

# ResVR: Joint Rescaling and Viewport Rendering of Omnidirectional Images

Weiqi Li<sup>\*1,2</sup>, Shijie Zhao<sup>†2</sup>, Bin Chen<sup>1</sup>, Xinhua Cheng<sup>1</sup>, Junlin Li<sup>2</sup>, Li Zhang<sup>2</sup>, Jian Zhang<sup>‡1</sup>

<sup>1</sup>Peking University    <sup>2</sup>ByteDance Inc  
liweiqi@stu.pku.edu.cn

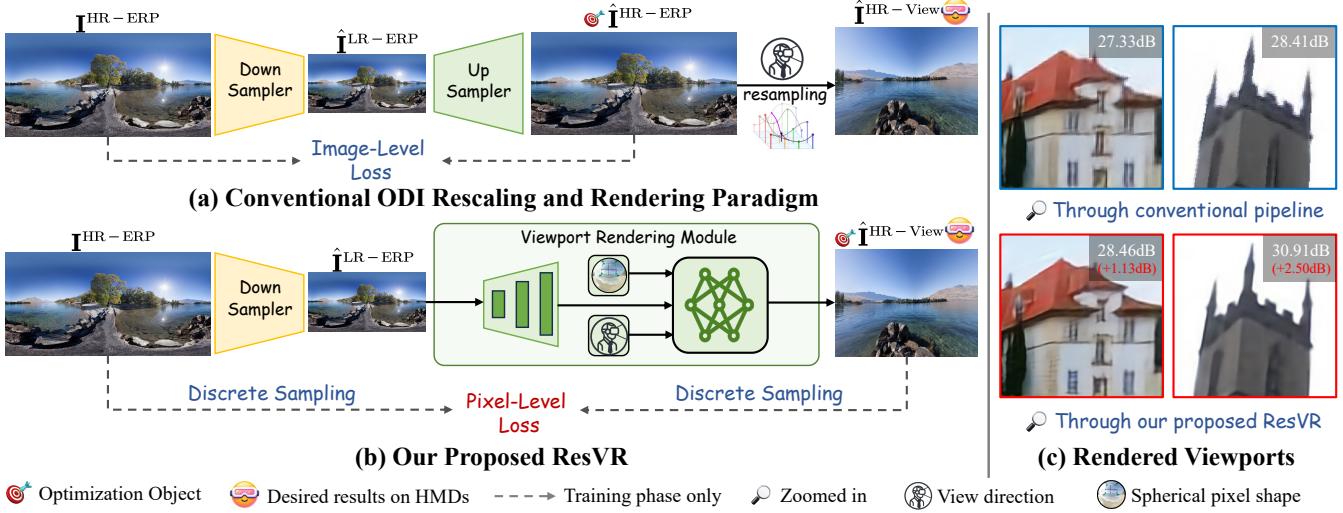


Figure 1: Proposed ResVR compared to previous ODI rescaling and viewport rendering paradigms. (a) Conventional methods focus on improving the quality of rescaled ERP images, resulting in inferior visual experiences. (b) Considering the fact that the desired content viewed on HMDs is a rendered viewport instead of an ERP image, our ResVR directly optimizes the quality of the final viewport for users through a novel discrete pixel sampling strategy and a spherical pixel shape representation technique. (c) Visual and PSNR comparisons of rendered viewports between the two pipelines in (a) and (b) on user HMDs.

## ABSTRACT

With the advent of virtual reality technology, omnidirectional image (ODI) rescaling techniques are increasingly embraced for reducing transmitted and stored file sizes while preserving high image quality. Despite this progress, current ODI rescaling methods predominantly focus on enhancing the quality of images in equirectangular projection (ERP) format, which overlooks the fact that the content viewed on head mounted displays (HMDs) is actually a rendered viewport instead of an ERP image. In this work, we emphasize that focusing solely on ERP quality results in inferior viewport visual experiences for users. Thus, we propose **ResVR**, which is the first comprehensive framework for the joint Rescaling and Viewport Rendering of ODIs. ResVR allows obtaining LR ERP images for transmission while rendering high-quality viewports for users to watch on HMDs. In our ResVR, a novel discrete pixel sampling strategy is developed to tackle the complex mapping between the viewport and ERP, enabling end-to-end training of ResVR pipeline. Furthermore, a spherical pixel shape representation technique is innovatively derived from spherical differentiation to significantly improve the visual quality of rendered viewports. Extensive experiments demonstrate that our ResVR outperforms existing methods in

viewport rendering tasks across different fields of view, resolutions, and view directions while keeping a low transmission overhead<sup>1</sup>.

## CCS CONCEPTS

- Computing methodologies → Computer vision.

## KEYWORDS

Omnidirectional Image, Image Rescaling, Viewport Rendering

## 1 INTRODUCTION

With the growing interest in virtual reality and augmented reality, omnidirectional images (ODIs), also referred to as 360° or panoramic images, attract great attention within the computer vision community for their immersive and interactive capabilities. Although ODIs can capture scenes across a comprehensive 360°×180° views, head-mounted displays (HMDs) often present a limited field-of-view (FoV), necessitating resolutions as high as 4K×8K [2] to preserve details in a small viewport. High-resolution (HR) ODIs are typically stored on cloud servers by platforms of virtual reality media, requiring real-time download by users. This can degrade the visual experience of users, particularly under poor internet conditions.

<sup>1</sup>The complete code and pre-trained models of our method will be made available.

\* Interns in MMLab, ByteDance.

† Corresponding author.

Image rescaling [13, 29, 51–53, 55] emerges as an effective method to reduce image file size for storage and transmission while preserving high quality in the reconstructed images at the user end. This technique first downscales HR images to low-resolution (LR) ones that keep the most important visual details, then upscales them back to their original HR versions. It not only minimizes the file size of LR images but also maintains the quality of reconstructed HR images [39], thus becoming a straightforward approach for efficient transmission and storage for ODIs from cloud servers to user HMDs. However, the prevalent storage and transmission format for ODIs, i.e., the equirectangular projection (ERP), has directed state-of-the-art ODI rescaling works [17] to focus on enhancing the quality of ERP images. As depicted in Fig. 1 (a), after receiving an LR ODI, the typical process on HMDs is two-step, involving (1) upscaling HR ODIs in ERP format from LR images ( $\hat{I}^{LR-ERP} \mapsto \hat{I}^{HR-ERP}$ ), and (2) projecting them onto the viewport ( $\hat{I}^{HR-ERP} \mapsto \hat{I}^{HR-View}$ ) using traditional interpolation methods such as bilinear. This pipeline does not fully account for the ultimate rendered image of the HMD viewport  $\hat{I}^{HR-View}$ , particularly lacking optimization for the final viewing experience. As Fig. 1 (c) shows, the quality of images seen by the user on HMDs can be significantly lower than anticipated.

In this paper, we point out that (1) the content viewed on HMDs is actually a rendered viewport, not an ERP image, and (2) focusing solely on the quality of ERP images will result in sub-optimal viewport visual experiences. To improve the viewing experience for users, there is a need to develop a comprehensive solution that is optimized for end-to-end ODI processing from the storage of ERP images to the display of the final viewport. To this end, we propose **ResVR**, a novel framework for joint Rescaling and Viewport Rendering of ODIs, marking an innovative step towards comprehensive end-to-end ODI processing. As shown in Fig. 1 (b), ResVR aligns the optimization of network parameters with our primary goal of improving the quality of the final viewport. By utilizing such a new methodology, the viewports rendered through our ResVR framework exhibit enhanced details and fewer artifacts compared to those produced by conventional pipelines, as shown in Fig. 1 (c).

In our proposed ResVR, HR ODIs are firstly embedded into LR images to facilitate efficient transmission. Then, HR viewports are directly rendered from LR ERP images on HMDs. This process ( $\hat{I}^{LR-ERP} \mapsto \hat{I}^{HR-View}$ ) does not need to produce HR ERP images. To deal with the irregular correspondence between the ERP area and the viewport, which hinders the joint optimization of downscaling and viewport rendering using traditional image-level loss training methods, we develop a discrete pixel sampling strategy. In each training iteration, this strategy is used to randomly sample paired sets of ground truth pixels and the reconstructed ones on viewports, thus making the end-to-end learning of our entire ResVR pipeline feasible in implementation. Furthermore, to enhance our ResVR’s awareness of the positions of different viewport pixels on the spherical surface, we introduce a technique for spherical pixel shape representation. This technique employs spherical differentiation to calculate the geometric orientation and curvature of various viewport areas, offering positional information that contains more precise spherical attributes than existing 2D image representation methods [23]. This advanced representation effectively improves

the quality of the final viewport, especially in regions of high latitude and longitude. Extensive experiments on various panoramic image datasets show that ResVR outperforms existing methods in multiple viewport rendering tasks across different FoVs, view directions, and resolutions. In summary, our contributions are:

- We propose ResVR, a novel framework for the comprehensive processing of omnidirectional images, seamlessly integrates image **R**escaling with **V**iewport **R**endering. Our ResVR effectively balances the transmission efficiency and users’ visual experience.

- We develop a discrete pixel sampling strategy to tackle the complex correspondence between viewport and equirectangular projection (ERP) areas within our framework. This strategy makes the end-to-end training of our whole processing pipeline feasible.

- We introduce a spherical pixel shape representation technique based on spherical differentiation to guide viewport rendering, which significantly enhances the visual quality of the final viewport.

- Extensive empirical evaluations on various panoramic image datasets exhibit that ResVR consistently achieves new state-of-the-art visual quality while maintaining a low transmission bitrate.

