent space interpolation method (ours). Thus indicating the superiority of our method over the naïve image space interpolation method.

Moreover, the visual comparison in Figure 5 demonstrates that our latent space interpolation method produces more plausible noise than the result by image (pixel) space interpolation. Specifically, we observe that the naïve image space interpolation method (leftmost image in Figure 5) does not simulate the complex noise present above the yellow letter 'M' of the ground truth RealSR $\times 3$ LR image (rightmost image in Figure 5). In Figure 4, we show a qualitative result of our method that accurately reproduces the complex degradation of RealSR. We can see that by changing the blending weight $a$, our InterFlow captures the change from LR$\times 2$ to LR$\times 4$ to accurately predict LR$\times 3$ image.

### 4.3. Real-world SR performance

We show that our generated dataset is indeed an effective dataset for real-world SR. To assess our method, we train conventional SR networks (VDSR, RCAN, KPN, HAN, NLSN, and SwinIR) with our generated dataset and evaluate them with real-world images. Our InterFlow is trained on RealSR Canon train set for $\times 2$ and $\times 4$ degradation levels (*i.e.,* $\{2, 4\} \in S$). The trained InterFlow is used to generate LR images, given RealSR Nikon HR images. The generated LR images are used to train these SR networks. These SR networks are evaluated on both Canon and Nikon test sets of LR $\times 3$ and corresponding HR images.

In Table 1, we compare SR networks trained with RealSR

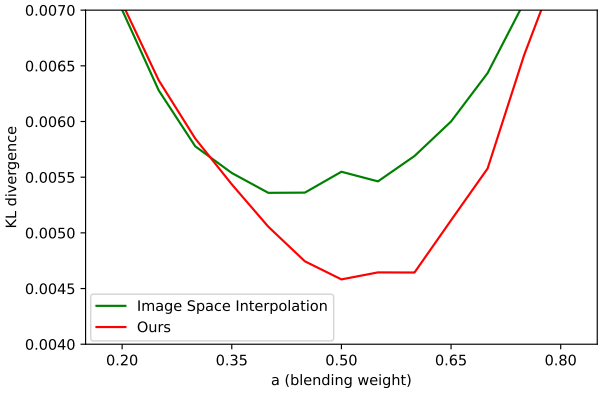


*Figure 6.* Comparison of high-frequency details. By changing the blending weight $a$, we measure the distance from high-frequency distribution of $\times 3$ Nikon LR images and distributions by synthetic datasets in terms of KL-divergence.

*Table 1.* SR results on the RealSR test set with scale $\times 3$ ($s = 3$). Results by the ground truth RealSR ($\times 3$) train set is the upper bound, since the ground truth train set is considered unavailable in this experimental setting (*i.e.,* $S = \{2, 4\}$ in (5)).

| Model | Train set | Metric | | |
|---|---|---|---|---|
| | | PSNR | SSIM | LPIPS |
| VDSR | RealSR ($\times 3$) | 30.33 | 0.8503 | 0.3351 |
| | RealSR ($\times 2$, $\times 4$) | 29.92 | 0.8404 | 0.3400 |
| | Ours ($\times 2$ ~$\times 4$) | **30.23** | **0.8497** | **0.3341** |
| RCAN | RealSR ($\times 3$) | 30.68 | 0.8641 | 0.3243 |
| | RealSR ($\times 2$, $\times 4$) | 30.30 | 0.8596 | 0.3281 |
| | Ours ($\times 2$ ~$\times 4$) | **30.57** | **0.8631** | **0.3155** |
| KPN | RealSR ($\times 3$) | 30.45 | 0.8576 | 0.3351 |
| | RealSR ($\times 2$, $\times 4$) | 30.07 | 0.8520 | 0.3361 |
| | Ours ($\times 2$ ~$\times 4$) | **30.32** | **0.8572** | **0.3263** |
| HAN | RealSR ($\times 3$) | 30.76 | 0.8659 | 0.3216 |
| | RealSR ($\times 2$, $\times 4$) | 30.43 | 0.8616 | 0.3261 |
| | Ours ($\times 2$ ~$\times 4$) | **30.68** | **0.8644** | **0.3167** |
| NLSN | RealSR ($\times 3$) | 30.72 | 0.8643 | 0.3187 |
| | RealSR ($\times 2$, $\times 4$) | 30.47 | 0.8614 | 0.3205 |
| | Ours ($\times 2$ ~$\times 4$) | **30.63** | **0.8636** | **0.3110** |
| SwinIR | RealSR ($\times 3$) | 30.69 | 0.8647 | 0.3217 |
| | RealSR ($\times 2$, $\times 4$) | 30.23 | 0.8597 | 0.3255 |
| | Ours ($\times 2$ ~$\times 4$) | **30.56** | **0.8634** | **0.3166** |
| KernelGAN | - | 28.55 | 0.8131 | 0.3636 |

$\times 3$ dataset (upper bound); RealSR $\times 2$, $\times 4$ datasets (baseline); and our $\times 2$ ~$\times 4$ synthetic datasets ($S = \{2, 4\}$ in (5)). RealSR $\times 3$ dataset will provide upper bound performance because the domain aligns between train and test (degradation level $\times 3$). Meanwhile, RealSR $\times 2$, $\times 4$ dataset serves as baseline for our datasets since they use the same available training sets. Specifically, we generate $\times 2$ ~$\times 4$ datasets using degradation level $s'$, where $2 < s' < 4$. Our method consistently outperforms the baseline, and the SR results are close to the upper bound.

