DINN360: Deformable Invertible Neural Network for Latitude-aware 360° Image Rescaling

Yichen Guo[1], Mai Xu[1]*, Lai Jiang[2]*, Leonid Sigal[2], Yunjin Chen
[1] School of Electronic and Information Engineering, Beihang University, Beijing, China
[2] Department of Computer Science, University of British Columbia, Vancouver, Canada

Abstract

With the rapid development of virtual reality, 360° images have gained increasing popularity. Their wide field of view necessitates high resolution to ensure image quality. This, however, makes it harder to acquire, store and even process such 360° images. To alleviate this issue, we propose the first attempt at 360° image rescaling, which refers to downscaling a 360° image to a visually valid low-resolution (LR) counterpart and then upscaling to a high-resolution (HR) 360° image given the LR variant. Specifically, we first analyze two 360° image datasets and observe several findings that characterize how 360° images typically change along their latitudes. Inspired by these findings, we propose a novel deformable invertible neural network (INN), named DINN360, for latitude-aware 360° image rescaling. In DINN360, a deformable INN is designed to downscale the LR image, and project the high-frequency (HF) component to the latent space by adaptively handling various deformations occurring at different latitude regions. Given the downscaled LR image, the high-quality HR image is then reconstructed in a conditional latitude-aware manner by recovering the structure-related HF component from the latent space. Extensive experiments over four public datasets show that our DINN360 method performs considerably better than other state-of-the-art methods for 2×, 4× and 8× 360° image rescaling.

1. Introduction

With the rapid development of virtual reality, 360° images have gained increasing popularity. Different from 2D images, 360° images cover a scene with a wide range of 360° × 180° views, requiring high resolution for ensuring the image quality. However, this also makes it considerably more costly to acquire, store and even process such high-resolution (HR) 360° images. To address these issues, it is necessary to conduct 360° image rescaling, which consists of image downscaling for generating low-resolution (LR) images with visually valid information and image upscaling for reconstructing HR 360° images. Different from image super-resolution (SR) that only upscales from LR images, image rescaling can directly utilize the texture information from the input HR 360° images, and therefore achieves better reconstruction results.

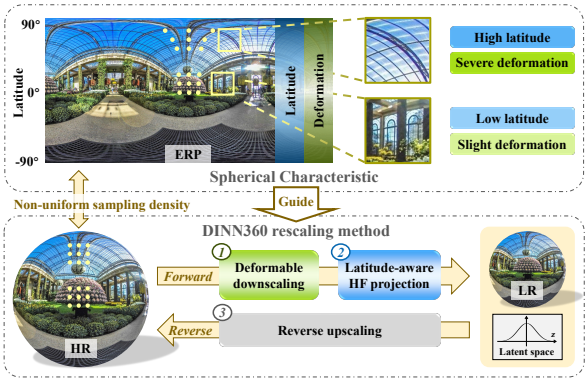


Figure 1. Motivation and pipeline of our DINN360 method. The non-uniform sampling density causes various deformations at different latitude regions, and this guides the design of our DINN360 model. Finally, the HR 360° image can be rescaled from the corresponding LR image and latent space.

Recently, 2-dimensional (2D) image rescaling has received increasing research interests [17, 21, 23, 36, 43, 44], due to its promising application potential. Specifically, Kim *et al.* [17] proposed a task-aware auto-encoder-based framework including a task-aware downscaling (TAD) model and a task-aware upscaling (TAU) model. In this work, the procedures of downscaling and upscaling are implemented by two individual deep neural networks (DNNs), and then they are jointly optimized. Xiao *et al.* [44] proposed an image rescaling framework based on invertible neural network (INN), in which downscaling and upscaling are regarded as invertible procedures. Different from 2D images, as shown in Fig. 1, 360° images contain various types of deformation at different latitude regions, due to the non-uniform sampling density of the sphere-to-plain projection. Therefore, it is inappropriate to directly apply the existing 2D rescaling methods on 360° images (see analysis in Section 3). Hence, it is necessary to develop a specialized framework

*Corresponding authors: Mai Xu (MaiXu@buaa.edu.cn), Lai Jiang (jianglai.china@buaa.edu.cn)

for rescaling of the 360° image, by fully considering its spherical characteristics.

This paper is the first attempt at 360° image rescaling. First, we conduct data analysis to find how the spherical characteristics of 360° images, such as texture complexity and high-frequency (HF) components, change along with the latitude. Inspired by our findings, we propose a deformable invertible neural network (DINN360) for latitude-aware 360° image rescaling. Specifically, as shown in Fig. 1, deformable downscaling with a set of invertible deformable blocks is developed in DINN360 to learn the adaptive receptive fields. As such, the 360° image can be downscaled in a deformation-adaptive manner. Subsequently, the bijective projection is conducted with the developed INN structure for the HF component extracted from the downscaling procedure, such that the texture details can be better recovered for the following upscaling. More importantly, a novel latitude-aware conditional mechanism is developed for the projection, in order to preserve the HF component of 360° images in a latitude-aware manner. Given the invertible structures of downscaling and HF projection, the 360° image can be reversely upscaled. Moreover, a new backflow training protocol is developed to reduce the information gap between the forward and reverse flows of the INN structure. The extensive experimental results show that our DINN360 outperforms state-of-the-art image rescaling and 360° SR methods for $2\times$, $4\times$ and $8\times$ rescaling over 4 public datasets. The codes are available at https://github.com/gyc9709/DINN360. The main contributions of this paper are three-fold.

- We find how the low-level characteristics of 360° images change along with its latitude, benefiting the designs of our DINN360 method.
- We propose a novel INN framework for 360° image rescaling, with the developed invertible deformable blocks to handle various 360° deformations.
- We develop a latitude-aware conditional mechanism in our framework, to better preserve the HF component of 360° images in a latitude-aware manner.

2. Related Work

Rescaling of 2D images. Image rescaling refers to downscaling an HR image into a visually valid LR image and then reconstructing the HR image plausibly from this LR image. As a recently emerged topic, there exists only a few works for rescaling of 2D images [17, 21, 23, 31, 36, 43, 44]. Specifically, most of the existing works [17, 21, 36] develop and jointly train two individual DNN structures for downscaling and upscaling, respectively. For instance, Kim *et al.* [17] designed a task-aware downscaling model to generate SR-friendly LR images, using the auto-encoder architecture. Similarly, Li *et al.* [21] proposed downscaling the HR image by a simple yet efficient DNN, and then upscaling the LR image by a modified EDSR structure [24]. Sun *et al.* [36] proposed predicting the downsampling kernel instead of directly generating the LR image, via a ResamplerNet for predicting both the weights and offsets of the sampling kernel. In addition to the individual downscaling and upscaling structures, Xiao *et al.* [43, 44] proposed an INN-based method to regard the downscaling and upscaling as invertible procedures. Based on [44], Liang *et al.* [23] proposed adding a conditional flow in INN to guarantee the dependency of high- and low-frequency (LF) components in the rescaled image. Unfortunately, there exists no rescaling work for 360° images; especially lacking works that consider spherical characteristics for 360° image rescaling.

SR of 360° images. Similar to image rescaling, SR reconstructs the HR image directly from the LR image. Benefiting from the great success of deep learning, Dong *et al.* [9] proposed a pioneering DNN structure called SRCNN for SR on 2D images with a great improvement over traditional methods. After that, a set of DNN-based methods have been developed for SR on 360° images [4, 10, 18, 25, 30]. Specifically, most of these works [5, 25, 29, 30] improve the SR performance by considering the latitude-based priors of 360° images. For instance, the latitude-aware weighted loss is adopted in [25, 30] to encourage the network to place more importance on the low-latitude regions. Similarly, Nishiyama *et al.* [29] proposed concatenating the latitude-aware weight with the input LR image, as the additional information for SR. Different from [29], Deng *et al.* [5] proposed adopting distinct upscaling factors for different latitude regions. There also exists many works for 360° scenes which address the latitude-aware distortion by tangent patch partition [20, 34] and contourlet transform [2, 35], etc. However, the existing SR methods cannot be used directly for image rescaling, since they are only able to upscale, but unable to downscale the image. More importantly, SR methods neglect the HF components from the input HR images, leading to the inferior reconstruction.

3. Analysis

In this section, we conduct analysis over the F-360iSOD [46] and SUN360 [42] datasets. Then, we obtain the following findings about the low-level characteristics of 360° images, to benefit the design of our DINN360 method.

