Check for updates

Invertible Rescaling Network and Its Extensions

Mingqing Xiao[1] · Shuxin Zheng[2] · Chang Liu[2] · Zhouchen Lin[1,3] · Tie-Yan Liu[2]

Abstract

Image rescaling is a commonly used bidirectional operation, which first downscales high-resolution images to fit various display screens or to be storage- and bandwidth-friendly, and afterward upscales the corresponding low-resolution images to recover the original resolution or the details in the zoom-in images. However, the non-injective downscaling mapping discards high-frequency contents, leading to the ill-posed problem for the inverse restoration task. This can be abstracted as a general image degradation–restoration problem with information loss. In this work, we propose a novel invertible framework to handle this general problem, which models the bidirectional degradation and restoration from a new perspective, i.e. invertible bijective transformation. The invertibility enables the framework to model the information loss of pre-degradation in the form of distribution, which could mitigate the ill-posed problem during post-restoration. To be specific, we develop invertible models to generate valid degraded images and meanwhile transform the distribution of lost contents to the fixed distribution of a latent variable during the forward degradation. Then restoration is made tractable by applying the inverse transformation on the generated degraded image together with a randomly-drawn latent variable. We start from image rescaling and instantiate the model as Invertible Rescaling Network, which can be easily extended to the similar decolorization–colorization task. We further propose to combine the invertible framework with existing degradation methods such as image compression for wider applications. Experimental results demonstrate the significant improvement of our model over existing methods in terms of both quantitative and qualitative evaluations of upscaling and colorizing reconstruction from downscaled and decolorized images, and rate-distortion of image compression. Code is available at https://github.com/pkuxmq/Invertible-Image-Rescaling.

1 Introduction

Image rescaling is becoming increasingly important in the age of high-resolution (HR) images/videos explosion on the Internet. For efficient storage, transmission, and sharing, such large-sized data are usually downscaled to significantly reduce the size and become bandwidth-friendly (Bruckstein et al., 2003; Li et al., 2018; Lin & Dong, 2006; Shen et al., 2011; Wu et al., 2009), while visually valid contents are maintained (Kim et al., 2018; Sun and Chen, 2020) for previewing or fitting for screens with different resolutions. On the other hand, the inverse restoration task is required by user demands, which aims to upscale the downscaled low-resolution (LR) images to a higher resolution or the original size (Giachetti & Asuni, 2011; Schulter et al., 2015; Yeo et al., 2017, 2018) so that vivid details could be presented. How-

✉ Chang Liu
Chang.Liu@microsoft.com

✉ Zhouchen Lin
zlin@pku.edu.cn

Mingqing Xiao
mingqing_xiao@pku.edu.cn.com

Shuxin Zheng
Shuxin.Zheng@microsoft.com

Tie-Yan Liu
Tie-Yan.Liu@microsoft.com

[1] Key Laboratory of Machine Perception (MoE), School of Intelligence Science and Technology, Peking University, Beijing 100871, People's Republic of China

[2] Machine Learning Group, Microsoft Research Asia, Beijing, People's Republic of China

[3] Peng Cheng Laboratory, Shenzhen, People's Republic of China

ever, the non-injective downscaling would cause information loss, as high-frequency contents are lost during downscaling according to the Nyquist-Shannon sampling theorem (Shannon, 1949). Such information loss leads to an intractable ill-posed problem of the inverse tasks (Dong et al., 2015; Glasner et al., 2009a; Yang et al., 2010), since the same downscaled LR image may correspond to multiple possible HR images, and therefore poses great challenges for recovery.

This can be abstracted as a general image degradation–restoration problem with information loss due to dimension reduction. Similar examples also include image decolorization–colorization (Xia et al., 2018; Ye et al., 2020) and image compression. In the following, we first focus on this general problem and then consider the specific instantiation examples.