Figure 7 further demonstrates that our synthetic dataset leads to visually more pleasing SR results, displaying higher robustness against undesirable artifacts compared to the baseline (see Canon_017_HR.png in Figure 7). The results indicate that baselines suffer from the discrepancy in degradation levels, while our method is resilient, owing to the accurate generation of LR images with diverse degradation levels. We also compare with KernelGAN (Bell-Kligler et al., 2019), which aims for real-world SR through estimating blur kernels. Models trained with RealSR and our method exhibit higher performance than KernelGAN, underlining the importance of acquiring rich accurate SR datasets. We also report the results on images with degradation levels $\times 2.5$, $\times 3.5$ (*i.e.,* RealSR $\times 2.5$, RealSR $\times 3.5$) in Table 3. Please see the supplementary material for visual results.

*Table 2.* SR performance results on different train set generated by Interflow applied different losses (LR-cons, IB) using HAN (tested on RealSR test set $\times 3$).

| Train set | LR-Cons | IB | RealSR $\times 3$ test set | | |
|---|---|---|---|---|---|
| | | | PSNR | SSIM | LPIPS |
| RealSR ($\times 2$, $\times 4$) | ✗ | ✗ | 30.43 | 0.8616 | 0.3261 |
| Ours ($\times 2$ ~$\times 4$) | ✗ | ✗ | 30.34 | 0.8567 | 0.3171 |
| Ours ($\times 2$ ~$\times 4$) | ✓ | ✗ | 30.16 | 0.8498 | 0.3233 |
| Ours ($\times 2$ ~$\times 4$) | ✗ | ✓ | 30.40 | 0.8610 | 0.3209 |
| Ours ($\times 2$ ~$\times 4$) | ✓ | ✓ | **30.68** | **0.8644** | **0.3167** |

*Table 3.* Quantitative results are evaluated on the RealSR $\times 2.5$ and RealSR $\times 3.5$ testsets.

| Model | Train set | PSNR | | SSIM | |
|---|---|---|---|---|---|
| | | $\times 2.5$ | $\times 3.5$ | $\times 2.5$ | $\times 3.5$ |
| RCAN | RealSR | 31.79 | 30.06 | 0.8864 | 0.8327 |
| | Ours | **32.11** | **30.20** | **0.8891** | **0.8337** |
| NLSN | RealSR | 31.85 | 30.01 | 0.8869 | 0.8295 |
| | Ours | **31.93** | **30.20** | **0.8879** | **0.8327** |

*Table 4.* SR results from HAN trained with different LR generation methods (tested on the RealSR testset $\times 3$).

| Model | Train set | LR Generation Method | RealSR $\times 3$ test set | |
|---|---|---|---|---|
| | | | PSNR | SSIM |
| HAN | RealSR $\times 2$,$\times 4$ | None | 30.43 | 0.8616 |
| | | Linear Interpolation | 30.48 | 0.8564 |
| | | InterFlow (Ours) | **30.68** | **0.8644** |

### 4.4. Ablation study

• **Impact of each loss term** We demonstrate the effectiveness of proposed loss terms (*i.e.,* , $\mathcal{L}_{\text{LR-cons}}$, $\mathcal{L}_{\text{IB}}$) by dissecting them, with results in Table 2. Disabling either loss terms results in significant performance degradation, often worse than the baseline trained with only RealSR ($\times 2$, $\times 4$) datasets. This highlights the importance of generating realistic and diverse degradations to tackle real-world SR.

• **Interpolating LR images in image space** To further demonstrate the effectiveness of our method, we compare our method with Linear Interpolation in Table 4, which is one of the naïve LR generation methods. Specifically, for Linear Interpolation, we generate LR images with scales from 2 to 4 by linearly interpolating $\times 2$ and $\times 4$ LR images. An SR network HAN trained with our dataset demonstrates

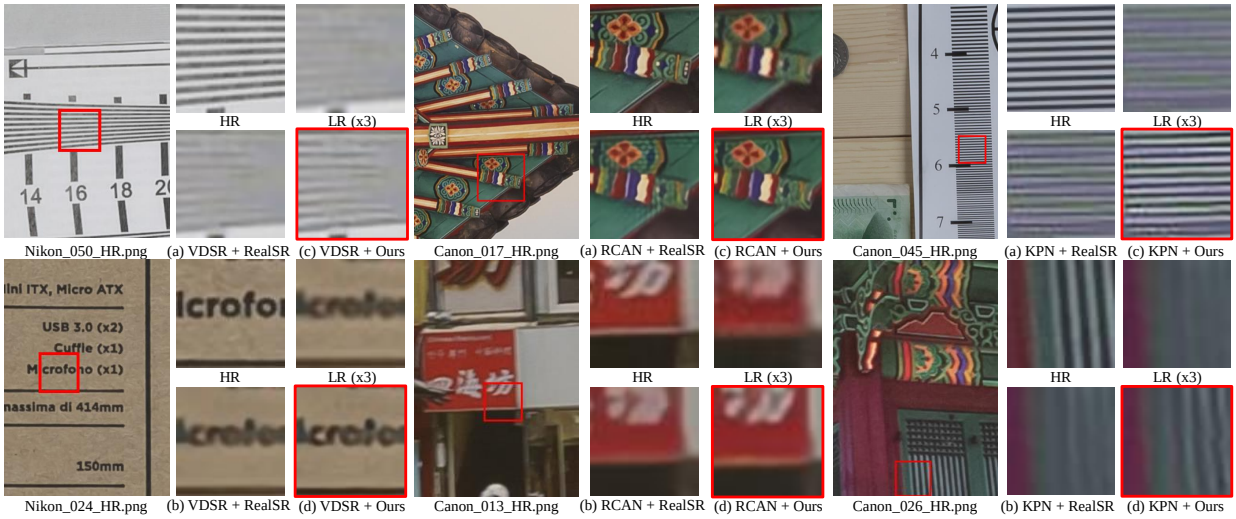


*Figure 7.* Qualitative comparisons on RealSR dataset with degradation level $\times 3$. (a)-(b) are generated from the SR models (VDSR, RCAN, KPN) trained with the images on degradation levels $\times 2$, $\times 4$ dataset in RealSR. (c)-(d) are generated from the SR models trained with our generated dataset. The leftmost column, center column and rightmost column shows the results of VDSR, RCAN and KPN, respectively.