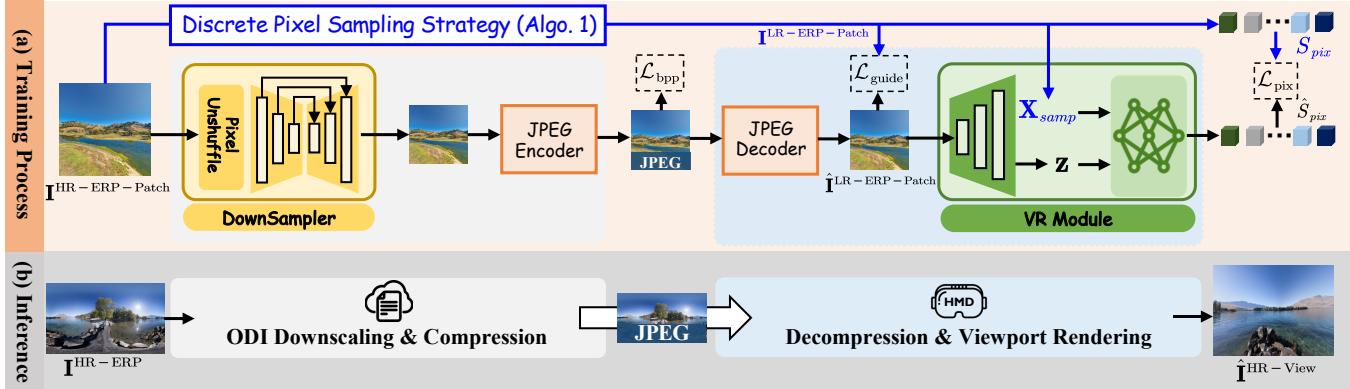
## 2 RELATED WORK

### 2.1 Omnidirectional Image Super-Resolution

Image super-resolution (SR) seeks to construct HR images from LR ones. Since the advent of deep neural networks (DNNs) in SR-CNN [16], subsequent studies [9, 11, 14, 27, 28, 30, 31, 33, 46, 48, 62, 67–69] have significantly advanced SR performance beyond traditional methods. In the specific context of omnidirectional image super-resolution (ODISR), DNN-based approaches have been tailored to account for the unique latitude-based characteristics of ODIs [1, 5–7, 15, 26, 32, 35, 37, 43, 63, 64]. For example, LAU-Net [15] divides the entire ERP image into latitude-based patches for separate upscaling. SphereSR [63] introduces the spherical local implicit image function (SLIIF) alongside a novel feature extraction module to leverage information from arbitrary projection types. OSRT [64] employs a distortion-aware transformer targeting dimension-related distortions in ERP images. OPDN [43] introduces a dual-stage framework incorporating a position-aware deformable network. Despite these noteworthy advancements, the majority of these methodologies presuppose a fixed downscaling approach (e.g. bicubic [34]) and overlook high-frequency components from HR inputs, thus limiting the quality of reconstructed details in SR ODIs.

### 2.2 Image Rescaling

Different from SR, image rescaling focuses on downscaling HR images to create visually pleasing LR images that retain essential information for accurate HR reconstruction. Recently, invertible neural network (INN) [4, 10, 36, 66] becomes a representative framework for image rescaling [13, 29, 38, 42, 51, 52, 56, 59], offering a direct route to inversely map the downsampled images back to HR. For instance, IRN [51, 52] is the first attempt to model image downscaling and upscaling using invertible transmissions. Liang *et al.* [29] formulate high-frequency components in INNs as a conditional distribution on the LR image. HyperThumbnail [39] employs an asymmetric encoder-decoder architecture for real-time reconstruction of 6K images and also optimizes the JPEG compression process [45, 49, 54]. Very recently, DINN [17] makes the first attempt



**Figure 2: Overview of our proposed ResVR framework.** The comprehensive ODI processing of ResVR contains two sequential steps: (1) ODI Downscaling & Compression and (2) Decompression & Viewport Rendering. (a) In the training process, HR ERP patches  $I^{HR-ERP-Patch}$  are randomly sampled through our proposed discrete pixel sampling strategy (Algo. 1) to generate the guided LR patches  $I^{LR-ERP-Patch}$ , query coordinates  $X_{samp}$  and the set of ground truth pixels  $S_{pix}$ . This strategy innovatively makes the end-to-end training of ResVR feasible in implementation. (b) During inference, our trained ResVR model can be directly applied for joined rescaling and viewpoint rendering of given HR ERP images from the cloud server to user HMDs.

to apply image rescaling to ODI and highlights the significance of leveraging ERP’s latitude characteristics by developing a latitude-aware conditional mechanism. However, existing ODI rescaling methods focus solely on improving the quality of ERP images. In contrast, our ResVR innovatively optimizes the quality of the final viewport, offering new improvements orthogonal to previous ODI rescaling methodologies.

### 2.3 Viewport Rendering of ODIs

ODIs are designed to encapsulate a full spherical view, enabling an immersive viewing experience. However, when viewed through HMDs towards a particular direction, only a specific viewport is displayed [19, 20]. Achieving high resolution and quality in these viewports is crucial for immersive experiences, as highlighted by various ODI visual quality assessment techniques [57, 58, 60, 61]. As a widely adopted method for viewport rendering [8], perspective projection employs a series of pixel mapping and resampling operations to effectively implement image warping. To reduce the interpolation-induced blurriness and artifacts, SRWarp [41] reinterprets image warping as an SR problem and introduces a differentiable warping module. LTEW [23] uses a continuous neural representation [12, 21, 24] for image warping by taking advantage of both Fourier features and spatially-varying Jacobian matrices. LeRF [25] assigns spatially varying steerable resampling functions to pixels, learning their orientations for continuous function prediction. Different from the above methods that focus on warping, our ResVR considers comprehensive ODI processing, serving as a new framework while enjoying high viewpoint rendering quality.

## 3 METHODOLOGY

In this section, we begin with a concise review of the viewport rendering process (Sec. 3.1). Following this, we provide an overview of our ResVR framework (Sec. 3.2), which is illustrated in Fig. 2. We then elaborate on the proposed discrete pixel sampling strategy (Sec. 3.3) and spherical pixel shape representation technique (Sec. 3.4). The training objectives are detailed in Sec. 3.5.

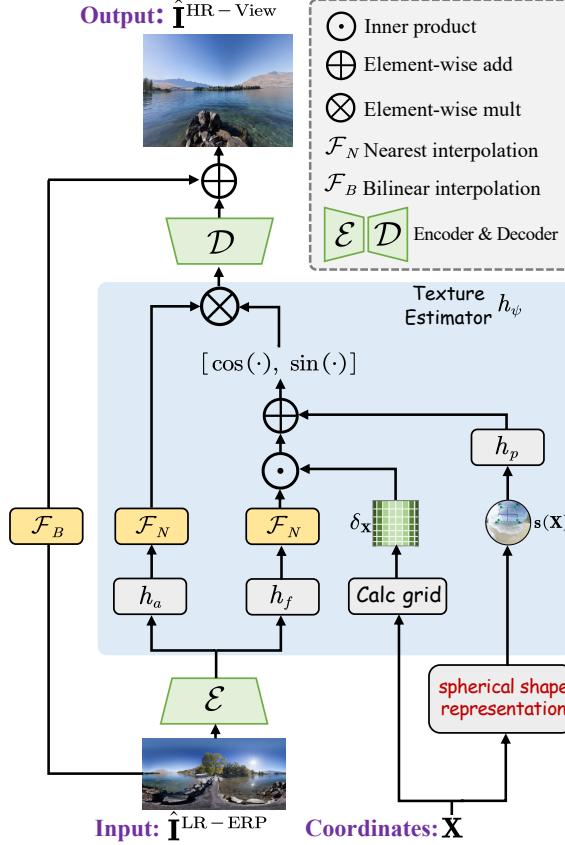
### 3.1 Preliminaries of Viewport Rendering

ODIs inherently provide a full spherical view. However, when viewed through HMDs directed towards a specific direction, only the corresponding viewport is displayed. This viewport appears as a 2D image, derived through perspective projection [22] from a segment of the spherical image. To formalize this process, we consider the view direction in spherical coordinates  $(\theta_c, \phi_c)$ , along with the horizontal and vertical fields of view  $(F_h, F_v)$ , and the height and width of the viewport  $(h_v, w_v)$ . An invertible coordinate mapping  $f : X \mapsto Y$  is established, where  $X := \{x | x \in \mathbb{R}^2\}$  denotes the coordinate space of ERP, and  $Y := \{y | y \in \mathbb{R}^2\}$  represents the coordinate space of the viewport to be rendered. In practice, the coordinates  $Y_{view}$  on the viewport are initially determined by  $(h_v, w_v)$ , and then the corresponding coordinates on ERP are obtained through backward mapping  $X_{view} = f^{-1}(Y_{view})$ . The rendering of the viewport is achieved through resampling techniques, such as interpolation, ensuring that this process remains fully differentiable. Additional mathematical details of perspective projection and viewport rendering are elaborated in the supplementary material.

### 3.2 Overview of ResVR

An overview of proposed joint Rescaling and Viewport Rendering (**ResVR**) of ODIs is presented in Fig. 2. The ODI processing of ResVR contains two steps: (1) ODI Downscaling & Compression: An HR ERP image is firstly downsampled and compressed to an LR ERP JPEG image for efficient transmission from cloud server to user HMDs, and (2) Decompression & Viewport Rendering: The LR ERP image is then decompressed and rendered to HR viewports on HMDs through our viewport rendering (VR) module.