There have been many efforts attempting to mitigate this ill-posed problem with machine learning algorithms. For instance, many works consider dealing with the unidirectional restoration task, e.g. for image rescaling, they choose super-resolution (SR) methods to upscale LR images by imposing or learning a prior, i.e. a preference on all possible HR images corresponding to a given LR image, for this inverse task. However, mainstream SR algorithms (Dai et al., 2019; Dong et al., 2015; Lim et al., 2017; Wang et al., 2018; Zhang et al., 2018a, 2018b) leverage a predefined and non-adjustable downscaling method, such as Bicubic interpolation, to guide the learning of upscaling, which omits the compatibility between these two mutually-inverse operations. Therefore, simply applying unidirectional restoration methods, e.g. SR, cannot fully leverage the bidirectional nature of the task, resulting in unsatisfactory recoveries.

Some recent works attempt to unify these bidirectional operations through an encoder-decoder framework rather than separating them as two independent tasks. In these methods for image rescaling, an encoder, which serves as a learning-based upscaling-optimal downscaling module, is jointly trained with an upscaling decoder (Kim et al., 2018) or an existing SR module (Li et al., 2018; Sun and Chen, 2020). This encoder-decoder framework is also applied in similar degradation–restoration tasks (Xia et al., 2018; Ye et al., 2020). Taking the bidirectional nature into consideration, such an integrated training method can largely improve the quality of image reconstruction. However, these efforts simply link the two operations through training objectives without any attempt to fully leverage the reciprocal nature of the tasks or capture features of lost contents. So the results cannot meet the expectation as well.

In this paper, we propose a novel invertible framework to largely mitigate this ill-posed problem through invertible bijective transformation. With inspiration from the reciprocal nature of this pair of tasks, keeping the knowledge of lost information in the forward procedure, e.g. high-frequency contents in the image rescaling task, would greatly help the

inverse recovery. However, it is not acceptable to store or transfer all lost contents to enable an exact recovery. To well address this challenge, we instead deal with these contents in the form of distribution, with the assumption that reasonable lost contents follows a distribution. We develop a novel invertible model to capture the knowledge of distribution in the form of distribution transformation function. Specifically, in the forward procedure, our invertible models will transform the original image x into a degraded image y and an auxiliary latent variable z by an invertible transformation. y belongs to a target set of valid degraded images, e.g. the set of visually-pleasing LR images given the HR image x for the image rescaling tasks, and z is a random variable following a fixed pre-specified distribution $p(z)$ (e.g. isotropic Gaussian). The joint distribution of y and z is bijectively transformed from the distribution of x and therefore the random variable z holds the lost "information" of y from the perspective of statistical modeling.[1] Learning this bijective transformation enables our model to capture the knowledge of lost contents. Then during the inverse restoration procedure, a random sample of z from the pre-specified distribution, together with the degraded image y, could recover most contents for the original image through the inverse function of the model. We consider two instantiation examples of the this bidirectional problem, i.e. image rescaling and image decolorization–colorization. As for the specific architectures, we start from image rescaling and develop Invertible Rescaling Network (IRN), which can be easily extended and adapted to decolorization–colorization.

To realize this invertible framework, several challenges should be tackled during training. Our basic targets include reconstructing original images with high quality and generating degraded images belonging to a target set, e.g. the set of visually-pleasing LR images. A further objective is to accomplish the restoration with an image-agnostic z, i.e., $z \sim p(z)$ instead of an image-specific $z \sim p(z|y)$, because it is easier for statistical modeling and sampling the independent $p(z)$ without the effort of handling conditions y. This is achievable since for any random vector with a density (i.e. $z' \sim p(z'|y)$), there exists a bijection f_y such that $f_y(z') \sim N(0, I)$ (Hyvärinen & Pajunen, 1999).[2] For these

---

[1] Note that the term "information" in this sentence means "uncertainty" of random variables from the definition of information theory, which does not imply that specific lost contents are "encoded" in z. The knowledge about lost contents is in our invertible model in the form of the bijective transformation between x and (y, z).