Finding 1: *In 360° images, low-latitude regions tend to contain more textures, leading to larger HF components.*

Analysis: We investigate the spherical characteristics from the aspects of space and frequency domains. Specifically, each 360° image is first horizontally divided into 8 strips with uniform latitude range of 22.5°. Then, we apply the gray-level co-occurrence matrix (GLCM) [6] to measure the entropy of each strip as its texture complexity. The entropy of each latitude strip is shown by the color bar

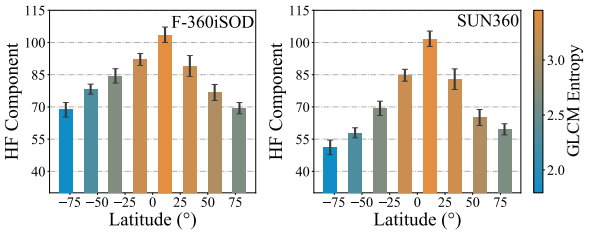


Figure 2. Magnitude of HF components and GLCM entropy at different latitude regions of 360° images.

of Fig. 2. As can be seen, the GLCM entropy distributes the highest at low-latitude regions (near 0°), and decreases along with the increased latitude. Similarly, the HF component of each strip is obtained via Haar transformation [1], and is shown in Fig. 2. As can be seen, the HF components distribute similarly to the GLCM entropy. This indicates that the HF component is highly related to the texture complexity. Moreover, some examples of the low- and high-latitude regions are illustrated in the supplementary, which shows the same results of Fig. 2. The above results complete the analysis of *Finding 1*.

Finding 2: *In 360° images, the larger HF components at low-latitude regions result in worse rescaling performance for the existing 2D rescaling methods.*

Analysis: Here, we investigate how the HF components of 360° images influence the performance of rescaling at different latitude regions. To this end, we first follow the way of *Finding 1* to calculate the average HF component at each latitude strip. Then, we implement 3 state-of-the-art 2D image rescaling methods (HCFlow [23], CAR [36] and TAD [17]) and the traditional Bicubic interpolation method [28] on $4\times$ image rescaling. Subsequently, we measure both the peak-signal-noise-ratios (PSNRs) and error maps between the ground-truth (GT) and rescaled HR images at different latitude regions. As illustrated in Fig. 3, the PSNRs of all 2D rescaling methods exhibit the same trend that dramatically decreases from high-latitude to low-latitude regions, corresponding to the increased magnitude of the HF component. Moreover, as shown in Fig. 4, there are larger errors for low-latitude regions, which have complex texture regions (i.e., the higher magnitudes of the HF components). These results imply that the higher magnitudes of HF components at low-latitude regions cause worse rescaling performance for 2D rescaling methods. The above analysis complete the analysis of *Finding 2*.

4. Method

4.1. Framework Overview

Pipeline of DINN360. Given an input 360° HR image $\hat{\mathbf{x}}$, DINN360 is proposed on the top of the INN structure for generating both downscaled LR image $\mathbf{y}$ and rescaled HR image $\mathbf{x}$. However, according to the Nyquist-Shannon sampling theorem [33], the HF component of $\hat{\mathbf{x}}$ is lost during the downscaling procedure. To overcome this issue, similar

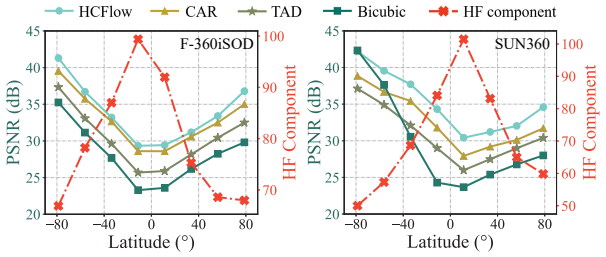


Figure 3. Results of PSNRs and HF components of $4\times$ rescaled HR images by 2D methods at different latitude regions.

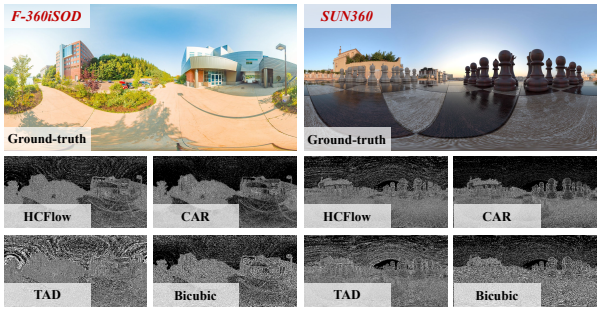


Figure 4. Error maps of the GT and rescaled HR images from the existing methods, over F-360iSOD and SUN360 datasets.

to flow-based models [7, 11, 15, 19], our DINN360 learns to project the HF component $\mathbf{h}$ into a latent variable $\mathbf{z}$ following a prior distribution, and then recover $\mathbf{h}$ during upscaling. This way, the rescaling procedure can be formulated as a bijective transformation: $\hat{\mathbf{x}} \leftrightarrow [\mathbf{y}; \mathbf{z}]$. The pipeline of DINN360 is illustrated in Fig. 5 and introduced below.

(1) **Deformable downscaling $\hat{\mathbf{x}} \rightarrow [\mathbf{y}; \mathbf{h}]$.** In this stage, the input HR image $\hat{\mathbf{x}}$ is decomposed into the downscaled LR image $\mathbf{y}$ and HF component $\mathbf{h}$. Specifically, as shown in Fig. 5, the Haar wavelet transformation is first conducted over $\hat{\mathbf{x}}$ to obtain the HF and LF information. Then, the invertible deformable (ID) blocks are developed in DINN360 to fuse both HF and LF components in an invertible manner, and generate the refined HF and LF components, denoted as $\mathbf{h}$ and $\mathbf{y}$. The refined LF component is also output as the downscaled LR image.

(2) **Latitude-aware HF projection $[\mathbf{y}; \mathbf{h}] \rightarrow \mathbf{z} \sim \mathcal{N}(0, 1)$.** Subsequently, the HF component $\mathbf{h}$ is learned to project to the latent variable $\mathbf{z}$, considering the spherical characteristics of the downscaled image $\mathbf{y}$. To be specific, as shown in Fig. 5, the content and latitude conditions are extracted from $\mathbf{y}$, and then concatenated with $\mathbf{h}$, as the input to a set of developed invertible projection (IP) blocks. Finally, the projected latent variable $\mathbf{z}$ is supervised to fit a normalized Gaussian distribution $\mathcal{N}(0, 1)$.

(3) **Reverse upscaling $[\mathbf{y}; \widetilde{\mathbf{z}} \sim \mathcal{N}(0, 1)] \rightarrow \mathbf{x}$.** For the reverse upscaling procedure, the downscaled image $\mathbf{y}$ is combined with a randomly sampled latent variable $\widetilde{\mathbf{z}} \sim \mathcal{N}(0, 1)$, and then input into the reverse IP blocks, to recover the HF component $\widetilde{\mathbf{h}}$. Finally, after the reverse ID blocks and Haar transformation, the HF component and downscaled image are inversely transformed to reconstruct the HR image $\mathbf{x}$.

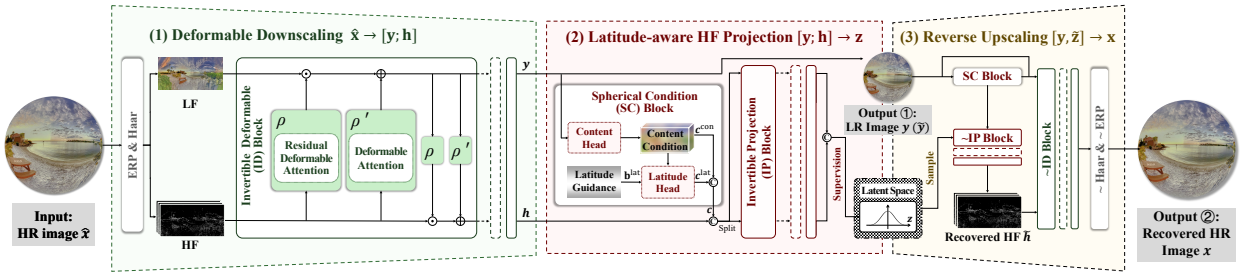


Figure 5. Pipeline of DINN360 in the setting of 2× rescaling. Here, ∼ID block and ∼IP block indicate the corresponding reverse structures.