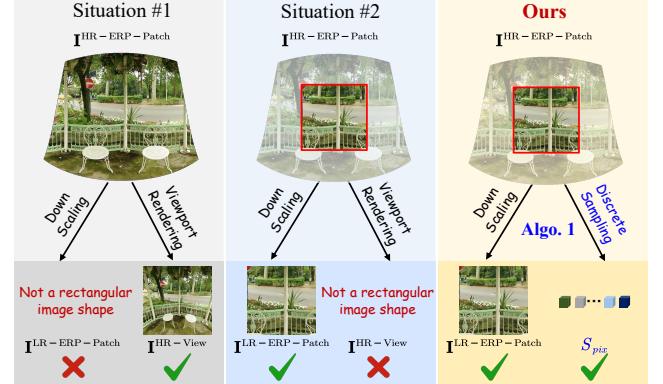
**ODI Downscaling & Compression:** Given an HR ERP image  $I^{HR-ERP} \in \mathbb{R}^{3 \times H \times W}$ , its LR representation is firstly generated through our downsampler, where  $s$  is the rescaling factor. The downsampler is a U-Net [40] with dense blocks [18]. To further decrease the size of the transmitted image file, a JPEG encoder is employed for the LR representation to obtain a JPEG image. Concretely, we follow [39] to predict adaptive quantization tables for each image.



**Figure 3: Illustration of the VR module, which consists of an encoder  $\mathcal{E}$ , a local texture estimator  $h_\psi$ , and an MLP decoder  $\mathcal{D}$ . Given query coordinates  $\mathbf{X}$ , it directly predicts  $\hat{\mathbf{I}}^{\text{HR-View}}$  from  $\hat{\mathbf{I}}^{\text{LR-ERP}}$  without the need to produce an HR ERP image.**

Finally, the LR JPEG image is obtained, and adaptive quantization tables and quantized DCT coefficients are also encoded into the JPEG file. More details about the downsample, the adaptive quantization table prediction module, and the training process of learned compression are provided in the supplementary material.

**Decompression & Viewport Rendering:** After receiving the LR JPEG image,  $\hat{\mathbf{I}}^{\text{LR-ERP}} \in \mathbb{R}^{3 \times \frac{H}{s} \times \frac{W}{s}}$  is firstly reconstructed by the JPEG decoder through inverse discrete cosine transformation (IDCT). Then our goal is to render high-resolution viewport  $\hat{\mathbf{I}}^{\text{HR-View}} \in \mathbb{R}^{3 \times h_v \times w_v}$  directly from  $\hat{\mathbf{I}}^{\text{LR-ERP}}$ . Inspired by recent implicit neural representation methods [12, 23, 24], we develop a viewport rendering (VR) module to predict pixel values of the query coordinates  $\mathbf{X}_{\text{view}}$  from the latent space of input  $\hat{\mathbf{I}}^{\text{LR-ERP}}$  instead of using traditional interpolation methods. As depicted in Fig. 3, our VR module consists of an encoder  $\mathcal{E}$ , a local texture estimator  $h_\psi = \{h_a, h_f, h_p\}$ , and an MLP decoder  $\mathcal{D}$ .  $h_\psi$  is a learnable dominant-frequency estimator, which is capable of characterizing image textures in 2D Fourier space [24]. Here,  $h_a$  is an amplitude estimator ( $\mathbb{R}^C \mapsto \mathbb{R}^{256}$ ),  $h_f$  is a frequency estimator ( $\mathbb{R}^C \mapsto \mathbb{R}^{2 \times 128}$ ), and  $h_p$  is a phase estimator ( $\mathbb{R}^{10} \mapsto \mathbb{R}^{128}$ ). Concretely, for a query point  $\mathbf{x} \in \mathbf{X}_{\text{view}}$ , the estimating function  $h_\psi$  is defined as:



**Figure 4: Training ResVR end-to-end faces challenges due to the mismatch in shapes between the ERP image patch ( $\mathbf{I}^{\text{HR-ERP-Patch}}$ ) and the viewport ( $\mathbf{I}^{\text{HR-View}}$ ). In Situation #1, although we obtain  $\mathbf{I}^{\text{HR-View}}$  with a rectangular image shape, its corresponding  $\mathbf{I}^{\text{LR-ERP-Patch}}$  does not have a rectangular image shape, preventing its use in supervising the downscaling process. Situation #2 experiences the opposite issue. Both two situations are impractical for training. In contrast, our method utilizes a novel discrete pixel sampling strategy (Algo. 1) to make end-to-end training feasible.**

$$h_\psi(\mathbf{z}_j, \delta_{\mathbf{x}}, \mathbf{s}(\mathbf{x})) = h_a(\mathbf{z}_j) \otimes \begin{bmatrix} \cos\{\pi(\langle h_f(\mathbf{z}_j), \delta_{\mathbf{x}} \rangle + h_p(\mathbf{s}(\mathbf{x})))\} \\ \sin\{\pi(\langle h_f(\mathbf{z}_j), \delta_{\mathbf{x}} \rangle + h_p(\mathbf{s}(\mathbf{x})))\} \end{bmatrix}, \quad (1)$$

where  $\mathbf{z} = \mathcal{E}(\hat{\mathbf{I}}^{\text{LR-ERP}})$ . Denote  $j$  as a pixel index of the  $\hat{\mathbf{I}}^{\text{LR-ERP}}$ ,  $\mathbf{x}_j$  and  $\mathbf{z}_j$  are the corresponding coordinates and latent variable of  $\hat{\mathbf{I}}^{\text{LR-ERP}}$ , respectively.  $\delta_{\mathbf{x}}$  is a local grid calculated by  $\delta_{\mathbf{x}} = \mathbf{x} - \mathbf{x}_j$ .  $\langle \cdot, \cdot \rangle$  is an inner product, and  $\otimes$  denotes element-wise multiplication. To be noted,  $\mathbf{s}(\mathbf{x})$  is the spherical pixel shape representation of the query coordinate  $\mathbf{x}$ , which is important for providing spatial-varying priors [23] for our VR module and will be elaborated on in Sec. 3.4. Finally, our VR module predicts the RGB values of a coordinate  $\mathbf{y} = f(\mathbf{x})$  on the viewport as:

$$\hat{\mathbf{I}}^{\text{HR-View}}[\mathbf{y}] = \mathcal{F}_B(\hat{\mathbf{I}}^{\text{LR-ERP}}) + \sum_{j \in \mathcal{J}} \omega_j \mathcal{D}(h_\psi(\mathbf{z}_j, \delta_{\mathbf{x}}, \mathbf{s}(\mathbf{x}))), \quad (2)$$

where  $\mathcal{F}_B$  is a bilinear interpolation operator to stabilize the convergence of network training and aid the VR module in learning high-frequency details.  $\mathcal{J}$  is a neighborhood set of  $\mathbf{x}$ , defined as  $\mathcal{J} = \{j | j = \mathbf{x} + [\frac{m_s}{W}, \frac{n_s}{H}], m, n \in \{-1, 1\}\}$ , while  $\omega_j$  is a local ensemble coefficient. More details of our VR module are provided in the supplementary material. Different from existing ODI rescaling methods which first produce  $\hat{\mathbf{I}}^{\text{HR-ERP}}$  and then obtain HR viewports through traditional interpolation methods, our ResVR directly renders the final HR viewports from  $\hat{\mathbf{I}}^{\text{LR-ERP}}$  through our VR module.

### 3.3 Discrete Pixel Sampling Strategy

**Motivation.** In image rescaling and SR tasks, the ground truth HR images  $\mathbf{I}^{\text{HR-ERP}}$  are typically cropped into patches  $\mathbf{I}^{\text{HR-ERP-Patch}}$  to reduce GPU memory overhead while increasing the diversity of training data. The goals of our ResVR's end-to-end training process include: (1) optimizing the downscaling process ( $\mathbf{I}^{\text{HR-ERP-Patch}} \mapsto$

**Algorithm 1** Discrete pixel sampling strategy

---

$\mathbf{I}^{\text{HR-ERP}}$ ,  $s$ : HR ERP image and the downscaling factor  
 $(a, b)$ ,  $p$ : left top coordinate and the size of cropped patch  
 $(\theta_c, \phi_c)$ ,  $(F_h, F_v)$ : view direction and FoVs of the viewport  
 $(h_v, w_v)$ ,  $\mathbf{Y}_{\text{view}}$ : shape and the coordinate space of the viewport

```

function DISAMP( $a, b, p, \theta_c, \phi_c, F_h, F_v, h_v, w_v, \mathbf{Y}_{\text{view}}, \mathbf{I}^{\text{HR-ERP}}$ )
     $\mathbf{I}^{\text{HR-ERP-Patch}} \leftarrow \text{CropPatch}(\mathbf{I}^{\text{HR-ERP}}, a, b, p)$ 
     $f \leftarrow \text{GetTransform}(\theta_c, \phi_c, F_h, F_v, h_v, w_v)$ 
     $\mathbf{X}_{\text{view}} \leftarrow f^{-1}(\mathbf{Y}_{\text{view}})$   $\triangleright$  Inverse mapping from viewport to ERP
     $\mathbf{X}'_{\text{view}} \leftarrow \text{FilterWithBounds}(\mathbf{X}_{\text{view}}, a, b, p)$   $\triangleright$  Eq. (3)
    if  $|\mathbf{X}'_{\text{view}}| > N$  then
         $\mathbf{X}'_{\text{view}} \leftarrow \text{RandomSample}(\mathbf{X}'_{\text{view}}, N)$ 
    end if
     $\mathbf{X}_{\text{samp}} \leftarrow \text{CoordSpaceTransform}(\mathbf{X}'_{\text{view}}, a, b, p)$   $\triangleright$  Eq. (4)
     $S_{\text{pix}} \leftarrow \text{BicubicSample}(\mathbf{I}^{\text{HR-ERP-Patch}}, \mathbf{X}_{\text{samp}})$ 
     $\mathbf{I}^{\text{LR-ERP-Patch}} \leftarrow \text{BicubicDownscale}(\mathbf{I}^{\text{HR-ERP-Patch}}, s)$ 
    return  $\mathbf{X}_{\text{samp}}, S_{\text{pix}}, \mathbf{I}^{\text{LR-ERP-Patch}}$ 
end function