[2] This can be viewed as transferring the dependence of z on y into the process of our model that bijectively transforms mixed y and z into x. This treatment avoids the manual allocation of model capacity between capturing the y-dependency of the process from z to x and the y-dependency of the distribution of z, and make it easier for statistical modeling and sampling the random variable z. The restoration process, i.e. the inverse transformation of our model with inputs y and z, is still dependent on the image content y.

purposes, we combine a reconstruction loss, a guidance loss, and a distribution matching loss to formulate a novel compact and effective objective function. Note that the last component aims at aligning recovered images with the true original image manifold as well as enforcing z to follow the image-agnostic distribution $p(z)$, which cannot be simply achieved by conventional generative adversarial networks (GANs) nor the maximum likelihood estimation (MLE) method. This is because our invertible model does not give a marginal distribution on the data (it is not a simple generative model), and these conventional methods do not guide the distribution in the latent space for degraded image generation. We formulate the distribution on y as the pushed-forward empirical distribution of x, which would inversely pass our invertible model in company with an independent distribution $p(z)$, to recover the distribution of x. Therefore, our distribution matching focuses on this recovered one and the data distribution of x, and we minimize the JS divergence between them in practice, as other alternative methods such as sample-based maximum mean discrepancy (MMD) method (Ardizzone et al., 2019) could poorly handle the high-dimensional data in our task. Moreover, we show that once the distribution matching on x is achieved, the matching also holds on the (y, z) space with z being image-agnostic, according to the invertible nature of our model.

Furthermore, we propose the combination between our invertible framework and existing degradation methods, and instantiate it by the combination of image rescaling and image compression. Since parts of degradation operations are not always available for adaption with restoration, e.g. image compression has common formats with general standards for convenient and wide applications, we study this combination to enable more applications. We demonstrate the effectiveness to combine our invertible framework with restoration from such degradation. We note that there could be many other generalized applications of the invertible framework and model as well, such as image steganography, video rescaling, image denoising, etc. Please refer to recent works that adapt the invertible framework and model into various tasks since the publication of our preliminary version of this work[3] for more details (Cheng et al., 2021; Huang et al., 2021; Jing et al., 2021; Liu et al., 2021; Lu et al., 2021; Tian et al., 2021; Xing et al., 2021; Xie et al., 2021; Zhao et al., 2021). Our contributions are concluded as follows:

- To our best knowledge, we are the first to model mutually-inverse image degradation and restoration with an invertible bijective transformation.[3] The deliberately designed invertibility enables the framework to model the infor-

mation loss, which can mitigate the ill-posed nature in this bidirectional problem.
- We propose a novel model design and efficient training objectives to realize this framework. It enforces the latent variable z to obey a simple image-agnostic distribution, which enables efficient inverse upscaling based on a sample from the distribution. We develop IRN with deliberately designed architecture for the image rescaling task and demonstrate the easy adaptation to the similar image decolorization–colorization task.
- The proposed IRN and its scale-flexible and efficient variants achieve significant performance improvement of reconstructed HR images from the downscaled LR images, compared with state-of-the-art downscaling-SR and encoder-decoder methods. Meanwhile, the largely reduced parameters of IRN compared with these methods indicate the lightweight property and high efficiency of our model.
- We further propose the combination between our invertible framework and restoration from existing degradation methods, e.g. combination of image rescaling and compression, for more general applications. Experiments show improvements in these scenarios as well.

2 Related Work

2.1 Image Upscaling After Downscaling

When only the unidirectional upscaling task is considered, image super-resolution (SR) is a widely adopted method with promising results in low-resolution (LR) image upscaling. SR works focus on mitigating the inherent ill-posed problem by learning strong prior information by example-based strategy (Freedman & Fattal, 2011; Glasner et al., 2009b; Kim & Kwon, 2010; Schulter et al., 2015) or deep learning models (Dai et al., 2019; Dong et al., 2015; Guo et al., 2020; Lim et al., 2017; Lugmayr et al., 2020; Wang et al., 2018; Zhang et al., 2018b, a; Zhong et al., 2018). The state-of-the-art SR models are to train a neural network with elaborately designed architecture to reconstruct high-resolution (HR) images from the LR counterparts, which are usually generated by Bicubic interpolation from the HR images. However, when it comes to the bidirectional task of image rescaling, considering the image downscaling method would largely benefit the upscaling reconstruction.