*Table 5.* Ablation study on design choices for the LR-consistency and IB losses with HAN (evaluated on the RealSR testset $\times 3$).

| Method | RealSR $\times 3$ test set | |
|---|---|---|
| | PSNR | SSIM |
| $\lambda_{\text{LR-cons}} = 1$ | 30.56 | 0.8635 |
| $\lambda_{\text{LR-cons}} =$ **10** | **30.68** | **0.8644** |
| $\lambda_{\text{LR-cons}} = 100$ | 30.52 | 0.8611 |
| $\lambda_{\text{IB}} = 0.1$ | 30.43 | 0.8635 |
| $\lambda_{\text{IB}} =$ **1** | **30.68** | **0.8644** |
| $\lambda_{\text{IB}} = 10$ | N/A | N/A |
| No downsampling | 30.20 | 0.8511 |
| $2\times$ downsampling | 30.25 | 0.8529 |
| **4× downsampling** | **30.68** | **0.8644** |

higher performance, stressing its effectiveness.

• **Impact of hyper-parameters** We investigate the influence of hyper-parameters in our loss function: $\lambda_{\text{LR-cons}}$, $\lambda_{\text{IB}}$, and downsampling factor in $\mathcal{L}_{\text{LR-cons}}$, as reported in Table 5 in supplementary materials. When the downsampling factor in $\mathcal{L}_{\text{LR-cons}}$ is too small, LR images with intermediate degradation levels are forced to contain high-frequency details that exist in the images $I_{s_1}$ and $I_{s_2}$. When $\lambda_{\text{IB}}$ decreases, generated LR images lose degradation information, making it unsuitable for LR input images. Otherwise, if IB loss is too high, we observed unstable training as the shift of distributions in NF becomes too large, preventing the LR images from being mapped to predefined distributions.

### 4.5. Scale-arbitrary SR networks with InterFlow

Scale-arbitrary networks (Chen et al., 2021; Hu et al., 2019) and InterFlow share the objective of handling LR images with various degradation levels. However, scale-arbitrary networks inherently require train dataset of arbitrary scales and thus have relied on synthetic dataset, limiting their performance on real-world images. We investigate whether our generated dataset enhances the performance of scale-arbitrary networks in modeling arbitrary degradation levels of real-world images.

• **Experimental settings** We perform experiments on two scale-arbitrary networks: MetaSR (Hu et al., 2019) and LIIF (Chen et al., 2021) with EDSR-baseline (Lim et al., 2017) as a backbone. We follow the same model configurations from the official implementations and the same learning configurations from Section 4.1, except that patch size is set as 48 $\times$ 48 to match the settings of MetaSR and LIIF while total iteration is set as 200k for convergence. We generate LR images using five different methods: Bicubic, Linear Interpolation, BSRGAN (Zhang et al., 2021), Real-ESRGAN (Wang et al., 2021), and the proposed InterFlow. For BSRGAN and Real-ESRGAN, we employed the same hyper-parameters from their official implementations to generate degraded images. As with previous experiments, we train networks on RealSR $\times 2$, $\times 4$ train set and evaluate them on RealSR $\times 3$ test set. For reference, we also measure the upper bound performance by training on images with the same degradation level as test set: RealSR $\times 3$ train set. Note that MetaSR and LIIF cannot use RealSR dataset as is because these networks expect LR images to have varying resolutions corresponding to degradation level, while RealSR LR and HR images have the same resolution to align pixels for correcting lens distortions and misaligned contents (Cai et al., 2019). To train them with RealSR, LR images are further downsampled via a bicubic downsampling operation with a downscale factor that is the same as degradation level $s$. However, there may be additional degradations introduced by downsampling operation,

*Table 6.* SR results by scale-arbitrary networks on the RealSR test set $\times 3$.

| Model | Train set | LR Generation Method | RealSR $\times 3$ test set | | |
|---|---|---|---|---|---|
| | | | PSNR | SSIM | LPIPS |
| MetaSR | RealSR $\times 3$ | None | 30.43 | 0.8572 | 0.3311 |
| | RealSR $\times 1$ | Bicubic (baseline) | 28.99 | 0.8165 | 0.3488 |
| | RealSR $\times 2, \times 4$ | Linear Interpolation | 30.30 | 0.8541 | 0.3231 |
| | RealSR $\times 1$ | BSRGAN | 28.15 | 0.8114 | 0.3867 |
| | RealSR $\times 1$ | Real-ESRGAN | 26.90 | 0.8077 | 0.3813 |
| | RealSR $\times 2, \times 4$ | InterFlow (Ours) | **30.42** | **0.8569** | **0.3222** |
| LIIF | RealSR $\times 3$ | None | 30.43 | 0.8578 | 0.3325 |
| | RealSR $\times 1$ | Bicubic (baseline) | 29.00 | 0.8167 | 0.3490 |
| | RealSR $\times 2, \times 4$ | Linear Interpolation | 30.33 | 0.8560 | 0.3271 |
| | RealSR $\times 1$ | BSRGAN | 28.23 | 0.8133 | 0.3875 |
| | RealSR $\times 1$ | Real-ESRGAN | 27.07 | 0.8090 | 0.3817 |
| | RealSR $\times 2, \times 4$ | InterFlow (Ours) | **30.44** | **0.8581** | **0.3263** |