```

---

$\mathbf{i}^{\text{LR-ERP-Patch}}$ ), and (2) optimizing the rendering process ( $\mathbf{i}^{\text{LR-ERP-Patch}} \mapsto \mathbf{i}^{\text{HR-View}}$ ). In this framework, the former process requires the corresponding  $\mathbf{I}^{\text{LR-ERP-Patch}}$  as supervision, while the latter process needs the corresponding  $\mathbf{I}^{\text{HR-View}}$  as supervision. However, due to the natural geometric properties of ODIs' different projection types, there exists a challenge of shape mismatch between a viewport and its corresponding ERP area. As depicted in Situations #1 and #2 in Fig. 4, one can not obtain  $\mathbf{I}^{\text{LR-ERP-Patch}}$  and  $\mathbf{I}^{\text{HR-View}}$  that are both with rectangular image shape at the same time. As a result, conventional image-level loss can not be directly used for end-to-end training of ResVR. To address this, we innovatively propose a discrete pixel sampling strategy (DPS), as shown in “Ours” of Fig. 4.

**Method.** Denoting  $p$  as both the height and width of training patches, our core idea is to keep the LR ERP patch with a rectangular image shape ( $\mathbf{I}^{\text{LR-ERP-Patch}} \in \mathbb{R}^{3 \times \frac{p}{s} \times \frac{p}{s}}$ ), while discretely sampling  $N$  pixels  $S_{\text{pix}} \in \mathbb{R}^{3 \times N}$  with their coordinates  $\mathbf{X}_{\text{samp}} \in \mathbb{R}^{2 \times N}$  in the irregular area. Thanks to the continuous representation ability of implicit neural representation methods [12], our VR module is able to predict corresponding pixel values  $\hat{S}_{\text{pix}}$  given coordinates  $\mathbf{X}_{\text{samp}}$ . Hence, we use the sampled  $S_{\text{pix}}$  as supervision for the predicted  $\hat{S}_{\text{pix}}$ . An overview of our strategy is presented in Algo. 1.

Specifically, given the left top coordinate  $(a, b)$  of the training patch, the view direction  $(\theta_c, \phi_c)$ , FoVs  $(F_h$  and  $F_v)$ , shapes  $(h_v, w_v)$ , and query coordinates  $(\mathbf{Y}_{\text{view}})$  of the desired viewport, we first determine the coordinate mapping  $f$ . Then the corresponding coordinates  $\mathbf{X}_{\text{view}}$  in the ERP space can be obtained through inverse coordinate mapping as  $\mathbf{X}_{\text{view}} = f^{-1}(\mathbf{Y}_{\text{view}})$ . By setting appropriate view direction and FoVs, it is possible to ensure that some elements in  $\mathbf{X}_{\text{view}}$  lie in the area of the cropped ERP patch. Recall that  $p$  is the size of the cropped training patch,  $\mathbf{X}'_{\text{view}}$  is then filtered to preserve the subset overlapped with the patch as follows:

$$\mathbf{X}'_{\text{view}} = \{\mathbf{x} \in \mathbf{X}_{\text{view}} | a \leq \mathbf{x}_1 < (a + p), b \leq \mathbf{x}_2 < (b + p)\}, \quad (3)$$

where  $\mathbf{x}_1$  and  $\mathbf{x}_2$  represent the coordinates of a point  $\mathbf{x}$  in the ERP space. This filter operation ensures that every point in  $\mathbf{X}'_{\text{view}}$  can find its corresponding latent variable extracted from  $\mathbf{i}^{\text{LR-ERP-Patch}}$ .

To balance the number of pixels sampled from different FoVs and view directions, if the filtered  $\mathbf{X}'_{\text{view}}$  contains more than  $N$  pixel coordinates, only  $N$  elements will be randomly retained. Finally, to uniform the coordinate correspondence between LR and HR patches,  $\mathbf{X}'_{\text{view}} \subseteq [0, H] \times [0, W]$  is converted from the whole HR ERP coordinate space to the patch coordinate space, and normalized to  $\mathbf{X}_{\text{samp}} \subseteq [-1, 1] \times [-1, 1]$ , as:

$$\mathbf{X}_{\text{samp}} = \{T(\mathbf{x}) | \mathbf{x} \in \mathbf{X}'_{\text{view}}\}, \quad (4)$$

where  $T(\mathbf{x}) = 2(((\mathbf{x}_1, \mathbf{x}_2) - (a, b)) / p) - 1$ .

As a result,  $S_{\text{pix}}$  are sampled through bicubic interpolation as  $S_{\text{pix}} = \mathcal{F}_{\text{bicubic}}(\mathbf{I}^{\text{HR-ERP-Patch}}, \mathbf{X}_{\text{samp}})$ . By adopting Algo. 1,  $\mathbf{I}^{\text{LR-ERP-Patch}} \in \mathbb{R}^{3 \times \frac{p}{s} \times \frac{p}{s}}$  can be used as supervision of the predicted  $\hat{\mathbf{I}}^{\text{LR-ERP-Patch}}$ , and  $S_{\text{pix}} \in \mathbb{R}^{3 \times N}$  can be used as supervision of the predicted  $\hat{S}_{\text{pix}}$ .

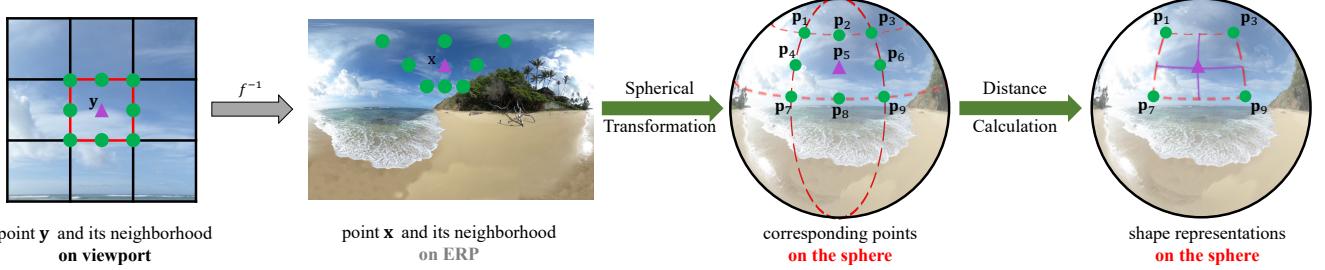
### 3.4 Spherical Pixel Shape Representation

**Motivation.** Pixel shape representations described by grid orientation and curvature provide informative geometric spatial-varying priors for image warping tasks [23]. Existing shape representation methods are designed for 2D images. However, different from 2D images, ERP is an unfolding of the spherical surface along meridians, which leads to the fact that the adjacent pixels on the sphere can be far apart in the ERP image. As a result, the previous 2D shape representation methods fall in the high-latitude/-longitude areas due to the nature properties of (1) latitude-related distortion and (2) wraparound consistency of ERP images (Fig. 7 in Sec. 4.3). Therefore, we develop a spherical pixel shape representation (SSR) technique to solve this challenge. As Fig. 5 shows, SSR leverages the information of transformed coordinates on the sphere as a more effective shape representation to guide the viewport rendering process.

**Method.** In image warping tasks, the first-order partial derivatives (i.e. Jacobian matrix) and the second-order partial derivatives (i.e. Hessian matrix), describe the orientation and curvature of pixels resulting from the transformation, respectively [23]. These two matrices provide informative geometric spatial-varying priors to guide the process of viewport rendering. Inspired by this, for a point  $\mathbf{x}$  on the original ERP plane, we represent its pixel shape  $s(\mathbf{x})$  with the gradient of the inverse coordinate transformation  $f^{-1}$ . Specifically, we start with the point  $\mathbf{y} = f(\mathbf{x})$  on the viewport plane, and the corresponding Jacobian matrix  $\tilde{\mathbf{J}}_{f^{-1}}(\mathbf{y})$  and Hessian matrix  $\tilde{\mathbf{H}}_{f^{-1}}(\mathbf{y})$  are analytically computed as:

$$\tilde{\mathbf{J}}_{f^{-1}}(\mathbf{y}) = \begin{bmatrix} \frac{\partial \mathbf{x}_1}{\partial u} & \frac{\partial \mathbf{x}_1}{\partial v} \\ \frac{\partial \mathbf{x}_2}{\partial u} & \frac{\partial \mathbf{x}_2}{\partial v} \end{bmatrix}, \quad \tilde{\mathbf{H}}_{f^{-1}}(\mathbf{y}) = \begin{bmatrix} \frac{\partial^2 \mathbf{x}_1}{\partial u^2} & \frac{\partial^2 \mathbf{x}_1}{\partial u \partial v} \\ \frac{\partial^2 \mathbf{x}_2}{\partial v \partial u} & \frac{\partial^2 \mathbf{x}_2}{\partial v^2} \end{bmatrix}, \quad (5)$$

where  $\mathbf{x} = f^{-1}(\mathbf{y}) = (\mathbf{x}_1, \mathbf{x}_2)$  represents the coordinates in the original ERP plane. We propose a spherical pixel shape representation (SSR) technique to numerically estimate  $\tilde{\mathbf{J}}_{f^{-1}}(\mathbf{y})$  and  $\tilde{\mathbf{H}}_{f^{-1}}(\mathbf{y})$  based on spherical differentiation. Specifically, as depicted in Fig. 5, the inverse coordinate transformation  $f^{-1}$  is firstly applied to  $\mathbf{y}$  and its eight nearest points ( $\mathbf{y} + [\frac{m}{w_v}, \frac{n}{h_v}]$  with  $m, n \in \{-1, 0, 1\}$ ) to get  $\mathbf{x}$  and its neighborhood on the ERP plane. Different from existing 2D shape representation methods [23] which estimate the shape of  $\mathbf{x}$  directly on the ERP image plane, we further transform  $\mathbf{x}$  and its neighborhood to the sphere denoted by the set