Traditional image downscaling methods sub-sample images by a low-pass filter with frequency-based kernels, such as Bilinear, Bicubic, etc. (Mitchell & Netravali, 1988). For perceptual quality, several detail- or structure-preserving downscaling methods were proposed recently (Kopf et al., 2013; Liu et al., 2017; Oeztireli & Gross, 2015; Wang et al., 2004; Weber et al., 2016) to mitigate the over-smoothness

---

of generated LR images. When the potential mutual reinforcement between downscaling and the inverse upscaling task is considered, the upscaling-optimal downscaling methods, which aim to learn the optimal downscaling model for the post-upscaling operation, gain increasing attention and efforts. For example, Kim et al. (2018) proposed a task-aware downscaling model based on an auto-encoder framework, which jointly trains the downscaling encoder and upscaling decoder as a united task. Similarly, Li et al. (2018) used a CNN to estimate downscaled images while a learned or specified SR model is adopted for HR image recovery. Recently, Sun and Chen (2020) proposed a new content-adaptive-resampler-based image downscaling method, which is jointly trained with existing differentiable upscaling (SR) models. And Chen et al. (2020) proposed a downscaling network based on the discretization of Hamiltonian System, which is trained jointly with SR models as well. Although these efforts take the bidirectional nature of image rescaling into consideration, they simply link downscaling and upscaling through training objectives while ignoring the lost information during downscaling that leads to the ill-posed problem they suffer from. In this paper, we propose to model the bidirectional downscaling and upscaling processes with invertible transformation based on invertible neural networks, which could model the lost information and largely mitigate the ill-posed problem.

Difference from Super-Resolution Please note that the task of image rescaling is different from super-resolution. In our scenario, ground-truth HR images are available at the beginning but we have to use the LR version (e.g. for transmission or preview) instead. We would generate LR images and hope to recover the HR ones afterward from them. While for SR, the target is to generate new HR images for any given LR images.

2.2 Image Decolorization–Colorization

Image decolorization methods convert color images to grayscale, which enables applications like aesthetic photography, backward compatibility for legacy display, etc. (Xia et al., 2018), while colorization methods aim to colorize grayscale images. Reconstructing original color images from the decolorized ones is also a bidirectional task with information loss, as color information is lost during decolorization and needs to be recovered, which can be viewed as "downscaling" and "upscaling" in the color channel dimension.

Image colorization methods could be used to colorize decolorized images, and existing methods usually requires user-hints (Levin et al., 2004; Zhang et al., 2017) or learning strong priors by deep learning models (Ardizzone et al., 2019; Deshpande et al., 2017; Zhang et al., 2016) to generate color for grayscale images. When it comes to precisely recovering

the original color of decolorized images, taking decolorization methods into consideration would help reconstruction as well.

The most commonly used image decolorization method is to only take the luminance channel and discard color information in color space. Later, several methods have been proposed to preserve the color contrast or structural information which is easily lost during color-to-gray conversion (Bala & Eschbach, 2004; Liu et al., 2015). Taking decolorization and colorization as a joint task, Xia et al. (2018) first proposed invertible grayscale, which leverages an encoder-decoder architecture of deep learning models to learn to generate grayscale images that is helpful for colorization reconstruction. Ye et al. (2020) further improved the network design under this architecture. Kim et al. (2018) also demonstrates the extension of their image rescaling method for this task. However, these methods do not explicitly model the lost information and still significantly suffer from the ill-posed problem. In this work, we demonstrate that our proposed invertible framework could adapt to this bidirectional task well.

2.3 Image Compression

Image compression is a kind of data compression on digital images, which can be lossy (e.g. JPEG, BPG) or lossless (e.g. PNG, BMP). Traditional lossy image compression usually involves quantization in the frequency domain and optimal coding rules, while recently image compression methods based on deep learning show promising results of compression ratio and image quality (Agustsson, 2019; Ballé et al., 2017, 2018; Cheng et al., 2020; Li et al., 2020; Minnen et al., 2018; Rippel & Bourdev, 2017; Wang et al., 2020). As image compression is only for storage saving, it will not change the resolution of images and there is no visually meaningful low-resolution image but only bit-stream output. Therefore image compression is different from image rescaling and their methods are usually different.