To sum up, the pipeline of DINN360 can be written as,

$$\hat{\mathbf{x}} \rightarrow [\mathbf{y};\mathbf{h}] \rightarrow [\mathbf{y};\mathbf{z}] \rightarrow [\mathbf{y};\widetilde{\mathbf{h}}] \rightarrow \mathbf{x}. \tag{1}$$

Invertible designs. The invertibility of our DINN360 method is achieved in terms of invertible structures and latent space projection [7]. First, the structure of invertible blocks (ID and IP blocks) endow DINN360 the ability for reverse procedure through the same model and parameters. Let $\mathbf{a}_1^l$ and $\mathbf{a}_2^l$ denote the input of the l-th invertible block, the corresponding output $\left[\mathbf{a}_1^{l+1}, \mathbf{a}_2^{l+1}\right]$ can be obtained by

$$\begin{aligned} \mathbf{a}_1^{l+1} &= \mathbf{a}_1^l \odot \exp(\rho(\mathbf{a}_2^l)) + \rho'(\mathbf{a}_2^l), \\ \mathbf{a}_2^{l+1} &= \mathbf{a}_2^l \odot \exp(\rho(\mathbf{a}_1^{l+1})) + \rho'(\mathbf{a}_1^{l+1}), \end{aligned} \tag{2}$$

where $\rho(\cdot)$ and $\rho'(\cdot)$ denote the learnable scale and translation functions, and $\odot$ is the Hadamard product. The details of the reverse procedure are stated in the supplementary.

Second, the lost HF component $\mathbf{h}$ of downscaling is refined and projected into a prior distribution, through the developed ID and IP blocks in DINN360, respectively. Then, benefiting from the reverse structure, the lost HF component is obtained given a randomly sampled latent variable $\widetilde{\mathbf{z}}$ from the prior distribution, and then used for reversely reconstructing the HR image $\mathbf{x}$. In other word, our DINN360 method learns to encode and preserve the HF component in the prior latent space, which makes the downscaling and upscaling procedures invertible.

4.2. Deformable Downscaling

Due to the non-uniform projection of 360° images, there exist various deformations at different latitude regions. To address this issue, a set of ID blocks is designed in the downscaling procedure, in order to learn the adaptive receptive fields for different types of deformation. Note that the $2^N\times$ rescaling is achieved in our DINN360 method by conducting N of $2\times$ rescaling. Taking the n-th downscaling as an example, the downscaled image $\mathbf{y}_{n-1}$ from the last procedure is further downscaled to $\mathbf{y}_n$, as follows,

$$\mathbf{h}_n, \mathbf{y}_n = I_{\mathrm{D}}(\mathrm{Haar}(\mathbf{y}_{n-1})). \tag{3}$$

Here, $\mathbf{h}_n$ is the HF component; $\mathrm{Haar}(\cdot)$ and $I_{\mathrm{D}}(\cdot)$ denote the Haar transformation and ID blocks, respectively. After N times of Eq. (3), the final downscaled LR image $\mathbf{y}_N$ is yielded. For the reverse procedure, as shown in Fig. 5, the recovered HF component $\widetilde{\mathbf{h}}_n$ and the LR image $\widetilde{\mathbf{y}}_n$ are input to the reverse ID blocks $I_{\mathrm{D}}^{-1}(\cdot)$ and reverse Haar transformation $\mathrm{Harr}^{-1}(\cdot)$. Mathematically, the reverse procedure of deformable downscaling can be formulated as,

$$\widetilde{\mathbf{y}}_{n-1} = \mathrm{Haar}^{-1}(I_{\mathrm{D}}^{-1}(\widetilde{\mathbf{h}}_n, \widetilde{\mathbf{y}}_n)). \tag{4}$$

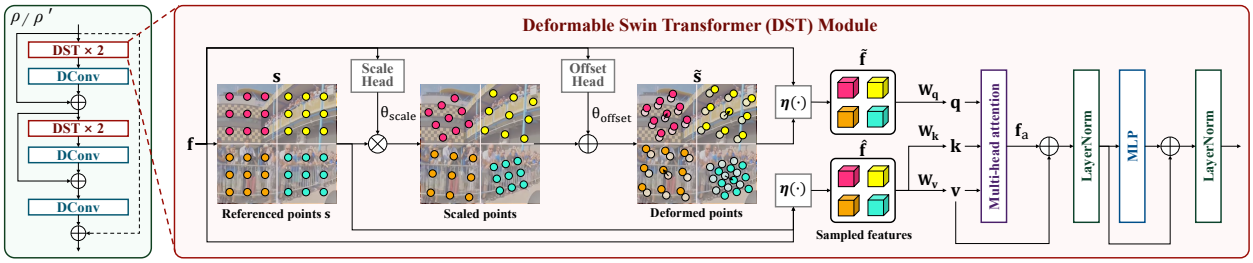
To be more specific, $I_{\mathrm{D}}(\cdot)$ consists of 4 cascaded invertible blocks as introduced in Eq. (2). For each invertible block, the detailed structures of functions $\rho(\cdot)$ and $\rho'(\cdot)$ are shown in Fig. 6-(a). The functions are built in a deformable manner, upon the residual structure with deformable convolution (DConv) layers [3] and the developed deformable swin transformer (DST) modules.

Deformable swin transformer (DST) module. The DST module is built on the top of the advanced structure of swin transformer [27], which is widely used in vision tasks [16, 22, 26, 45]. Different from the traditional swin transformer [27, 38], a deformable transformation is learned in the DST module, for projecting the query $\mathbf{q}$, key $\mathbf{k}$ and value $\mathbf{v}$. As a result, the various geometry deformations occurring at different latitude regions in the 360° image can be aware when calculating the self-attention of each image patch. Specifically, for each input feature $\mathbf{f} \in \mathbb{R}^{H_{\mathbf{f}}\times W_{\mathbf{f}}\times C_{\mathbf{f}}}$, the referenced sampling points $\mathbf{s} = \{(\mu_i, \nu_i)\}_{i=1}^{I}$ are generated from the uniform grids according to [41], where I denotes the number of sampling points. Then, as shown in Fig. 6, the transformation factors $\boldsymbol{\theta}_{\mathrm{scale}} \in \mathbb{R}^{2\times 2}$ and $\boldsymbol{\theta}_{\mathrm{offset}} \in \mathbb{R}^{2\times 1}$ are learned from the input feature $\mathbf{f}$, via the developed scale and offset heads. Both heads are consisted of two learnable fully-connected layers and two hyperbolic activation layers. Given the transformation factors $\boldsymbol{\theta}_{\mathrm{scale}}$ and $\boldsymbol{\theta}_{\mathrm{offset}}$, the deformed sampling points $\widetilde{\mathbf{s}} = \{(\widetilde{\mu}_i, \widetilde{\nu}_i)\}_{i=1}^{I}$ can be calculated as,

$$\left[\widetilde{\mu}_i, \quad \widetilde{\nu}_i\right]^T = \left[\boldsymbol{\theta}_{\mathrm{scale}}, \quad \boldsymbol{\theta}_{\mathrm{offset}}\right] \left[\mu_i, \quad \nu_i, \quad 1\right]^T. \tag{5}$$

Subsequently, given the referenced and the learned deformed sampling points $\mathbf{s}$ and $\widetilde{\mathbf{s}}$, the referenced feature $\hat{\mathbf{f}}$ and deformed feature $\widetilde{\mathbf{f}}$ can be obtained by the Bilinear sampling function $\eta(\cdot;\cdot)$ in [14], as follows,

$$\begin{aligned} \hat{\mathbf{f}}_i &= \sum_{h=1,w=1}^{H_{\mathbf{f}},W_{\mathbf{f}}} \mathbf{f}_{h,w}\eta(\mu_i; h)\eta(\nu_i; w), \\ \widetilde{\mathbf{f}}_i &= \sum_{h=1,w=1}^{H_{\mathbf{f}},W_{\mathbf{f}}} \mathbf{f}_{h,w}\eta(\widetilde{\mu}_i; h)\eta(\widetilde{\nu}_i; w), \end{aligned} \tag{6}$$



(a) Architecture of functions ρ(·) and ρ′(·) in ID block (b) Architecture of DST Module

Figure 6. Architectures of (a) functions $\rho(\cdot)$ and $\rho'(\cdot)$ in ID blocks; (b) DST module.