**Figure 5: Illustration of our proposed spherical pixel shape representation (SSR) technique.** We illustrate using a point  $y$  on the viewport. The inverse mapping is firstly applied for  $y$  and its eight nearest neighbors to get  $x$  and its neighbors on ERP. Then these points are transformed into sphere coordinates  $\{p_1, p_2, \dots, p_9\}$ , which are used for calculating numerical derivatives to estimate the pixel shape representation  $s(x)$ , according to proposed spherical central difference method in Eqs. (6) and (7).

$\{p_i|p_i = (\theta_i, \phi_i), i = 1, 2, \dots, 9\}$ , and innovatively calculate numerical derivatives to estimate  $\tilde{J}_{f^{-1}}(y)$  and  $\tilde{H}_{f^{-1}}(y)$  by the proposed spherical central difference method, overriding Eq. (5) as:

$$\begin{aligned}\tilde{J}_{f^{-1}}(y) &\approx \begin{bmatrix} D(p_6, p_4) \\ D(p_2, p_8) \end{bmatrix}, \\ \tilde{H}_{f^{-1}}(y) &\approx \begin{bmatrix} D(p_6, p_5) + D(p_5, p_4) & D(p_3, p_1) + D(p_9, p_7) \\ D(p_3, p_1) + D(p_9, p_7) & D(p_2, p_5) + D(p_5, p_8) \end{bmatrix},\end{aligned}\quad (6)$$

where  $D(p_i, p_j)$  calculates the distance between  $p_i$  and  $p_j$  in the spherical coordinate system as:

$$D(p_i, p_j) = \begin{pmatrix} \min(|\phi_j - \phi_i|, 2\pi - |\phi_j - \phi_i|) \\ \min(|\theta_j - \theta_i|, 2\pi - |\theta_j - \theta_i|) \cdot \cos((\theta_i + \theta_j)/2) \end{pmatrix}^\top,\quad (7)$$

where  $|\cdot|$  represents the absolute value operation. The  $\min(\cdot)$  operator ensures the computation of minor arc distances, and the average latitude is utilized to adjust the longitudinal distance to account for the convergence of meridians. We follow [23] to use six elements in  $\tilde{H}_{f^{-1}}(y)$ , and finally  $s(x) \in \mathbb{R}^{10}$  is obtained by concatenating and flattening  $\tilde{J}_{f^{-1}}(y)$  and  $\tilde{H}_{f^{-1}}(y)$ . As a result,  $s(x)$  serves as an auxiliary input, together with the LR latent representation  $z = \mathcal{E}(I^{\text{LR-ERP}})$  to predict final viewports through the VR module.

### 3.5 Training Objectives

Thanks to the discrete pixel sampling strategy (Sec. 3.3) and the fully differentiable pipeline (Fig. 2), the processes of downscaling, compression, and viewport rendering can be jointly optimized end-to-end, which aligns the learning of network parameters with our goals of obtaining high-quality viewport while reducing the transmission overhead. The total loss is a weighted sum of a pixel-level reconstruction loss, an LR guidance loss, and a bitrate loss as:

$$\mathcal{L} = \mathcal{L}_{\text{pix}} + \lambda_1 \mathcal{L}_{\text{guide}} + \lambda_2 \mathcal{L}_{\text{bpp}},\quad (8)$$

where  $\lambda_1$  and  $\lambda_2$  are two trade-off parameters.

**Pixel-level reconstruction and guidance loss.** In the training phase, we employ Algo. 1 to get  $S_{\text{pix}}$  as the ground truth and use the VR module to predict the corresponding values  $\hat{S}_{\text{pix}}$  given  $X_{\text{samp}}$ .

$$\mathcal{L}_{\text{pix}} = \frac{\|\hat{S}_{\text{pix}} - S_{\text{pix}}\|_1}{N},\quad (9)$$

where  $N$  is the number of sampled pixels. Additionally, following [39, 51, 52], an  $L_2$  guidance loss on the LR ERP patch is defined as:

$$\mathcal{L}_{\text{guide}} = \frac{\|\hat{I}^{\text{LR-ERP-Patch}} - I^{\text{LR-ERP-Patch}}\|_2^2}{(p/s) \times (p/s)}.\quad (10)$$

**Bitrate loss.** To optimize the size of the transmitted JPEG image, we firstly follow [3] to estimate the rate  $R$  of the quantized coefficients  $\tilde{C}$  with differentiable fully factorized entropy models as:  $R = \mathbb{E}_{x \sim p_x} [-\log p_L(\tilde{C}_y) - \log p_C(\tilde{C}_{Cb}) - \log p_C(\tilde{C}_{Cr})]$ , where  $p_L$  and  $p_C$  are two fully factorized entropy models for luma and chroma coefficient maps, respectively. Then the bpp of the transmitted LR ERP JPEG image is calculated as:

$$\mathcal{L}_{\text{bpp}} = \frac{R}{H \times W}.\quad (11)$$

## 4 EXPERIMENTS

### 4.1 Experimental Setup

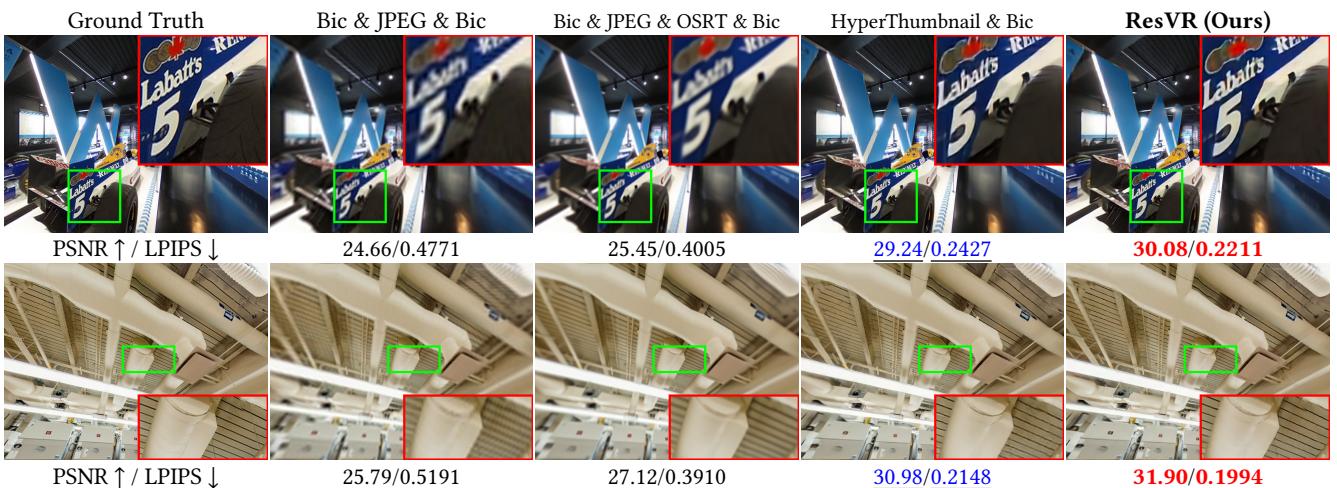
**Implementation details.** We follow the previous work [39] to put more computation into the downampler to keep the VR module lightweight. The VR module is composed of a lightweight feature extractor and a tiny MLP. Please refer to our supplementary materials for more details of our network architecture.

**Training details.** The downscaling factor  $s$  is fixed to 4, and patch size  $p$  of random cropped  $I^{\text{HR-ERP-Patch}}$  is set to 256. To ensure the overlapping of  $I^{\text{HR-ERP-Patch}}$  and discretely sampled points in Algo. 1, we calculate the viewport center  $(\theta_c, \phi_c)$  according to the center of cropped patch. To enable ResVR to handle different resolutions and FoVs, we randomly sample FoVs and resolutions during training. Specifically, the FoVs ( $F_h, F_v$ ) are randomly sampled from  $\{80^\circ, 90^\circ, 100^\circ, 110^\circ, 120^\circ\}$  and the resolutions of viewport ( $h_v, w_v$ ) are randomly sampled from  $\{512, 576, 640, 768, 832, 960, 1024\}$ . The number of sampled pixels  $N$  is set to 25600, and the batch size is set to 16. All experiments are conducted on one V100 GPU. The network is trained for  $5 \times 10^5$  iterations with learning rate  $2 \times 10^{-4}$ .  $\lambda_1$  and  $\lambda_2$  are set to 0.6 and 0.01 for all experiments, respectively.