*Table 7.* SR results by scale-arbitrary networks on the DRealSR test set $\times 3$.

| Model | Train set | LR Generation Method | DRealSR $\times 3$ test set | | |
|---|---|---|---|---|---|
| | | | PSNR | SSIM | LPIPS |
| MetaSR | RealSR $\times 3$ | None | 31.14 | 0.8705 | 0.3596 |
| | RealSR $\times 1$ | BSRGAN | 30.07 | 0.8579 | 0.3831 |
| | RealSR $\times 1$ | Real-ESRGAN | 29.00 | 0.8489 | 0.3825 |
| | RealSR $\times 2, \times 4$ | InterFlow (Ours) | **31.16** | **0.8651** | **0.3559** |
| LIIF | RealSR $\times 3$ | None | 31.15 | 0.8705 | 0.3616 |
| | RealSR $\times 1$ | BSRGAN | 30.21 | 0.8601 | 0.3823 |
| | RealSR $\times 1$ | Real-ESRGAN | 29.00 | 0.8493 | 0.3827 |
| | RealSR $\times 2, \times 4$ | InterFlow (Ours) | **31.15** | **0.8662** | **0.3613** |

possibly even damaging original degradation information. Thus, generating real-world LR images with scale resolutions according to degradation levels is an interesting future research direction.

• **Results** Table 6 reports the results after training networks with four different versions of train sets: RealSR $\times 3$ for upper bound reference; bicubic interpolation on RealSR $\times 1$ HR images; linear interpolation on RealSR $\times 2$ and $\times 4$; and dataset generated by Interflow (Ours) trained on RealSR $\times 2$ and $\times 4$. Note that generating LR images by bicubic interpolation on $\times 1$ HR images is the same procedure used by MetaSR, LIIF, and other conventional SR networks, serving as a baseline. Inferior performance by baselines verifies that conventional synthetic LR generation methods are not suitable for real-world images. In contrast, dataset by our InterFlow greatly improves the performance and even achieves performance near upper bound. The results demonstrate that InterFlow is complementary to scale-arbitrary networks and helps them better handle unseen degradation levels.

To further demonstrate the generalizability of our method, we compare BSRGAN and Real-ESRGAN on an unseen test set that was not used during the training phase. We adopt DRealSR (Wei et al., 2020) dataset because it contains images captured in diverse environments and settings, using cameras from various manufacturers such as Canon, Sony, Nikon, Olympus, and Panasonic. In Table 7, we observe that both MetaSR and LIIF, trained with our synthetic dataset, demonstrate superior performance on the unseen test set compared to BSRGAN and Real-ESRGAN. They even achieve performance levels similar to those trained with the specific scale factor (*i.e.,* RealSR $\times 3$). Therefore, our results demonstrate that using the proposed dataset for training allows the SR networks to have a broader general applicability.

While the BSRGAN and Real-ESRGAN methods for synthesizing datasets for SR have the advantage of making neural networks robust to various types of degradations, they have a fundamental limitation. These methods do not accurately model the real-world degradations that need to be considered when simulating real camera zooming. As a result, neural networks trained on the generated dataset from BSRGAN and Real-ESRGAN tend to produce restored images with lower quality, as indicated by lower PSNR, SSIM, and LPIPS values, compared to neural networks trained on our synthesized dataset (see Table 6 and Table 7). Furthermore, in the context of arbitrary scale single image super-resolution (SISR) tasks, the dataset synthesis methods using BSRGAN or Real-ESRGAN require determining complex hyperparameters for degradation mixtures at each scale factor. This becomes difficult and impractical in real-world scenarios.

## 5. Limitations and Conclusion

In this work, we propose to generate LR images including realistic artifacts for unseen degradation levels with our Interflow approach. One limitation of our approach is that we cannot explicitly determine the blending weight $a$ for a specific degradation level since real degradation changes non-linearly. Thus, alleviating this limitation is what we will study in the near future.

Nevertheless, we addressed a novel task, namely, generating LR-HR paired dataset for SR with unavailable degradation levels. To solve this problem, we proposed a constrained NF which enforces LR consistency and discriminability on each degradation level. Despite the difficulty of the task, our InterFlow successfully generates accurate LR images for unseen degradation levels and can be used to successfully train SR networks for the real-world SR task. Furthermore, if additional real-world degradation such as noise (Abdelhamed et al., 2018; Anaya & Barbu, 2018) or image compression artifacts are integrated as a unified dataset, we believe that our model will be able to generate a larger class of complex image degradation to train deep neural networks.

## Acknowledgements

This work was supported by by Hanhwa Vision Co. Ltd. and Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No.2022-0-00156, Fundamental research on continual meta-learning for quality enhancement of casual videos and their 3D metaverse transformation).

## References


Abdelhamed, A., Lin, S., and Brown, M. S. A high-quality denoising dataset for smartphone cameras. In *CVPR*, 2018.

Abdelhamed, A., Brubaker, M. A., and Brown, M. S. Noise flow: Noise modeling with conditional normalizing flows. In *ICCV*, 2019.

Agustsson, E. and Timofte, R. Ntire 2017 challenge on single image super-resolution: Dataset and study. In *CVPRW*, 2017.

Anaya, J. and Barbu, A. Renoir–a dataset for real low-light image noise reduction. *Journal of Visual Communication and Image Representation*, 2018.

Ardizzone, L., Mackowiak, R., Rother, C., and Köthe, U. Training normalizing flows with the information bottleneck for competitive generative classification. *NeurIPS*, 2020.

Bell-Kligler, S., Shocher, A., and Irani, M. Blind super-resolution kernel estimation using an internal-gan. In *NeurIPS*, 2019.