where $\eta(a;b) = \max(0, 1-|a-b|)$. In the above equation, $\mathbf{f}_{h,w}$ denotes the pixel value at coordinate (h, w). Besides, $\hat{\mathrm{f}}_i$ and $\widetilde{\mathrm{f}}_i$ denote the sampled pixel according to the i-th sampling point of $\mathbf{s}$ and $\widetilde{\mathbf{s}}$, respectively. Compared with $\hat{\mathbf{f}}$, the deformed feature $\widetilde{\mathbf{f}}$ is adaptive to the geometry deformation of 360° image by learning with deformable sampling. Then, the tokens of query $\mathbf{q}$, key $\mathbf{k}$ and value $\mathbf{v}$ are calculated by

$$\mathbf{q} = \widetilde{\mathbf{f}}\mathbf{W}_{\mathbf{q}}, \mathbf{k} = \hat{\mathbf{f}}\mathbf{W}_{\mathbf{k}}, \mathbf{v} = \hat{\mathbf{f}}\mathbf{W}_{\mathbf{v}}, \tag{7}$$

where $\mathbf{W}_{\mathbf{q}}$, $\mathbf{W}_{\mathbf{k}}$ and $\mathbf{W}_{\mathbf{v}}$ denote the learnable parameter matrices. This way, the query token can be embedded with the deformation information from $\widetilde{\mathbf{f}}$. As shown in Fig. 6-(b), given the tokens of $\mathbf{q}$, $\mathbf{k}$ and $\mathbf{v}$, the self-attention can be calculated through the multi-head attention layer. Then, the attended feature $\mathbf{f}_{\mathrm{a}}$ is further processed after two LayerNorm layers and a multilayer perceptron layer, as the input to the subsequent structures in $\rho(\cdot)$ or $\rho'(\cdot)$ (see Fig. 6-(a)).

4.3. Latitude-aware HF Projection

After the deformable downscaling, a spherical condition (SC) block and a set of IP blocks are developed for latitude-aware HF projection, i.e., bijectively projecting the HF component to the latent space in a latitude-aware manner. As shown in Fig. 5, for the n-th rescaling, the SC block $G_{\mathrm{SC}}(\cdot)$ is designed to extract the spherical features $\mathbf{c}_n$ from the downscaled images $\{\mathbf{y}_i\}_{i=n}^{N}$, as the conditions of IP blocks. Then, the n-th HF component $\mathbf{h}_n$ is conditioned on $\mathbf{c}_n$, and input to IP blocks $I_{\mathrm{P}}(\cdot)$ as formulated in Eq. (2). Note that $\rho(\cdot)$ and $\rho'(\cdot)$ of Eq. (2) in this section are implemented by dense blocks [13], instead of the architectures in ID blocks (see Section 4.2). This way, the 360° characteristics are able to perform as the external constraint, which conditionally guide the projection between the HF component and latent space. Consequently, the HF component $\mathbf{h}_n$ can be projected to the corresponding latent variable $\mathbf{z}_n$ as,

$$\mathbf{z}_n = I_{\mathrm{P}}(\mathbf{h}_n, \mathbf{c}_n), \text{ where } \mathbf{c}_n = G_{\mathrm{SC}}(\mathbf{y}_n, \cdots, \mathbf{y}_N). \tag{8}$$

For the reverse procedure, the previous upscaled images $\{\widetilde{\mathbf{y}}_i\}_{i=n}^{N}$ ($\mathbf{y}_N = \widetilde{\mathbf{y}}_N$) are first input to the SC block $G_{\mathrm{SC}}(\cdot)$ to yield the conditions. Then, as shown in Fig. 5, the randomly sampled latent variable $\widetilde{\mathbf{z}}_n$ is input to the reverse IP blocks $I_{\mathrm{P}}^{-1}(\cdot)$, for generating the recovered HF component $\widetilde{\mathbf{h}}_n$. That is, the reverse procedure of latitude-aware HF projection can be formulated as,

$$\widetilde{\mathbf{h}}_n = I_{\mathrm{P}}^{-1}\left(\widetilde{\mathbf{z}}_n, G_{\mathrm{SC}}(\widetilde{\mathbf{y}}_n, \cdots, \mathbf{y}_N)\right). \tag{9}$$

Spherical condition (SC) block. As shown in Fig. 5, the SC block contains two heads for extracting content condition $\mathbf{c}_n^{\mathrm{con}}$ and latitude condition $\mathbf{c}_n^{\mathrm{lat}}$. Then, $\mathbf{c}_n^{\mathrm{lat}}$ and $\mathbf{c}_n^{\mathrm{con}}$ are concatenated as the overall condition $\mathbf{c_n} = \left[\mathbf{c}_n^{\mathrm{con}}, \mathbf{c}_n^{\mathrm{lat}}\right]$ for the HF projection in Eq. (8). The details about the condition heads are discussed as follows.

- **Content head.** A content head is developed to extract the content-related condition $\mathbf{c}_n^{\mathrm{con}}$ for HF projection. Specifically, three learnable residual-in-residual dense blocks [39] in $\gamma_n^{\mathrm{con}}(\cdot)$ is adopted to learn the content information from the downscaled images $\{\mathbf{y}_i\}_{i=n}^{N}$ [1] as,

$$\mathbf{c}_n^{\mathrm{con}} = \gamma_n^{\mathrm{con}}\left([\mathbf{y}_n, \mathbf{y}_{n+1}, \cdots, \mathbf{y}_N]\right). \tag{10}$$

- **Latitude head.** According to Finding 1, the distribution of HF components is highly related to the latitudes for 360° images. Thus, we follow [29, 37] to generate the latitude-aware distortion map $\mathbf{b}_n^{\mathrm{lat}}$ as follows,

$$\mathbf{b}_n^{\mathrm{lat}}(u,:) = \cos\left((u - \frac{H}{2^{n+1}} + \frac{1}{2})\frac{2^n\pi}{H}\right). \tag{11}$$

  In the above equation, u denotes the vertical ordinate value for each pixel in $\mathbf{b}_n^{\mathrm{lat}}$, while H is the height of the original HR image. Then, a latitude head with dense block $\gamma_n^{\mathrm{lat}}(\cdot)$ is built to generate the latitude condition $\mathbf{c}_n^{\mathrm{lat}}$ from the combination of $\mathbf{b}_n^{\mathrm{lat}}$ and content condition $\mathbf{c}_n^{\mathrm{con}}$, i.e., $\mathbf{c}_n^{\mathrm{lat}} = \gamma_n^{\mathrm{lat}}(\left[\mathbf{b}_n^{\mathrm{lat}}, \mathbf{c}_n^{\mathrm{con}}\right])$.

4.4. Loss and Training Protocol

Loss functions. In general, 360° image rescaling aims to generate a visually valid LR image and then reconstruct the HR image from the LR image and latent space. Therefore, the loss functions for training our DINN360 model include HR, LR, and latent variable losses. (1) *HR loss*: for the rescaled HR image $\mathbf{x}$ and its corresponding GT $\hat{\mathbf{x}}$ (the input HR image), a weighted ℓ_1 loss is applied to measure the distance between two images as $\mathcal{L}_{\mathrm{HR}} = \ell_1(\boldsymbol{\omega} \odot \mathbf{x}, \boldsymbol{\omega} \odot \hat{\mathbf{x}})$.

[1] Note that the LR images $\{\mathbf{y}_i\}_{i=n+1}^{N}$ are upscaled to the same size with $\mathbf{y}_n$ by interpolation before concatenation.

Algorithm 1: Training process for $2\times$ rescaling.

Input: HR image $\hat{\mathbf{x}}$, LR image $\hat{\mathbf{y}}$ and distortion map $\hat{\mathbf{c}}^{lat}$.
Output: Trained $I_{\mathrm{D}}(\cdot)$ $I_{\mathrm{P}}(\cdot)$ and $G_{\mathrm{SC}}(\cdot)$.
Variables: Training variables Φ, latent variables $\mathbf{z}$, $\widetilde{\mathbf{z}}$.
Parameters: λ_{H}, λ_{L}, $\lambda_{\mathbf{z}}$, α and learning rate lr.