Despite this, image rescaling is orthogonal to image compression: they can be combined naturally and be applied together in many real applications (Sullivan et al., 2013). On one hand, the downscaled low-resolution images could be encoded by advanced lossless compression methods; on the other hand, first downscaling images and then compressing them is a common method for larger compression rate (Bruckstein et al., 2003). Direct image compression methods perform poorly under extremely large compression rate, and are always combined with image rescaling for high compression rate of high-resolution images. In this work, we demonstrate the combination between IRN and lossless as well as lossy compression methods for better lossy compression performance.

2.4 Invertible Neural Network

The invertible neural network (INN) (Behrmann et al., 2019; Chen et al., 2019; Dinh et al., 2015, 2017; Grathwohl et al., 2019; Kingma & Dhariwal, 2018; Kumar et al., 2020; Kobyzev et al., 2020) is usually used for generative models. The invertible transformation of INN f_θ specifies the generative process $x = f_\theta(z)$ given a latent variable z, while the inverse mapping f_θ^{-1} enables explicit computation for the density of the model distribution, i.e. $p_X(x) = p_Z(f^{-1}(x)) \left| \det J f^{-1}(x) \right|$. Therefore, it is possible to use the maximum likelihood method for stable training of INN. The flexibility for modeling distributions allows INN to be applied in many variational inference tasks as well (Berg et al., 2018; Kingma et al., 2016; Rezende & Mohamed, 2015). Also, due to the strict invertibility, INN has been used to learn representations without information loss (Jacobsen et al., 2018), which has been applied in the super-resolution task as a feature embedding module (Li et al., 2019; Zhu, 2019).

Several prior works apply INN for tasks with paired data (x, y). For example, Ardizzone et al. (2019) deal with real-world inverse problems from medicine and astrophysics with INN. And Asim et al. (2020) leverage INN as effective priors at inverse problems including denoising, compressive sensing, and inpainting. Ren et al. (2020) further analyze INN as deep inverse models for generic inverse problems with four benchmarking tasks. Besides, conditional generation with INN, where the invertible modeling between x and z is conditioned on y, has also been explored and analyzed, such as in the task of image colorization (Ardizzone et al., 2019) and super resolution (Lugmayr et al., 2020). Different from these tasks considering unidirectional generation, image degradation–restoration is bidirectional, i.e. both generating y given x and the inverse reconstruction of x are required. Therefore these models are unsuitable for our task, and we propose to model information loss in this task with INN. On the other hand, INN has been applied to conduct image-to-image translation (van der Ouderaa and Worrall, 2019). They consider the paired domain (X, Y) rather than paired data, which is also different from our scenario.

The computational architecture of INN is specially designed to enable invertibility. For example, the mainstream architecture of INN is composed of coupling layers proposed in (Dinh et al., 2015, 2017). In this architecture, INN consists of several invertible blocks. For the computation of the l-th block, different from conventional neural networks that directly apply neural network transformation on the input h^l as $f(h^l)$, the input $h^l \in \mathbb{R}^{N \times H \times W \times C}$ is first split into h_1^l, h_2^l, usually along the channel axis so that $h_1^l \in \mathbb{R}^{N \times H \times W \times C_1}, h_2^l \in \mathbb{R}^{N \times H \times W \times C_2}, C_1 + C_2 = C$, and the following additive

transformations are applied (Dinh et al., 2015):

$$
\begin{aligned}
h_1^{l+1} &= h_1^l + \phi(h_2^l), \\
h_2^{l+1} &= h_2^l + \eta(h_1^{l+1}),
\end{aligned}
\tag{1}
$$

where ϕ, η are functions parameterized by neural networks, e.g. convolutional neural networks. There is no restriction for ϕ, η. The output of the block is the concatenation of the two parts, i.e. $[h_1^{l+1}, h_2^{l+1}]$, which will be the input to the $(l+1)$-th block. The strictly inverse transformation is easily computed given the output:

$$
\begin{aligned}
h_2^l &= h_2^{l+1} - \eta(h_1^{l+1}), \\
h_1^l &= h_1^{l+1} - \phi(h_2^l),
\end{aligned}
\tag{2}
$$