Berthelot*, D., Raffel*, C., Roy, A., and Goodfellow, I. Understanding and improving interpolation in autoencoders via an adversarial regularizer. In *ICLR*, 2019.

Cai, J., Zeng, H., Yong, H., Cao, Z., and Zhang, L. Toward real-world single image super-resolution: A new benchmark and a new model. In *ICCV*, 2019.

Chang, H., Yeung, D.-Y., and Xiong, Y. Super-resolution through neighbor embedding. In *CVPR*, 2004.

Chen, Y., Liu, S., and Wang, X. Learning continuous image representation with local implicit image function. In *CVPR*, 2021.

Chen, Y.-C., Xu, X., Tian, Z., and Jia, J. Homomorphic latent space interpolation for unpaired image-to-image translation. In *CVPR*, 2019.

Dinh, L., Sohl-Dickstein, J., and Bengio, S. Density estimation using real NVP. In *ICLR*, 2017.

Dong, C., Loy, C. C., He, K., and Tang, X. Image super-resolution using deep convolutional networks. *IEEE TPAMI*, 2015.

El Helou, M., Zhou, R., and Süsstrunk, S. Stochastic frequency masking to improve super-resolution and denoising networks. In *ECCV*, 2020.

Glasner, D., Bagon, S., and Irani, M. Super-resolution from a single image. In *ICCV*, 2009.

Hu, X., Mu, H., Zhang, X., Wang, Z., Tan, T., and Sun, J. Meta-sr: A magnification-arbitrary network for super-resolution. In *CVPR*, 2019.

Ignatov, A., Kobyshev, N., Timofte, R., Vanhoey, K., and Van Gool, L. Dslr-quality photos on mobile devices with deep convolutional networks. In *ICCV*, 2017.

Jang, G., Lee, W., Son, S., and Lee, K. M. C2n: Practical generative noise modeling for real-world denoising. In *ICCV*, 2021.

Ji, X., Cao, Y., Tai, Y., Wang, C., Li, J., and Huang, F. Real-world super-resolution via kernel estimation and noise injection. In *CVPRW*, 2020.

Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., and Aila, T. Analyzing and improving the image quality of stylegan. In *CVPR*, 2020.

Kim, J., Lee, J. K., and Lee, K. M. Accurate image super-resolution using very deep convolutional networks. In *CVPR*, 2016.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. In *ICLR*, 2015.

Kingma, D. P. and Dhariwal, P. Glow: Generative flow with invertible 1x1 convolutions. In *NeurIPS*, 2018.

Kirichenko, P., Izmailov, P., and Wilson, A. G. Why normalizing flows fail to detect out-of-distribution data. *NeurIPS*, 2020.

Köhler, T., Bätz, M., Naderi, F., Kaup, A., Maier, A., and Riess, C. Toward bridging the simulated-to-real gap: Benchmarking super-resolution on real data. *IEEE TPAMI*, 2019.

Lee, J. and Jin, K. H. Local texture estimator for implicit representation function. In *CVPR*, 2022.

Liang, J., Cao, J., Sun, G., Zhang, K., Van Gool, L., and Timofte, R. Swinir: Image restoration using swin transformer. In *ICCV*, 2021.

Lim, B., Son, S., Kim, H., Nah, S., and Mu Lee, K. Enhanced deep residual networks for single image super-resolution. In *CVPRW*, 2017.

Liu, Y., Sangineto, E., Chen, Y., Bao, L., Zhang, H., Sebe, N., Lepri, B., Wang, W., and De Nadai, M. Smoothing the disentangled latent style space for unsupervised image-to-image translation. In *CVPR*, 2021.

Lugmayr, A., Danelljan, M., Van Gool, L., and Timofte, R. Srflow: Learning the super-resolution space with normalizing flow. In *ECCV*, 2020.

Mei, Y., Fan, Y., and Zhou, Y. Image super-resolution with non-local sparse attention. In *CVPR*, 2021.

Mittal, A., Soundararajan, R., and Bovik, A. C. Making a "completely blind" image quality analyzer. *IEEE Signal processing letters*, 2012.

Nalisnick, E., Matsukawa, A., Teh, Y. W., Gorur, D., and Lakshminarayanan, B. Do deep generative models know what they don't know? *arXiv preprint arXiv:1810.09136*, 2018.

Niu, B., Wen, W., Ren, W., Zhang, X., Yang, L., Wang, S., Zhang, K., Cao, X., and Shen, H. Single image super-resolution via a holistic attention network. In *ECCV*, 2020.

Patashnik, O., Wu, Z., Shechtman, E., Cohen-Or, D., and Lischinski, D. Styleclip: Text-driven manipulation of stylegan imagery. In *ICCV*, 2021.

Qu, C., Luo, D., Monari, E., Schuchert, T., and Beyerer, J. Capturing ground truth super-resolution data. In *ICIP*. IEEE, 2016.

Rezende, D. and Mohamed, S. Variational inference with normalizing flows. In *International conference on machine learning*. PMLR, 2015.

Shen, Y., Gu, J., Tang, X., and Zhou, B. Interpreting the latent space of gans for semantic face editing. In *CVPR*, 2020.

Tishby, N., Pereira, F. C., and Bialek, W. The information bottleneck method. *arXiv preprint physics/0004057*, 2000.

Wang, X., Yu, K., Dong, C., and Loy, C. C. Recovering realistic texture in image super-resolution by deep spatial feature transform. In *CVPR*, 2018.

Wang, X., Xie, L., Dong, C., and Shan, Y. Real-esrgan: Training real-world blind super-resolution with pure synthetic data. In *ICCVW*, 2021.