Initialize Φ with Gaussian initialization.
while *Step* $<$ *max_steps* **do**
  $\mathbf{y}, \mathbf{h} = I_{\mathrm{D}}(\hat{\mathbf{x}})$.
  $\mathbf{c} = [\mathbf{c}^{\mathrm{con}}, \mathbf{c}^{\mathrm{lat}}] = G_{\mathrm{SC}}(\mathbf{y})$.
  if *Step* $<$ *backflow_steps* **then**
    $\widetilde{\mathbf{h}} = I_{\mathrm{P}}^{-1}(\widetilde{\mathbf{z}}, \mathbf{c})$.
    $\mathbf{h} = \alpha\widetilde{\mathbf{h}} + (1-\alpha)\mathbf{h}$.
  end
  $\mathbf{z} = I_{\mathrm{D}}(\mathbf{h}, \mathbf{c})$.
  $\widetilde{\mathbf{h}} = I_{\mathrm{P}}^{-1}(\widetilde{\mathbf{z}}, \mathbf{c})$.
  $\mathbf{x} = I_{\mathrm{D}}^{-1}(\mathbf{y}, \widetilde{\mathbf{h}})$.
  $\mathcal{L} = \lambda_{\mathrm{H}}\ell_1(\mathbf{x}, \hat{\mathbf{x}}) + \lambda_{\mathrm{L}}\ell_2(\mathbf{y}, \hat{\mathbf{y}}) + \lambda_{\mathbf{z}}\ell_2(\mathbf{z}, \widetilde{\mathbf{z}})$.
  $\mathbf{\Phi} \leftarrow \mathbf{\Phi} - lr \cdot \nabla_{\mathbf{\Phi}}\mathcal{L}$.
end
return $\mathbf{\Phi}$.

Note that $\boldsymbol{\omega}$ denotes the pixel-wise weights for highlighting the importance of the low-latitude regions according to [29]. (2) *LR loss*: for the downscaled LR images $\{\mathbf{y}_n\}_{n=1}^N$ from DINN360, we use the Bicubic downscaled images $\{\hat{\mathbf{y}}_n\}_{n=1}^N$ from HR image $\hat{\mathbf{x}}$ as the ground truth, for calculating the ℓ_2 pixel loss $\mathcal{L}_{\mathrm{LR}} = \sum_{n=1}^N \ell_2(\mathbf{y}_n, \hat{\mathbf{y}}_n)$. Recall that $2^N\times$ rescaling is conducted by N of $2\times$ rescaling, and $\mathbf{y}_n$ denotes the downscaled LR image of the n-th downscaling. (3) *Latent variable loss*: for the generated latent variables $\{\mathbf{z}_n\}_{n=1}^N$ from our latitude-aware HF projection (see Eq. (8)), the KL divergence $\mathrm{D}_{\mathrm{KL}}(\cdot)$ is calculated between the latent variable and a normalized Gaussian distribution $\mathcal{N}(0,1)$, i.e., $\mathcal{L}_{\mathrm{latent}} = \sum_{n=1}^N \mathrm{D}_{\mathrm{KL}}\left(\mathbf{z}_n || \mathcal{N}(0,1)\right)$. Consequently, the overall loss can be formulated as

$$\mathcal{L} = \lambda_{\mathrm{H}}\mathcal{L}_{\mathrm{HR}} + \lambda_{\mathrm{L}}\mathcal{L}_{\mathrm{LR}} + \lambda_{\mathbf{z}}\mathcal{L}_{\mathrm{latent}}, \tag{12}$$

where λ_{H}, λ_{L}, and $\lambda_{\mathbf{z}}$ are the hyper-parameters for balancing each individual loss.

Backflow training protocol. As discussed in Section 4.3, the forward and reverse procedures can be represented as $\mathbf{z} = I_{\mathrm{P}}(\mathbf{h})$ and $\widetilde{\mathbf{h}} = I_{\mathrm{P}}^{-1}(\widetilde{\mathbf{z}})$ for simplicity, where $\mathbf{h}$ and $\widetilde{\mathbf{h}}$ are the input and recovered HF components. Here, $\mathbf{z}$ is the projected latent variable, while $\widetilde{\mathbf{z}}$ is the sampled latent variable from the latent space for the reverse procedure. The gap between $\mathbf{z}$ and $\widetilde{\mathbf{z}}$ results in the difference between $\mathbf{h}$ and $\widetilde{\mathbf{h}}$, further causing the recovery loss between the input HR and final rescaled images (see Eq. (3) and Eq. (4)). To bridge this gap, we propose a new backflow training protocol for INN structures in DINN360, inspired by the proportional-integral-derivative (PID) control [40] in classic automatic control system. In PID control, a proportional system error is added to the input, as the negative feedback. As such, the system error between the input and output can be reduced. As shown in Fig. 7, the forward and reverse procedures in our DINN360 can be regarded as an invertible system, the error of which is assumed to be sufficiently small, i.e., $\widetilde{\mathbf{h}} = \mathbf{h}$. In backflow training protocol, the proportional difference between $\widetilde{\mathbf{h}}$ and $\mathbf{h}$ is fed back to the input:

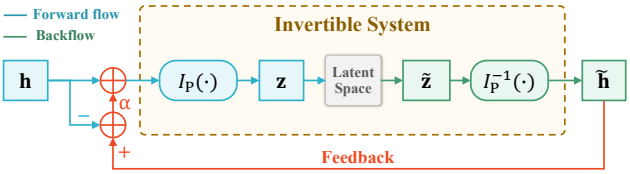


Figure 7. Framework of our backflow training protocol for the invertible structure of HF projection.

$$\mathbf{z} = I_{\mathrm{P}}\left(\mathbf{h} + \alpha(\widetilde{\mathbf{h}} - \mathbf{h})\right) = I_{\mathrm{P}}\left(\alpha I_{\mathrm{P}}^{-1}(\widetilde{\mathbf{z}}) + (1-\alpha)\mathbf{h}\right), \tag{13}$$

where α is the feedback proportion. Algorithm 1 summarizes our backflow training protocol. It is worth noting that the proposed backflow training protocol is also potential to be used in other INN works [7, 8, 11, 12, 19, 32].

5. Experiment

5.1. Settings

In this section, we conduct the experiments to validate the effectiveness of our DINN360 method. Here, we adopt the training set of ODISR [5] to train DINN360, which includes 1,000 high-quality 360° images at 2K resolution. Then, the trained model is directly evaluated over the test set of ODISR with 100 images, and other three 360° datasets for generalization evaluation: SUN360 [42], F-360iSOD [46] and YouTube360 [29]. For training DINN360, we apply the stochastic gradient descent algorithm with the Adam optimizer to update parameters. The hyper-parameters can be found in the supplementary.

Following the settings of [25] and [5], we quantitatively evaluate the weighted-to-spherically-uniform PSNR (WS-PSNR) and weighted-to-spherically-uniform structural similarity index (WS-SSIM) on the Y channel of YCbCr image color representation. Finally, we compare the performance of our DINN360 method on $2\times$, $4\times$ and $8\times$ rescaling with (1) traditional interpolation methods: Bicubic, Bilinear and Lanczos; (2) Bicubic downscaling followed by 360° SR methods: 360SR [30], 360SISR [29] and LAU-Net [5]; (3) 2D rescaling methods: TAD & TAU [17], CAR & EDSR [24, 36], IRN [43] and HCFlow [23]. Note that all 360° SR and rescaling methods are based on DNNs, and they are retrained over ODISR dataset with the same settings as our DINN360 method, except for CAR and LAU-Net, due to the lack of training codes.

5.2. Performance Evaluation

Quantitative results. First, we compare the quantitative results of the rescaled 360° HR images by our and other

Table 1. Results of WS-PSNR / WS-SSIM ($\times 10^{-2}$) on the rescaled HR images of our DINN360 and compared methods over four datasets. The best results are in **bold** and the underline scores represent the second-best results.