**Datasets and evaluation metrics.** ODI-SR dataset [15] and SUN360 Panorama dataset [50] are used in our experiment. We follow the data split setting in [15] and train on the ODI-SR training set. For evaluation of transmission efficiency, we follow [39] to use the real file size of JPEG for evaluating the bitrate:  $\text{bpp} = \mathbb{E}_{x \sim p_x} [\text{filesize}/(H \times W)]$ . For evaluation of the quality of rendered viewports, we choose ten different view directions, get the ground

**Table 1: Quantitative evaluation of rendered viewports with  $(F_h, F_v) = (120^\circ, 90^\circ)$  and  $(w_v, h_v) = (2048, 1536)$ . We keep bpp around 0.3 on different datasets. The WS-PSNR is evaluated for methods that need to explicitly get HR ERP to render the viewport. Focusing solely on the quality of ERP images results in the sub-optimal visual experience of final viewports. Throughout this paper, the best and second-best results of each test setting are highlighted in **bold red** and underlined blue, respectively.**

Method Down & Compression & Up & Render	ODI-SR [15]					SUN 360 [50]				
	bpp↓	WS-PSNR	PSNR↑	SSIM↑	LPIPS↓	bpp↓	WS-PSNR	PSNR↑	SSIM↑	LPIPS↓
Bicubic & JPEG & Bicubic	0.29	N/A	27.98	0.7897	0.4815	0.28	N/A	28.06	0.8077	0.4642
Bicubic & JPEG & OSRT [64] & Bicubic	0.29	25.73	28.42	0.7975	0.4395	0.28	26.09	28.84	0.8202	0.5510
HyperThumbnail [39] & Nearest	0.30	27.84	30.13	0.8314	0.4197	0.29	28.97	31.16	0.8601	0.3768
HyperThumbnail [39] & Bilinear	0.30	27.84	30.82	0.8475	0.3397	0.29	28.97	32.20	0.8775	0.2792
HyperThumbnail [39] & Bicubic	0.30	27.84	<u>31.00</u>	<u>0.8515</u>	<u>0.3222</u>	0.29	28.97	<u>32.54</u>	<u>0.8822</u>	<u>0.2610</u>
<b>ResVR (Ours)</b>	0.30	N/A	<b>31.39</b>	<b>0.8568</b>	<b>0.3026</b>	0.29	N/A	<b>32.95</b>	<b>0.8862</b>	<b>0.2462</b>



**Figure 6: Comparisons of two rendered viewports from ODI-SR [15] (“img\_005”, top) and SUN 360 [50] (“img\_046”, bottom), with  $(\theta = 0^\circ, \phi = 90^\circ)$  and  $(\theta = 45^\circ, \phi = 180^\circ)$ , respectively. The viewports are with  $F_h, F_v = (120^\circ, 90^\circ)$  and resolutions  $(w_v, h_v) = (2048, 1536)$ . “Bic” stands for Bicubic interpolation. Please zoom in for more details.**

truth image using bicubic interpolation, and calculate the average PSNR, SSIM, and LPIPS [65]. More details are provided in the supplementary materials. For those competing methods that need to explicitly produce HR ERP images to render final viewports, we evaluate the WS-PSNR [44] of their predicted  $\hat{I}^{\text{HR-ERP}}$ .

## 4.2 Comparison with State-of-the-Art Methods

We compare three categories of methods: (1) a baseline method which downscale with Bicubic interpolation, compresses with standard JPEG codec, and renders viewports with Bicubic interpolation; (2) SR pipeline which downscale with Bicubic, compresses with standard JPEG codec, upscale with state-of-the-art methods [64], and uses bicubic interpolation for viewport rendering; (3) Rescaling pipeline which firstly uses state-of-the-art asymmetric rescaling methods [39] and renders viewports with Bicubic interpolation. For a fair comparison, we retrain SR methods [64] and rescaling methods [39] with our training sets and constrain their bpp around 0.3 by adjusting the quality factor of JPEG compression in all baselines.

Tab. 1 presents the quantitative comparisons of the quality of rendered viewports with  $(F_h, F_v) = (120^\circ, 90^\circ)$  and  $(w_v, h_v) = (2048, 1536)$  among different methods. Taking advantage of directly

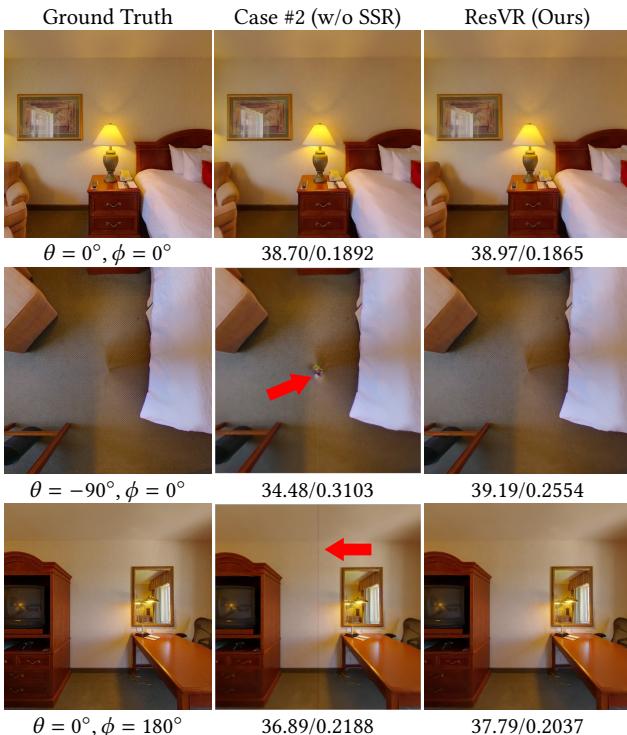
optimizing the final viewport through end-to-end training, ResVR outperforms previous methods by reconstruction accuracy (about 0.4dB gain on PSNR) and with better realness (about 0.02 gain on LPIPS). Notably, even though existing rescaling methods can achieve PSNR of about 28dB, they only obtain sub-optimal results due to the lack of awareness and optimization of the viewport rendering process, which demonstrates the fact that focusing solely on the quality of ERP images results in sub-optimal visual experience of viewports. ResVR directly optimizes the rendered viewports without the need to predict HR ERP and achieves SOTA performance. Fig. 6 further provides the visual comparison of rendered viewports. It can be seen that our ResVR reconstructs enhanced details with fewer artifacts (e.g. the text “labatt’s” in the top image and the lines in the bottom image) compared to other methods. Additionally, ResVR shows better wraparound continuity at the location of the seams (e.g., pipe in the zoomed-in area) in the bottom case. We attribute this to the awareness of arbitrary view directions during training and our MLP’s continuous representation of images.

## 4.3 Ablation Study

In this section, we analyze the effect of the proposed discrete pixel sampling strategy and spherical pixel shape representation. We

**Table 2: Ablation study on proposed discrete pixel sampling strategy (DPS) and spherical pixel shape representation (SSR) with setting  $(F_h, F_v) = (90^\circ, 90^\circ)$  and  $(w_v, h_v) = (1024, 1024)$ .**

Case	Test Set	DPS	SSR	PSNR↑	SSIM↑	LPIPS↓
#1 <b>ResVR</b>	ODISR	✗	✗	27.35	0.8296	0.3321
		✓	✗	<b>31.47</b>	<b>0.8560</b>	<b>0.2820</b>
		✓	✓	<b>31.87</b>	<b>0.8606</b>	<b>0.2696</b>
		✗	✗	28.09	0.8503	0.3040
#2 <b>ResVR</b>	SUN360	✓	✗	<b>32.53</b>	<b>0.8742</b>	<b>0.2443</b>
		✓	✓	<b>33.08</b>	<b>0.8793</b>	<b>0.2324</b>



**Figure 7: Visual comparison of two variants (Case #2 and Ours).** 2D shape representation falls in high-latitude and high-longitude areas (highlighted by red arrow). In contrast, our spherical representation ensures stable rendering results in various directions with better PSNR (dB)↑ and LPIPS ↓.

conduct experiments on three ResVR variants: (1) Case #1: We train the rescaling process and VR module separately with 2D shape representation; (2) Case #2: We train the whole pipeline in an end-to-end manner by employing discrete pixel sampling strategy with 2D shape representation; (3) The complete ResVR with our sphere pixel shape representation. Quantitative results are shown in Tab. 2.

**Effect of discrete pixel sampling strategy.** Comparison #1 vs. #2 exhibits the effectiveness of the proposed discrete pixel sampling strategy which enables end-to-end training of the whole pipeline, thus providing substantial PSNR improvement of 4.12-4.44dB. To further analyze the sampled coordinates and pixels, we visualize the sampled areas and pixels in the supplementary materials.