Wang, Y., Wan, R., Yang, W., Li, H., Chau, L.-P., and Kot, A. Low-light image enhancement with normalizing flow. In *AAAI*, 2022.

Wei, P., Xie, Z., Lu, H., Zhan, Z., Ye, Q., Zuo, W., and Lin, L. Component divide-and-conquer for real-world image super-resolution. In *ECCV*, 2020.

Wolf, V., Lugmayr, A., Danelljan, M., Van Gool, L., and Timofte, R. Deflow: Learning complex image degradations from unpaired data with conditional flows. In *CVPR*, 2021.

Xiao, J., Yong, H., and Zhang, L. Degradation model learning for real-world single image super-resolution. In *ACCV*, 2020.

Yang, J., Shen, S., Yue, H., and Li, K. Implicit transformer network for screen content image continuous super-resolution. *NeurIPS*, 2021.

Zhang, K., Zuo, W., Chen, Y., Meng, D., and Zhang, L. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE TIP*, 2017.

Zhang, K., Liang, J., Van Gool, L., and Timofte, R. Designing a practical degradation model for deep blind image super-resolution. In *ICCV*, 2021.

Zhang, X., Chen, Q., Ng, R., and Koltun, V. Zoom to learn, learn to zoom. In *CVPR*, 2019.

Zhang, Y., Li, K., Li, K., Wang, L., Zhong, B., and Fu, Y. Image super-resolution using very deep residual channel attention networks. In *ECCV*, 2018a.

Zhang, Y., Tian, Y., Kong, Y., Zhong, B., and Fu, Y. Residual dense network for image super-resolution. In *CVPR*, 2018b.

Zhou, B., Zhao, H., Puig, X., Fidler, S., Barriuso, A., and Torralba, A. Scene parsing through ade20k dataset. In *CVPR*, 2017.

Zhu, J., Shen, Y., Zhao, D., and Zhou, B. In-domain gan inversion for real image editing. In *ECCV*, 2020.

# Supplementary Material

## A. InterFlow Architecture

For our InterFlow, we use a slightly modified version of SRFlow (Lugmayr et al., 2020). SRFlow is a conditional normalizing flow (NF) model based on the architectures of Glow (Kingma & Dhariwal, 2018) and RealNVP (Dinh et al., 2017). Similar to Glow, the forward operation of SRFlow consists of $L$ levels. At each level, it begins with a squeeze operation to reduce the resolution of the input image while increasing the depth. Then, $K$ flow steps are followed, where each step consists of its own distinct invertible operation. Finally, it ends with a split operation which only passes a part of the output to the next level and uses the rest as latent variables. For our InterFlow, we use $K = 16$, and $L = 2$. Each flow step consists of four trainable operations. They are Actnorm (Kingma & Dhariwal, 2018), invertible 1x1 convolution (Kingma & Dhariwal, 2018), affine injector (Lugmayr et al., 2020), and conditional affine coupling (Lugmayr et al., 2020). Especially, affine injector and conditional affine coupling are conditional operations that make the architecture conditionally invertible. We let $h$ denote the encoder of our InterFlow, In SRFlow, the input resolution for the SRFlow encoder is smaller than the output image of the flow function. In contrast, in our InterFlow, the input resolution of $h$ and the resolution of the output image are the same. Therefore, we add a squeeze operation at the beginning of the encoder $h$ to match the resolution and change the input depth of the first convolution.

## B. Training and Evaluation

### B.1. Train-time and Run-time

As our Interflow is a slightly modified version of SRFlow, it has few more parameters compared to SRFlow network. It has 31 M parameters with settings of the final version mentioned in our main manuscript. It takes 1.5 days to train with an NVIDIA V100 GPU and takes 0.35 seconds to generate a single LR image which has a resolution of 160 $\times$ 160.

### B.2. Training InterFlow with RealSR $\times$2, $\times$4

Experiments conducted in Figure 4-6 and Table 1-3 in the manuscript are designed to evaluate how well models can handle unseen degradation levels. To this end, we assume that we only have access to $\times 2$ and $\times 4$ degradation levels during training, then evaluate the models on scale factor $\times 3$. Therefore, we train our InterFlow using RealSR train dataset with degradation level s $\times 2$ and $\times 4$ captured by a Canon camera. With the trained InterFlow, we generate LR images by changing the blending weight $a$ for degradation levels $2 < s' < 4$ using RealSR train images captured by a Nikon camera for $\times 2$ and $\times 4$ degradation levels. These synthesized LR images are used to train the SR networks. To evaluate the performance of the trained SR networks, we use RealSR test dataset for $\times 3$ degradation level. Notably, the test dataset for the evaluation includes images captured by both Canon and Nikon cameras. Experimental results from this setting are shown in Figure 4-6, and Table 1-3 in the manuscript.

## C. Qualitative results

### C.1. Results on RealSR $\times$2.5 and RealSR $\times$3.5

Experiments in Table 4 in the manuscript, and Figure 8 in the supplementary material aim to evaluate under more unseen degradation levels: $\times 2.5$ and $\times 3.5$. To do so, SR networks are evaluated on RealSR $\times 2.5$ and $\times 3.5$ testset that we collected. Our collected RealSR $\times 2.5$ and $\times 3.5$ testset contains a total of 80 HR-LR image pairs taken in outdoor and indoor environments. Some sample images are shown in Figure 12.

We first train our InterFlow, which is then used to generate LR images for $2 < s' < 4$ to train the SR networks. Notably, we train our InterFlow using all the RealSR Canon train-images (*i.e.,* , $S \in \{2, 3, 4\}$), and generate LR images using RealSR Nikon train-images for all scale factors (*i.e.,* , $S \in \{2, 3, 4\}$).