Scale	Method	ODISR [5]	SUN360 [42]	F-360iSOD [46]	YouTube360 [29]
2×	Bicubic	29.46 ± 2.54 / 86.23 ± 5.05	30.06 ± 2.46 / 87.92 ± 4.85	30.68 ± 4.53 / 87.43 ± 7.23	34.93 ± 4.92 / 94.82 ± 4.47
	Bilinear	28.94 ± 2.45 / 83.15 ± 5.83	29.39 ± 2.40 / 85.09 ± 6.08	29.97 ± 4.23 / 84.61 ± 8.61	33.20 ± 4.21 / 9.255 ± 6.06
	Lanczos	28.58 ± 2.54 / 84.04 ± 5.57	29.16 ± 2.47 / 85.72 ± 5.48	29.86 ± 4.56 / 85.44 ± 8.11	34.26 ± 5.13 / 93.95 ± 5.24
	Bicubic & 360SR [30]	27.05 ± 2.36 / 80.46 ± 4.32	27.69 ± 2.20 / 81.55 ± 4.87	26.08 ± 4.17 / 78.08 ± 5.39	32.12 ± 3.77 / 89.84 ± 4.85
	Bicubic & 360SISR [29]	30.81 ± 2.90 / 87.44 ± 5.17	32.72 ± 2.79 / 90.53 ± 5.10	31.33 ± 4.81 / 89.63 ± 6.28	37.62 ± 5.21 / 96.23 ± 5.08
	TAD & TAU [17]	35.84 ± 3.28 / 96.12 ± 8.12	37.70 ± 2.68 / 97.17 ± 1.10	33.94 ± 5.11 / 93.87 ± 4.47	39.50 ± 4.08 / 98.22 ± 1.00
	CAR & EDSR [24, 36]	33.00 ± 3.51 / 91.31 ± 4.41	35.68 ± 3.37 / 93.91 ± 4.07	35.38 ± 5.48 / 93.05 ± 5.09	40.49 ± 5.24 / 97.75 ± 2.55
	IRN [43]	40.51 ± 3.52 / 98.63 ± 0.71	42.72 ± 2.73 / 99.11 ± 0.32	39.83 ± 5.74 / 97.83 ± 2.05	46.15 ± 4.02 / 99.50 ± 0.32
	HCFlow [23]	42.05 ± 3.79 / 99.02 ± 0.57	45.05 ± 3.00 / 99.49 ± 0.24	40.53 ± 5.78 / 97.92 ± 1.99	50.56 ± 3.07 / 99.71 ± 0.10
	DINN360	**42.64 ± 3.87 / 99.13 ± 0.52**	**45.72 ± 3.00 / 99.56 ± 0.21**	**40.77 ± 5.88 / 97.93 ± 2.21**	**50.75 ± 3.07 / 99.73 ± 0.10**
4×	Bicubic	25.39 ± 2.28 / 72.27 ± 7.45	25.38 ± 2.33 / 73.75 ± 8.83	26.16 ± 3.91 / 73.75 ± 12.38	28.29 ± 3.80 / 83.73 ± 10.58
	Bilinear	26.24 ± 2.27 / 72.96 ± 7.54	26.22 ± 2.29 / 74.72 ± 8.85	26.85 ± 3.78 / 74.30 ± 12.38	28.92 ± 3.53 / 83.94 ± 10.44
	Lanczos	24.97 ± 2.28 / 70.69 ± 7.64	24.99 ± 2.33 / 71.95 ± 9.05	25.77 ± 3.94 / 72.10 ± 12.84	27.97 ± 3.85 / 82.65 ± 11.02
	Bicubic & 360SR	25.42 ± 2.26 / 71.06 ± 6.89	25.42 ± 2.16 / 72.46 ± 8.64	25.19 ± 3.69 / 70.79 ± 9.83	28.43 ± 3.26 / 83.06 ± 9.36
	Bicubic & 360SISR	27.03 ± 2.45 / 76.15 ± 7.97	27.81 ± 2.44 / 80.45 ± 9.39	27.45 ± 4.35 / 78.79 ± 11.66	30.96 ± 3.87 / 89.36 ± 10.75
	TAD & TAU	28.98 ± 2.51 / 82.69 ± 5.91	29.70 ± 2.47 / 84.86 ± 6.21	28.71 ± 4.55 / 81.34 ± 10.40	33.24 ± 4.61 / 92.48 ± 6.08
	CAR & EDSR	29.61 ± 2.86 / 82.82 ± 6.76	31.32 ± 2.82 / 86.60 ± 7.49	31.33 ± 4.94 / 85.30 ± 9.10	34.85 ± 4.69 / 93.08 ± 6.45
	IRN	30.86 ± 3.06 / 87.47 ± 5.56	32.69 ± 2.92 / 90.41 ± 5.41	32.58 ± 5.19 / 88.95 ± 7.29	36.85 ± 4.78 / 95.86 ± 4.07
	HCFlow	31.48 ± 3.16 / 89.07 ± 5.02	33.62 ± 3.03 / 92.00 ± 4.78	32.40 ± 5.79 / 88.44 ± 8.85	40.31 ± 4.44 / 97.72 ± 2.13
	DINN360	**31.92 ± 3.26 / 89.90 ± 4.82**	**34.19 ± 3.12 / 92.77 ± 4.48**	**32.93 ± 5.90 / 89.34 ± 8.82**	**40.55 ± 4.29 / 97.89 ± 1.89**
8×	Bicubic	23.25 ± 2.19 / 64.10 ± 8.64	22.92 ± 2.21 / 65.18 ± 10.24	23.45 ± 3.48 / 64.12 ± 15.04	24.98 ± 3.06 / 74.70 ± 12.89
	Bilinear	24.16 ± 2.19 / 65.35 ± 8.65	23.81 ± 2.19 / 66.77 ± 10.20	24.25 ± 3.41 / 65.42 ± 14.89	25.78 ± 2.96 / 75.86 ± 12.65
	Lanczos	22.95 ± 2.19 / 63.15 ± 8.68	22.65 ± 2.21 / 63.98 ± 10.27	23.19 ± 3.49 / 63.05 ± 15.18	24.77 ± 3.08 / 73.78 ± 13.03
	Bicubic & 360SR	23.61 ± 2.06 / 64.15 ± 8.53	23.28 ± 2.17 / 65.11 ± 10.14	23.19 ± 3.17 / 63.30 ± 13.68	25.02 ± 2.85 / 78.19 ± 12.37
	Bicubic & 360SISR	24.63 ± 2.26 / 67.75 ± 8.99	24.56 ± 2.27 / 70.80 ± 10.66	24.53 ± 3.62 / 68.64 ± 14.76	26.28 ± 3.01/ 80.02 ± 12.73
	Bicubic & LAU-Net [5]	24.37 ± 2.22 / 66.64 ± 8.83	24.21 ± 2.26 / 69.37 ± 10.63	24.18 ± 3.57 / 66.94 ± 14.99	25.81 ± 2.94 / 77.33 ± 12.61
	TAD & TAU	26.36 ± 2.30 / 71.36 ± 7.86	26.50 ± 2.33 / 73.43 ± 9.36	25.94 ± 4.10 / 70.35 ± 14.15	28.36 ± 3.46 / 81.61 ± 11.04
	CAR & EDSR	25.97 ± 2.38 / 69.40 ± 8.82	26.40 ± 2.42 / 72.77 ± 10.75	26.87 ± 4.12 / 71.19 ± 14.36	27.98 ± 3.44 / 79.83 ± 11.27
	IRN	28.06 ± 2.72 / 77.41 ± 8.12	29.48 ± 2.74 / 82.02 ± 9.66	29.55 ± 4.89 / 80.03 ± 11.87	32.16 ± 4.24 / 89.01 ± 9.30
	HCFlow	28.25 ± 2.76 / 78.20 ± 8.00	29.77 ± 2.77 / 82.84 ± 9.42	29.83 ± 4.94 / 80.98 ± 11.47	34.19 ± 4.02 / 91.78 ± 7.14
	DINN360	**28.60 ± 2.86 / 79.17 ± 7.98**	**30.36 ± 2.87 / 84.02 ± 9.36**	**30.29 ± 5.13 / 82.07 ± 11.22**	**34.93 ± 4.17 / 92.58 ± 6.83**

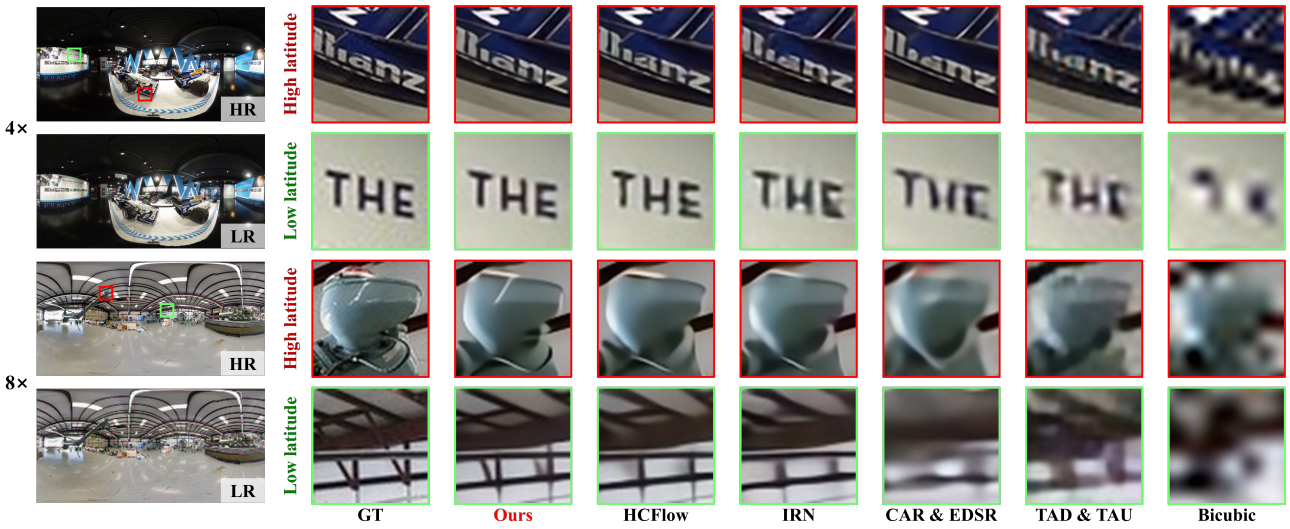


Figure 8. Quantitative results of 4× and 8× image rescaling on ODISR dataset.