**Table 3: Quantitative evaluation on viewports of different resolutions with  $(F_h, F_v) = (120^\circ, 120^\circ)$ . ResVR outperforms existing methods under various resolutions. Bic: Bicubic interpolation, Res: Resolution, HT: HyperThumbnail [39].**

Method	Res	ODI-SR [15]			SUN 360 [50]		
		PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
HT [39] & Bic	512 <sup>2</sup>	<b>30.89</b>	<b>0.8704</b>	<b>0.1976</b>	<b>32.08</b>	<b>0.8880</b>	<b>0.1630</b>
<b>ResVR (Ours)</b>		<b>31.03</b>	<b>0.8754</b>	<b>0.1888</b>	<b>32.16</b>	<b>0.8914</b>	<b>0.1527</b>
HT [39] & Bic	1024 <sup>2</sup>	<b>30.92</b>	<b>0.8498</b>	<b>0.2802</b>	<b>32.09</b>	<b>0.8718</b>	<b>0.2343</b>
<b>ResVR (Ours)</b>		<b>31.25</b>	<b>0.8579</b>	<b>0.2632</b>	<b>32.39</b>	<b>0.8782</b>	<b>0.2176</b>
HT [39] & Bic	2048 <sup>2</sup>	<b>30.93</b>	<b>0.8541</b>	<b>0.3230</b>	<b>32.08</b>	<b>0.8769</b>	<b>0.2713</b>
<b>ResVR (Ours)</b>		<b>31.32</b>	<b>0.8596</b>	<b>0.3046</b>	<b>32.48</b>	<b>0.8812</b>	<b>0.2567</b>

**Effect of spherical pixel shape representation.** Comparison #2 vs. ResVR reveals that the spherical shape representation improves PSNR by 0.40-0.55dB. We observe that conventional 2D shape representation methods [23] fall in high-longitude and high-latitude areas, as depicted in Fig. 7. We attribute this to the following reasons: adjacent pixels on the viewport should be very close to each other on the sphere. However, (1) for high-latitude areas, due to the natural distortion of ERP, the distance between corresponding points is very close to the original sphere but is very far on the ERP image. (2) For high-longitude areas, due to the wraparound consistency of the left and right ends of ERP, the corresponding points are originally adjacent pixels on the sphere but are at both ends of the ERP image. The traditional shape representation method based on 2D ERP ignores the spherical characteristics, so in the above situations, it provides wrong pixel shape priors for the VR module, thus leading to abnormal visual results. The proposed SSR is performed on the native spherical surface, thus ensuring stable and high-quality rendering quality in various view directions.

#### 4.4 Analysis

**LR JPEG image.** The file size and visual quality of LR JPEG images are also important for efficient transmission and user preview. ResVR effectively balances transmission efficiency with the final viewer experience through our end-to-end training. Visualizations of LR images are provided in the supplementary materials.

**Arbitrary-resolution viewport rendering.** Due to the continuous representation ability [12, 23, 24] of the MLP in our VR module, ResVR is able to render viewports of different resolutions using only one set of parameters. Tab. 3 compares ResVR with SOTA methods on different resolution settings. It can be seen that ResVR achieves better performance at different resolutions.

## 5 CONCLUSION

The rise of virtual reality and augmented reality applications has popularized ODI rescaling to shrink the file size of images while maintaining their quality. However, existing methods focus on improving the ERP image quality, ignoring that HMDs use rendered viewports for display, not ERP images. This focus on ERP quality alone will lead to compromised user experiences. To address this, in this paper, we propose ResVR, a novel framework for joint rescaling and viewport rendering of ODIs. In ResVR, we develop a discrete pixel sampling strategy to tackle the irregular correspondence between the ERP area and the viewport, enabling end-to-end training of the whole pipeline. A spherical pixel shape representation

technique is introduced to serve as an effective guidance for the rendering process, further enhancing the visual quality of viewports. Experiments demonstrate that our ResVR achieves state-of-the-art performance across different settings of FoVs, view directions, and resolutions while keeping a low transmission overhead.

**Limitations and future work.** Existing ODI datasets are of relatively low resolution and with compression artifacts, which limits the further performance improvements of ResVR. Besides, although ResVR provides a promising pipeline for comprehensive ODI processing and achieves SOTA performance, it is crucial to extend ResVR to 360° video tasks [47]. We leave this for future work.

## REFERENCES

- [1] Hao Ai, Zidong Cao, Yan-Pei Cao, Ying Shan, and Lin Wang. 2023. HRDFuse: Monocular 360deg depth estimation by collaboratively learning holistic-with-regional depth distributions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 13273–13282.
- [2] Hao Ai, Zidong Cao, Jinjing Zhu, Haotian Bai, Yucheng Chen, and Ling Wang. 2022. Deep Learning for Omnidirectional Vision: A Survey and New Perspectives. *arXiv preprint arXiv:2205.10468* (2022).
- [3] Johannes Ballé, Valero Laparra, and Eero P Simoncelli. 2016. End-to-end optimized image compression. *arXiv preprint arXiv:1611.01704* (2016).
- [4] Jens Behrmann, Will Grathwohl, Ricky TQ Chen, David Duvenaud, and Jörn-Henrik Jacobsen. 2019. Invertible residual networks. In *Proceedings of the International Conference on Machine Learning (ICML)*. PMLR, 573–582.
- [5] Mingdeng Cao, Chong Mou, Fanghua Yu, Xiantao Wang, Yinqi Zheng, Jian Zhang, Chao Dong, Gen Li, Ying Shan, Radu Timofte, et al. 2023. Ntire 2023 challenge on 360deg omnidirectional image and video super-resolution: Datasets, methods and results. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 1731–1745.
- [6] Zidong Cao, Hao Ai, Yan-Pei Cao, Ying Shan, Xiaohu Qie, and Lin Wang. 2023. OmniZoomer: Learning to Move and Zoom in on Sphere at High-Resolution. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 12897–12907.
- [7] Zidong Cao, Hao Ai, and Lin Wang. 2023. 360deg High-Resolution Depth Estimation via Uncertainty-aware Structural Knowledge Transfer. *arXiv preprint arXiv:2304.07967* (2023).
- [8] Robert Carroll, Maneesh Agrawala, and Aseem Agarwala. 2009. Optimizing content-preserving projections for wide-angle images. *ACM Trans. Graph.* 28, 3 (2009), 43.
- [9] Bin Chen, Zhenyu Zhang, WeiQi Li, Chen Zhao, Jiwen Yu, Shijie Zhao, Jie Chen, and Jian Zhang. 2024. Invertible Diffusion Models for Compressed Sensing. *arXiv preprint arXiv:2403.17006* (2024).
- [10] Ricky TQ Chen, Jens Behrmann, David K Duvenaud, and Jörn-Henrik Jacobsen. 2019. Residual flows for invertible generative modeling. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*.
- [11] X Chen, X Wang, J Zhou, and C Dong. 2023. Activating More Pixels in Image Super-Resolution Transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 22367–22377.
- [12] Yinbo Chen, Sifei Liu, and Xiaolong Wang. 2021. Learning continuous image representation with local implicit image function. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 8628–8638.
- [13] Ka Leong Cheng, Yueqi Xie, and Qifeng Chen. 2021. licnet: A generic framework for reversible image conversion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 1991–2000.
- [14] Ming Cheng, Haoyu Ma, Qiufang Ma, Xiaopeng Sun, Weiqi Li, Zhenyu Zhang, Xuhan Sheng, Shijie Zhao, Junlin Li, and Li Zhang. 2023. Hybrid transformer and cnn attention network for stereo image super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshop (CVPRW)*. 1702–1711.
- [15] Xin Deng, Haos Wang, Mai Xu, Yichen Guo, Yuhang Song, and Li Yang. 2021. Lau-net: Latitude adaptive upscaling network for omnidirectional image super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 9189–9198.
- [16] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. 2015. Image super-resolution using deep convolutional networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* 38, 2 (2015), 295–307.
- [17] Yichen Guo, Mai Xu, Lai Jiang, Leonid Sigal, and Yunjin Chen. 2023. DINN360: Deformable Invertible Neural Network for Latitude-Aware 360deg Image Rescaling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 21519–21528.
- [18] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. 2017. Densely connected convolutional networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 4700–4708.
- [19] Falah Jabar, João Ascenso, and Maria Paula Queluz. 2017. Perceptual analysis of perspective projection for viewport rendering in 360° images. In *2017 IEEE International Symposium on Multimedia (ISM)*. IEEE, 53–60.
- [20] Falah Jabar, João Ascenso, and Maria Paula Queluz. 2019. Objective assessment of perceived geometric distortions in viewport rendering of 360° images. *IEEE Journal of Selected Topics in Signal Processing (JSTSP)* 14, 1 (2019), 49–63.
- [21] Minsu Kim, Jaewon Lee, Byeonghun Lee, Sunghoon Im, and Kyong Hwan Jin. 2024. Implicit Neural Image Stitching With Enhanced and Blended Feature Reconstruction. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. 4087–4096.
- [22] Max D. Larsen and James L. Fejfar. 1974. *INTRODUCTION TO GEOMETRY*. 225.
- [23] Jaewon Lee, Kwang Pyo Choi, and Kyong Hwan Jin. 2022. Learning local implicit fourier representation for image warping. In *Proceedings of the European Conference on Computer Vision (ECCV)*. Springer, 182–200.
- [24] Jaewon Lee and Kyong Hwan Jin. 2022. Local texture estimator for implicit representation function. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 1929–1938.
- [25] Jiacheng Li, Chang Chen, Wei Huang, Zhiqiang Lang, Fenglong Song, Youliang Yan, and Zhiwei Xiong. 2023. Learning steerable function for efficient image resampling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 5866–5875.
- [26] Runyi Li, Xuhan Sheng, Weiqi Li, and Jian Zhang. 2024. OmniSSR: Zero-shot Omnidirectional Image Super-Resolution using Stable Diffusion Model. *arXiv preprint arXiv:2404.10312* (2024).
- [27] Weiqi Li, Bin Chen, and Jian Zhang. 2022. D3c2-net: Dual-domain deep convolutional coding network for compressive sensing. *arXiv preprint arXiv:2207.13560* (2022).
- [28] Jingyun Liang, Jiezhang Cao, Guolei Sun, Kai Zhang, Luc Van Gool, and Radu Timofte. 2021. Swinir: Image restoration using swin transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 1833–1844.
- [29] Jingyun Liang, Andreas Lugmayr, Kai Zhang, Martin Danelljan, Luc Van Gool, and Radu Timofte. 2021. Hierarchical conditional flow: A unified framework for image super-resolution and image rescaling. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 4076–4085.
- [30] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. 2017. Enhanced deep residual networks for single image super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. 136–144.
- [31] Xinqi Lin, Jingwen He, Ziyuan Chen, Zhaoyang Lyu, Ben Fei, Bo Dai, Wanli Ouyang, Yu Qiao, and Chao Dong. 2023. Diffbir: Towards blind image restoration with generative diffusion prior. *arXiv preprint arXiv:2308.15070* (2023).
- [32] Hongying Liu, Zhubo Ruan, Chaowei Fang, Peng Zhao, Fanhua Shang, Yuanyuan Liu, and Lijun Wang. 2020. A single frame and multi-frame joint network for 360-degree panorama video super-resolution. *arXiv preprint arXiv:2008.10320* (2020).
- [33] Qin Liu, Letong Han, Rui Tan, Hongfei Fan, Weiqi Li, Hongming Zhu, Bowen Du, and Sicong Liu. 2021. Hybrid attention based residual network for pansharpening. *Remote Sensing* 13, 10 (2021), 1962.
- [34] Don P Mitchell and Arun N Netravali. 1988. Reconstruction filters in computer-graphics. *ACM SIGGRAPH Computer Graphics* 22, 4 (1988), 221–228.
- [35] Akito Nishiyama, Satoshi Ikehata, and Kiyoharu Aizawa. 2021. 360 single image super resolution via distortion-aware network and distorted perspective images. In *2021 IEEE International Conference on Image Processing (ICIP)*. IEEE, 1829–1833.
- [36] Hao Ouyang, Tengfei Wang, and Qifeng Chen. 2022. Restorable image operators with quasi-invertible networks. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, Vol. 36. 2008–2016.
- [37] Cagri Ozcinar, Aakanksha Rana, and Aljosa Smolic. 2019. Super-resolution of omnidirectional images using adversarial learning. In *2019 IEEE 21st International Workshop on Multimedia Signal Processing (MMSP)*. IEEE, 1–6.
- [38] Zhihong Pan, Baopu Li, Dongliang He, Mingde Yao, Wenhao Wu, Tianwei Lin, Xin Li, and Errui Ding. 2022. Towards bidirectional arbitrary image rescaling: Joint optimization and cycle idempotence. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 17389–17398.
- [39] Chenyang Qi, Xian Yang, Ka Leong Cheng, Ying-Cong Chen, and Qifeng Chen. 2023. Real-time 6K Image Rescaling with Rate-distortion Optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 14092–14101.
- [40] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. U-net: Convolutional networks for biomedical image segmentation. In *Proceedings of International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*. Springer, 234–241.
- [41] Sanghyun Son and Kyoung Mu Lee. 2021. Srwarp: Generalized image super-resolution under arbitrary transformation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 7782–7791.
- [42] Wanjie Sun and Zhenzhong Chen. 2020. Learned image downscaling for upscaling using content adaptive resampler. *IEEE Transactions on Image Processing (TIP)* 29 (2020), 4027–4040.