*Figure 8.* Qualitative comparisons on RealSR $\times 2.5$ and $\times 3.5$ test images that we acquired using the same data acquisition process for RealSR dataset (Cai et al., 2019). (a)-(b) SR results from RCAN and HAN trained with conventional dataset only. (c)-(d) SR results from RCAN and HAN trained with our synthetic dataset.

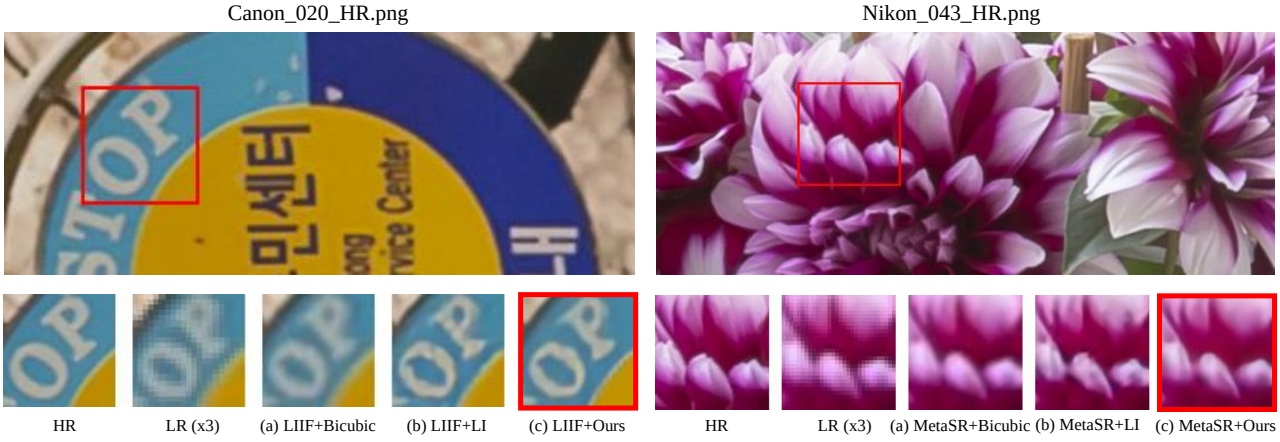


*Figure 9.* Qualitative comparisons on RealSR $\times 3$ test images with the two repersentive scale-arbitrary SR networks: LIIF (Chen et al., 2021) and MetaSR (Hu et al., 2019) with EDSR-baseline (Lim et al., 2017). Each SR result of (a),(b),(c) is generated from the SR networks with different sets of synthesized LR images: bicubic interpolation on RealSR $\times 1$ HR (a), linear interpolation on RealSR $\times 2$ and $\times 4$ (b), and synthesized from our Interflow using RealSR $\times 2$ and $\times 4$ (c).

In the visual results in Figure 8, we see that the SR networks trained with our synthetic datasets output much sharper and clearer images with more details, while the SR networks trained with conventional datasets produce unexpected artifacts (see HAN $\times 2.5$).

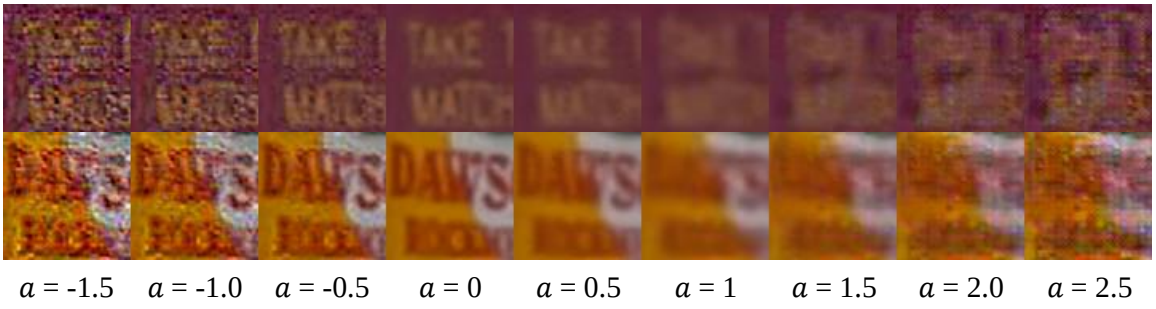


*Figure 10.* Visual results with extra-polation. Generated LR images with blending weights where $-1.5 \leq a \leq 2.5$. Note that we train our InterFlow only for $0 < a < 1$.