compared methods. As shown in Tab. 1, DINN360 achieves the best performance over the ODISR dataset, in terms of both WS-PSNR and WS-SSIM. Specifically, our DINN360 method increases WS-PSNR by at least 0.59dB, 0.44dB and 0.35dB, respectively, for the 2×, 4× and 8× rescaling tasks. Similarly, WS-SSIM is increased by at least 0.0011, 0.0083 and 0.0097, respectively, for the 2×, 4× and 8× rescaling tasks. It is also interesting to see that both our DINN360 and the 2D rescaling methods perform considerably better than the 360° SR methods. This demonstrates that it is effective to utilize the texture information of the input HR image in the task of 360° image rescaling. In a word, the results verify the high quality of the rescaled HR images by our DINN360 method.

Generalization results. To validate the generalization ability, we further test our DINN360 and other compared methods over other three datasets (SUN360, F-360iSOD and YouTube360) without fine-tuning. Tab. 1 shows that our DINN360 method still works best over all three datasets in terms of both WS-PSNR and WS-SSIM. For example, over the SUN360 dataset, DINN360 improves WS-PSNR by at least 0.67dB, 0.57dB and 0.59dB on 2×, 4× and 8× rescal-

Table 2. Results of WS-PSNR and WS-SSIM ($\times 10^{-2}$) in ablation experiments for $4\times$ image rescaling on ODISR dataset.

	Ablation settings	WS-PSNR	WS-SSIM
ID block	w/o DST module	31.79 ± 3.14	89.68 ± 4.17
	w/o deform	31.83 ± 3.63	89.75 ± 4.29
IP block	w/o latitude head	31.85 ± 3.21	89.74 ± 4.31
	w/o content head	31.76 ± 3.21	89.56 ± 4.08
backflow	w/o feedback	31.85 ± 3.32	89.83 ± 4.27
-	**DINN360**	$\mathbf{31.92 \pm 3.26}$	$\mathbf{89.90 \pm 4.82}$

ing tasks, respectively. Similar results can be found for the WS-SSIM metric and other datasets. Our DINN360 method again outperforms all compared methods over these three datasets, indicating its high generalization ability.

Qualitative results. Furthermore, we visualize the subjective results of the downscaled LR and upscaled HR 360° images by our and other rescaling methods, for the $4\times$ and $8\times$ rescaling tasks. Fig. 8 shows these subjective results for some randomly selected 360° images with the low-latitude and high-latitude regions in HR images zoomed in. It can be seen that at different latitude regions, DINN360 is able to better preserve the image details and recover more realistic textures. Specifically, both the character edges and object details recovered by DINN360 is significantly better than other rescaling methods. Besides, the downscaled LR image is also visually valid. This validates the effectiveness of our DINN360 method in the qualitative performance for both downscaled LR and upscaled HR images.

5.3. Ablation Studies

Ablation on the ID block. We evaluate the effectiveness of ID blocks for deformable downscaling in our DINN360 method through two ablation experiments: (1) w/o DST module: set the functions $\rho(\cdot)$ and $\rho'(\cdot)$ in ID block as dense blocks instead of DST modules; (2) w/o deform: set the DST module as normal swin transformer. As can be seen in Tab. 2, the two settings degrade WS-PSNR by 0.13dB and 0.09dB, respectively. Similar results can be found in terms of WS-SSIM. This validates the design of ID blocks is effective in our DINN360 method. Besides, we compare the rescaling performance when implementing different numbers of ID blocks in DINN360. As shown in Fig. 9, the results of WS-PSNR and WS-SSIM slightly improve, when the number of ID blocks is larger than 4. Thus, we set the number of ID blocks as 4 in our DINN360 method.

Ablation on the IP block. We further evaluate the effectiveness of the IP block for latitude-aware HF projection by two ablation experiments: (1) w/o latitude head; (2) w/o content head. As seen in Tab. 2, the WS-PSNR results decrease by 0.07dB and 0.16dB, respectively. This indicates that both latitude and content heads contribute to the final performance of our DINN360 method, in which the content head acts as a more important role in condition generation. Besides, we also study the impact of the number of IP blocks in DINN360. As shown in Fig. 10, similar to ID blocks, we set the number of IP blocks to 4.

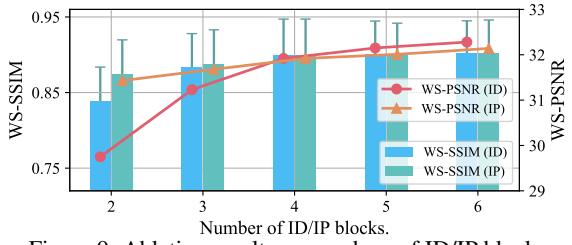


Figure 9. Ablation results on numbers of ID/IP blocks.

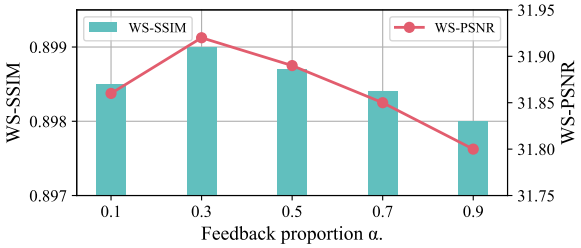


Figure 10. Ablation results on feedback ratio α in Eq. (13).

Ablation on the backflow training protocol. We also validate the effectiveness of the feedback mechanism in the backflow training protocol by directly removing the feedback connection as: w/o feedback in Tab. 2. As can be seen, WS-PSNR degrades by 0.07dB and WS-SSIM decreases by 0.0007, when removing the backflow feedback mechanism. Such results indicate the positive contribution of the backflow protocol in our DINN360. Furthermore, we evaluate the impact of feedback proportion α of Eq. (13) on the rescaling performance. As shown in Fig. 10, the backflow training protocol performs the best at $\alpha = 0.3$.

6. Conclusion

In this paper, we have proposed a DINN360 method for 360° image rescaling. First, we investigated two 360° image datasets and obtained the findings about how spherical characteristics change along with the latitude of 360° images. Motivated by our findings, the structure of DINN360 was developed with three rescaling stages: deformable downscaling, latitude-aware HF projection and reverse upscaling. For deformable downscaling, a deformable INN was designed to generate both the downscaled LR image and the HF component in a deformation-adaptive manner. Then, the latitude-aware HF projection was proposed to learn the bijective projection between the HF component and latent space in a latitude-aware manner. For reverse upscaling, the HR image can be reconstructed through the reversal of the above two stages. Finally, the extensive experimental results validate the effectiveness of our DINN360 method for $2\times$, $4\times$ and $8\times$ 360° image rescaling.

Acknowledgments

This work was supported by NSFC under Grants 62250001, 62231002, Beijing Natural Science Foundation under Grant JQ20020, L223021, and Alibaba Innovative Research.