- [43] Xiaopeng Sun, Weiqi Li, Zhenyu Zhang, Qiufang Ma, Xuhan Sheng, Ming Cheng, Haoyu Ma, Shijie Zhao, Jian Zhang, Junlin Li, et al. 2023. OPDN: Omnidirectional position-aware deformable network for omnidirectional image super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshop (CVPRW)*. 1293–1301.
- [44] Yule Sun, Ang Lu, and Lu Yu. 2017. Weighted-to-spherically-uniform quality evaluation for omnidirectional video. *IEEE Signal Processing Letters* 24, 9 (2017), 1408–1412.
- [45] Gregory K Wallace. 1991. The JPEG still picture compression standard. *Commun. ACM* 34, 4 (1991), 30–44.
- [46] Jianyi Wang, Zongsheng Yue, Shangchen Zhou, Kelvin CK Chan, and Chen Change Loy. 2023. Exploiting diffusion prior for real-world image super-resolution. *arXiv preprint arXiv:2305.07015* (2023).
- [47] Qian Wang, Weiqi Li, Chong Mou, Xinhua Cheng, and Jian Zhang. 2024. 360DVD: Controllable Panorama Video Generation with 360-Degree Video Diffusion Model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.
- [48] Xintao Wang, Liangbin Xie, Chao Dong, and Ying Shan. 2021. Real-esrgan: Training real-world blind super-resolution with pure synthetic data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 1905–1914.
- [49] Yue Wu, Guotao Meng, and Qifeng Chen. 2021. Embedding novel views in a single jpeg image. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 14519–14527.
- [50] Jianxiong Xiao, Krista A Ehinger, Aude Oliva, and Antonio Torralba. 2012. Recognizing scene viewpoint using panoramic place representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2695–2702.
- [51] Mingqing Xiao, Shuxin Zheng, Chang Liu, Zhouchen Lin, and Tie-Yan Liu. 2023. Invertible rescaling network and its extensions. *International Journal of Computer Vision (IJCV)* 131, 1 (2023), 134–159.
- [52] Mingqing Xiao, Shuxin Zheng, Chang Liu, Yaolong Wang, Di He, Guolin Ke, Jiang Bian, Zhouchen Lin, and Tie-Yan Liu. 2020. Invertible image rescaling. In *Proceedings of the European Conference on Computer Vision (ECCV)*. Springer, 126–144.
- [53] Yueqi Xie, Ka Leong Cheng, and Qifeng Chen. 2021. Enhanced invertible encoding for learned image compression. In *Proceedings of the 29th ACM International Conference on Multimedia (ACM MM)*. 162–170.
- [54] Qunliang Xing, Mai Xu, Shengxi Li, Xin Deng, Meisong Zheng, Huaida Liu, and Ying Chen. 2024. Enhancing Quality of Compressed Images by Mitigating Enhancement Bias Towards Compression Domain. *arXiv preprint arXiv:2402.17200* (2024).
- [55] Yazhou Xing, Zian Qian, and Qifeng Chen. 2021. Invertible image signal processing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 6287–6296.
- [56] Bingna Xu, Yong Guo, Luoqian Jiang, Mianjie Yu, and Jian Chen. 2023. Downscaled representation matters: Improving image rescaling with collaborative downsampled images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 12237–12247.
- [57] Mai Xu, Lai Jiang, Chen Li, Zulin Wang, and Xiaoming Tao. 2020. Viewport-based CNN: A multi-task approach for assessing 360° video quality. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* 44, 4 (2020), 2198–2215.
- [58] Mai Xu, Chen Li, Shanyi Zhang, and Patrick Le Callet. 2020. State-of-the-art in 360 video/image processing: Perception, assessment and compression. *IEEE Journal of Selected Topics in Signal Processing (JSTSP)* 14, 1 (2020), 5–26.
- [59] Jinhai Yang, Mengxi Guo, Shijie Zhao, Junlin Li, and Li Zhang. 2023. Self-asymmetric invertible network for compression-aware image rescaling. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, Vol. 37. 3155–3163.
- [60] Li Yang, Mai Xu, Shengxi Li, Yichen Guo, and Zulin Wang. 2022. Blind VQA on 360° Video via Progressively Learning From Pixels, Frames, and Video. *IEEE Transactions on Image Processing (TIP)* 32 (2022), 128–143.
- [61] Li Yang, Mai Xu, Tie Liu, Liangyu Huo, and Xinbo Gao. 2022. TVFormer: Trajectory-guided visual quality assessment on 360° images with transformers. In *Proceedings of the 30th ACM International Conference on Multimedia (ACM MM)*. 799–808.
- [62] Shuzhou Yang, Moxuan Ding, Yanmin Wu, Zihan Li, and Jian Zhang. 2023. Implicit neural representation for cooperative low-light image enhancement. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 12918–12927.
- [63] Youngho Yoon, Inchul Chung, Lin Wang, and Kuk-Jin Yoon. 2022. SphereSR: 360deg Image Super-Resolution With Arbitrary Projection via Continuous Spherical Image Representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 5677–5686.
- [64] Fanghua Yu, Xintao Wang, Mingdeng Cao, Gen Li, Ying Shan, and Chao Dong. 2023. OSRT: Omnidirectional Image Super-Resolution with Distortion-aware Transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.
- [65] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. 2018. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 586–595.
- [66] Xuanyu Zhang, Runyi Li, Jiwen Yu, Youmin Xu, Weiqi Li, and Jian Zhang. 2024. Editguard: Versatile image watermarking for tamper localization and copyright protection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.
- [67] Xuanyu Zhang, Yongbing Zhang, Ruiqin Xiong, Qilin Sun, and Jian Zhang. 2022. Herosnet: Hyperspectral explicable reconstruction and optimal sampling deep network for snapshot compressive imaging. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 17532–17541.
- [68] Yulun Zhang, Kunpeng Li, Kai Li, Lichen Wang, Bineng Zhong, and Yun Fu. 2018. Image super-resolution using very deep residual channel attention networks. In *Proceedings of the European Conference on computer vision (ECCV)*. 286–301.
- [69] Yulun Zhang, Yapeng Tian, Yu Kong, Bineng Zhong, and Yun Fu. 2018. Residual dense network for image super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2472–2481.