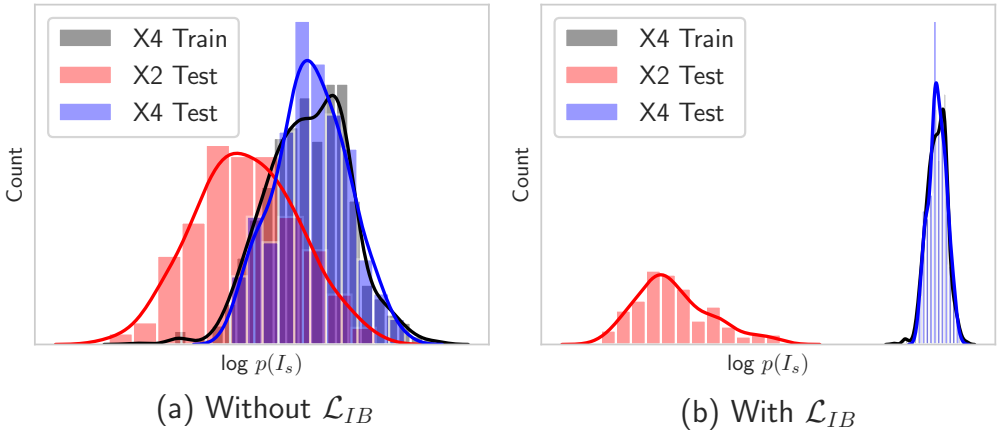


*Figure 11.* Visualization of log-likelihood histograms for learned normal distribution $\mathcal{N}(\mu_4, \sigma_4^2)$. Note that Interflow is trained on RealSR Canon $\times 2$, $\times 4$ trainset, and the statistics mean $\mu_4$ and variance $\sigma_4^2$ are trained during training-phase. Each log-likelihood value is calculated by transformed latent variable $z_s$ and $\mathcal{N}(\mu_4, \sigma_4^2)$, respectively, where $s \in \{2, 4\}$. (a) The figure illustrates learned distributions without using the IB loss $\mathcal{L}_{IB}$. Test samples for $s = 4$ do not follow the learned distribution $\mathcal{N}(\mu_4, \sigma_4^2)$. (b) The figure illustrates learned distributions with the IB loss $\mathcal{L}_{IB}$. Test sample distribution for $s = 4$ is similar to the learned distribution $\mathcal{N}(\mu_4, \sigma_4^2)$.

### C.2. Results on RealSR $\times$3.0 with scale-arbitrary SR networks

To show that our proposed real-world LR generation with arbitrary degradation levels is complementary to scale-arbitrary SR networks, we show the performance improvement on real-world LR images when scale-arbitrary SR networks are trained with our generated LR images. We perform evaluation on two representative scale-arbitrary SR networks: LIIF (Chen et al., 2021) and MetaSR (Hu et al., 2019). While Table 6 in the main manuscript provides the quantitative results, we provide the qualitative counterpart in Figure 9. We can observe that when the networks are trained in a standard manner (LR generation by bicubic downsampling), the estimated SR have blurred and unclear edges. By contrast, when the networks are trained with LR images generated by linear interpolation, the estimated HR images seem to have noticeable and undesirable artifacts. Lastly, when the networks are trained with our method, the networks provide sharper images with lest artifacts. Along with Table 6 in the main manuscript, the qualitative results further validate the effectiveness of our proposed framework InterFlow in generating real-world LR images with arbitrary degradation level and thereby facilitating the robustness of scale-arbitrary networks to real-world LR images with unknown degradation level.

### C.3. Extrapolation in Latent Space

In this section, we stress-test our InterFlow outside of its operating region to investigate how well InterFlow can generalize. Specifically, we analyze how the quality of generated LR images changes if extrapolation is performed in the latent space instead of interpolation.

*Table 8.* SR results on several real-world test sets with scale $\times 3$ ($s = 3$).

| Model | LR Generation Method | NIQE ↓ | | |
|---|---|---|---|---|
| | | OST300 | DPED-iphone | ADE20K-val |
| Bicubic | - | 7.4154 | 7.7566 | 7.3238 |
| LIIF | BSRGAN | 7.4710 | 8.6454 | 7.8580 |
| | Real-ESRGAN | 6.8803 | 8.2202 | 7.2591 |
| | InterFlow (Ours) | **5.9209** | **6.4287** | **6.0690** |

In Figure 10, we generate LR images by changing the blending weight, where we use $a = \{-1.5, -1.0, -0.5, 0, 0.5, 1, 1.5, 2.0, 2.5\}$. From the results, we see that generated LR images by extrapolation in the latent space for $a < 0$ include more details and sharp edges, and the results for $a > 1$ include more blurs, as we expected. However, unexpected noise is added to the resulting images, and removing these artifacts to achieve more realistic LR images will be our future work.

## D. Visualizing latent distribution

In this section, we analyze our InterFlow by visualizing the distributions in its learned latent space. In particular, we visualize the histograms of log-likelihood of RealSR Canon trainset and testset. In (a), we can observe that distribution of $\times 2$, $\times 4$ Test is significantly overlapped. Furthermore, some portions of $\times 2$ Test are assigned with a higher likelihood compared to $\times 4$ Train. It is a well-known problem that NF with naïve NLL loss cannot discriminate between in and out distribution when NF transform images into latent variables (Nalisnick et al., 2018; Kirichenko et al., 2020). That is, this entangled degradation information inhibits NF from generating distinctive LR images with unseen, intermediate degradation levels. On the contrary, in (b), through InterFlow trained with $\mathcal{L}_{LR-cons}, \mathcal{L}_{IB}$, we can observe that each distribution of log-likelihoods for $\times 2$, $\times 4$ Test is separated. With the help of $\mathcal{L}_{LR-cons}$ that maintains smooth transition along sparse density area, InterFlow is able to generate more discriminated features such as real-world noises or artifacts. Thus, the visualization analysis further corroborates the effectiveness of our proposed framework InterFlow to generate synthetic datasets with arbitrary and continuous degradation levels.

## E. More quantitative comparisons

In this section, we compare BSRGAN and Real-ESRGAN on other real-world datasets such as OST300 (Wang et al., 2018), DPED (Ignatov et al., 2017), and ADE20K (Zhou et al., 2017). As the datasets do not have corresponding ground-truth HR images, we measure restored image quality using non-reference image quality assessment such as NIQE (Mittal et al., 2012). In Table 8, we follow the same experiment settings from Table 7 where the test datasets are not used in a training phase, and our method shows superior performance compared to BSRGAN and Real-ESRGAN on all of three unseen test sets.

## F. Sample images of extended RealSR dataset

We collect extended RealSR dataset to measure SR performance for degradation levels $\times 2.5$ and $\times 3.5$. It contains a total of 80 HR-LR image pairs taken in outdoor and indoor environments. Some sample images are shown in Figure 12.


*Figure 12.* Sample images in our own RealSR $\times 2.5$ and RealSR $\times 3.5$ testsets.