References

[1] Kenneth R Castleman. *Digital image processing*. Prentice Hall Press, 1996. 3

[2] Lidong Chen, JingTao Lou, Maojun Zhang, Wei Wang, and Yu Liu. Fusion of complementary catadioptric panoramic images based on nonsubsampled contourlet transform. *Optical Engineering*, 50(12):127002–127002, 2011. 2

[3] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 764–773, 2017. 4

[4] Mallesham Dasari, Arani Bhattacharya, Santiago Vargas, Pranjal Sahu, Aruna Balasubramanian, and Samir R Das. Streaming 360-degree videos using super-resolution. In *IEEE INFOCOM 2020-IEEE Conference on Computer Communications*, pages 1977–1986. IEEE, 2020. 2

[5] Xin Deng, Hao Wang, Mai Xu, Yichen Guo, Yuhang Song, and Li Yang. Lau-net: Latitude adaptive upscaling network for omnidirectional image super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9189–9198, 2021. 2, 6, 7

[6] Xin Deng, Hao Wang, Mai Xu, Li Li, and Zulin Wang. Omnidirectional image super-resolution via latitude adaptive network. *IEEE Transactions on Multimedia*, 2022. 2

[7] Laurent Dinh, David Krueger, and Yoshua Bengio. Nice: Non-linear independent components estimation. *arXiv preprint arXiv:1410.8516*, 2014. 3, 4, 6

[8] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real nvp. *arXiv preprint arXiv:1605.08803*, 2016. 6

[9] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Learning a deep convolutional network for image super-resolution. In *European conference on computer vision*, pages 184–199. Springer, 2014. 2

[10] Vida Fakour-Sevom, Esin Guldogan, and Joni-Kristian Kämäräinen. 360 panorama super-resolution using deep convolutional networks. In *Int. Conf. on Computer Vision Theory and Applications (VISAPP)*, volume 1, 2018. 2

[11] Jonathan Ho, Xi Chen, Aravind Srinivas, Yan Duan, and Pieter Abbeel. Flow++: Improving flow-based generative models with variational dequantization and architecture design. In *International Conference on Machine Learning*, pages 2722–2730. PMLR, 2019. 3, 6

[12] Chin-Wei Huang, David Krueger, Alexandre Lacoste, and Aaron Courville. Neural autoregressive flows. In *International Conference on Machine Learning*, pages 2078–2087. PMLR, 2018. 6

[13] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017. 5

[14] Max Jaderberg, Karen Simonyan, Andrew Zisserman, and koray kavukcuoglu. Spatial transformer networks. In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015. 4

[15] Priyank Jaini, Kira A Selby, and Yaoliang Yu. Sum-of-squares polynomial flow. In *International Conference on Machine Learning*, pages 3009–3018. PMLR, 2019. 3

[16] Lai Jiang, Yifei Li, Shengxi Li, Mai Xu, Se Lei, Yichen Guo, and Bo Huang. Does text attract attention on e-commerce images: A novel saliency prediction dataset and method. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2088–2097, 2022. 4

[17] Heewon Kim, Myungsub Choi, Bee Lim, and Kyoung Mu Lee. Task-aware image downscaling. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 399–414, 2018. 1, 2, 3, 6, 7

[18] Hee-Jae Kim, Je-Won Kang, and Byung-Uk Lee. Super-resolution of multi-view erp 360-degree images with two-stage disparity refinement. In *2020 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pages 1283–1286. IEEE, 2020. 2

[19] Durk P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. *Advances in neural information processing systems*, 31, 2018. 3, 6

[20] Yuyan Li, Yuliang Guo, Zhixin Yan, Xinyu Huang, Ye Duan, and Liu Ren. Omnifusion: 360 monocular depth estimation via geometry-aware fusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2801–2810, 2022. 2

[21] Yue Li, Dong Liu, Houqiang Li, Li Li, Zhu Li, and Feng Wu. Learning a convolutional neural network for image compact-resolution. *IEEE Transactions on Image Processing*, 28(3):1092–1107, 2018. 1, 2

[22] Jingyun Liang, Jiezhang Cao, Guolei Sun, Kai Zhang, Luc Van Gool, and Radu Timofte. Swinir: Image restoration using swin transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1833–1844, 2021. 4

[23] Jingyun Liang, Andreas Lugmayr, Kai Zhang, Martin Danelljan, Luc Van Gool, and Radu Timofte. Hierarchical conditional flow: A unified framework for image super-resolution and image rescaling. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4076–4085, 2021. 1, 2, 3, 6, 7

[24] 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. 2, 6, 7

[25] Hongying Liu, Zhubo Ruan, Chaowei Fang, Peng Zhao, Fanhua Shang, Yuanyuan Liu, and Lijun Wang. A single frame and multi-frame joint network for 360-degree panorama video super-resolution. *arXiv preprint arXiv:2008.10320*, 2020. 2, 6

[26] Ze Liu, Han Hu, Yutong Lin, Zhuliang Yao, Zhenda Xie, Yixuan Wei, Jia Ning, Yue Cao, Zheng Zhang, Li Dong, et al. Swin transformer v2: Scaling up capacity and resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12009–12019, 2022. 4

[27] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer:

Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021. 4

[28] Don P Mitchell and Arun N Netravali. Reconstruction filters in computer-graphics. *ACM Siggraph Computer Graphics*, 22(4):221–228, 1988. 3

[29] Akito Nishiyama, Satoshi Ikehata, and Kiyoharu Aizawa. 360 single image super resolution via distortion-aware network and distorted perspective images. In *2021 IEEE International Conference on Image Processing (ICIP)*, pages 1829–1833. IEEE, 2021. 2, 5, 6, 7

[30] Cagri Ozcinar, Aakanksha Rana, and Aljosa Smolic. Super-resolution of omnidirectional images using adversarial learning. In *2019 IEEE 21st International Workshop on Multimedia Signal Processing (MMSP)*, pages 1–6. IEEE, 2019. 2, 6, 7

[31] Zhihong Pan, Baopu Li, Dongliang He, Mingde Yao, Wenhao Wu, Tianwei Lin, Xin Li, and Errui Ding. Towards bidirectional arbitrary image rescaling: Joint optimization and cycle idempotence. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17389–17398, 2022. 2

[32] George Papamakarios, Theo Pavlakou, and Iain Murray. Masked autoregressive flow for density estimation. *Advances in neural information processing systems*, 30, 2017. 6

[33] Claude E Shannon. Communication in the presence of noise. *Proceedings of the IRE*, 37(1):10–21, 1949. 3

[34] Zhijie Shen, Chunyu Lin, Kang Liao, Lang Nie, Zishuo Zheng, and Yao Zhao. Panoformer: Panorama transformer for indoor 360 depth estimation. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part I*, pages 195–211. Springer, 2022. 2

[35] Zhijie Shen, Chunyu Lin, Lang Nie, Kang Liao, and Yao Zhao. Neural contourlet network for monocular 360 depth estimation. *IEEE Transactions on Circuits and Systems for Video Technology*, 32(12):8574–8585, 2022. 2

[36] Wanjie Sun and Zhenzhong Chen. Learned image downscaling for upscaling using content adaptive resampler. *IEEE Transactions on Image Processing*, 29:4027–4040, 2020. 1, 2, 3, 6, 7

[37] Y Sun, A Lu, and L Yu. Ahg8: Ws-psnr for 360 video objective quality evaluation. In *Joint Video Exploration Team of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, JVET-D0040, 4th Meeting*, 2016. 5

[38] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 4

[39] Xintao Wang, Ke Yu, Shixiang Wu, Jinjin Gu, Yihao Liu, Chao Dong, Yu Qiao, and Chen Change Loy. Esrgan: Enhanced super-resolution generative adversarial networks. In *Proceedings of the European conference on computer vision (ECCV) workshops*, pages 0–0, 2018. 5

[40] MJ Willis. Proportional-integral-derivative control. *Dept. of Chemical and Process Engineering University of Newcastle*, 1999. 6

[41] Zhuofan Xia, Xuran Pan, Shiji Song, Li Erran Li, and Gao Huang. Vision transformer with deformable attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4794–4803, 2022. 4

[42] Jianxiong Xiao, Krista A Ehinger, Aude Oliva, and Antonio Torralba. Recognizing scene viewpoint using panoramic place representation. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2695–2702. IEEE, 2012. 2, 6, 7

[43] Mingqing Xiao, Shuxin Zheng, Chang Liu, Zhouchen Lin, and Tie-Yan Liu. Invertible rescaling network and its extensions. *International Journal of Computer Vision*, pages 1–26, 2022. 1, 2, 6, 7

[44] Mingqing Xiao, Shuxin Zheng, Chang Liu, Yaolong Wang, Di He, Guolin Ke, Jiang Bian, Zhouchen Lin, and Tie-Yan Liu. Invertible image rescaling. In *European Conference on Computer Vision*, pages 126–144. Springer, 2020. 1, 2

[45] Zhenda Xie, Yutong Lin, Zhuliang Yao, Zheng Zhang, Qi Dai, Yue Cao, and Han Hu. Self-supervised learning with swin transformers. *arXiv preprint arXiv:2105.04553*, 2021. 4

[46] Yi Zhang, Lu Zhang, Wassim Hamidouche, and Olivier Deforges. A fixation-based 360 benchmark dataset for salient object detection. In *2020 IEEE International Conference on Image Processing (ICIP)*, pages 3458–3462. IEEE, 2020. 2, 6, 7