For stronger expression ability, the following computation is often leveraged (Dinh et al., 2017):

$$
\begin{aligned}
h_1^{l+1} &= h_1^l \odot \exp(\psi(h_2^l)) + \phi(h_2^l), \\
h_2^{l+1} &= h_2^l \odot \exp(\rho(h_1^{l+1})) + \eta(h_1^{l+1}), \\
h_2^l &= (h_2^{l+1} - \eta(h_1^{l+1})) \odot \exp(-\rho(h_1^{l+1})), \\
h_1^l &= (h_1^{l+1} - \phi(h_2^l)) \odot \exp(-\psi(h_2^l)).
\end{aligned}
\tag{3}
$$

This is the basic component of mainstream INNs that enforces the invertibility of the computation, and the expressive ability of such kind of architecture has been theoretically studied (Teshima et al., 2020). There are also other choices for INN architectures. For example, Behrmann et al. (2019); Chen et al. (2019) prove that for the commonly used residual neural network architecture $y = f_\theta(x) + x$, when the spectral norm of the residual function f_θ is restricted under 1, this computation is invertible and therefore can be used as a kind of INN. On the other hand, Lu et al. (2021) further proposes implicit normalizing flows, in which the computation of INN is implicitly defined by solving an equation. We will design our invertible architecture based on the typical coupling-layer-based invertible blocks, i.e. Eqs. (1,3), and task-related considerations in Sect. 3.3.1.

3 Methods

In this section, we first formally present the general mathematical formulation of the image degradation–restoration problem in Sect. 3.1. Then we describe the invertible modeling framework of this bidirectional problem in Sect. 3.2. As for the specific model, we start from image rescaling and elaborate on the specific invertible architecture and training methods for IRN in Sect. 3.3. Then we show the adaptation of IRN to the similar decolorization–colorization task in Sect. 3.4. Finally, we propose to combine the invertible framework with existing degradation methods with an

instantiation of the combination between image rescaling and compression in Sect. 3.5.

3.1 Mathematical Formulation of Image Degradation–Restoration

The basic formulation of the image degradation–restoration problem can be described as:

$$\min_{\theta} \quad \sum_{x} \mathcal{L}\left(x, \mathcal{U}\left(\mathcal{D}(x; \theta); \theta\right)\right),$$

$$\text{s.t.} \quad y = \mathcal{D}(x; \theta) \in Y(x), \forall x, \tag{4}$$

where x is the original image, e.g. HR image for the image rescaling task, $\mathcal{D}$ and $\mathcal{U}$ are respectively the degradation and restoration models parameterized by θ, e.g. downscaling and upscaling of image rescaling, $\mathcal{L}$ is a criterion justifying the quality of recovered images, $y = \mathcal{D}(x; \theta)$ is the model-degraded image, and $Y(x)$ denotes the target set of valid degraded images given x, e.g. visually valid LR images given the HR image x for the image rescaling task. When $\mathcal{D}$ is a given mapping without parameters to optimize, the problem of learning $\mathcal{U}$ only resorts to a typical restoration problem, e.g. image super-resolution. In contrast, in the degradation–restoration problem, $\mathcal{D}$ is also learned and contributes to a better restoration.

In many tasks, although we do not have the explicit expression of $Y(x)$, it is much easier to obtain a valid degraded image in this set. For example, typical interpolation methods (e.g. Bicubic) could produce visually valid LR images for the image rescaling tasks. As for the rescaling and decolorization–colorization tasks in this paper, we instantiate the constraint in (4) by narrowing the set around a given sample. Specifically, let $y_{\text{guide}}(x)$ denote an available degraded image, e.g. an LR image downscaled by a typical interpolation method which well demonstrates what is a visually valid LR image as a sample in $Y(x)$. We instantiate $Y(x)$ by $Y_{\text{guide}}(x) = \{y \mid \|y - y_{\text{guide}}(x)\| < \epsilon\}$. So in practice only one valid degraded image $y_{\text{guide}}(x)$ is required and the original problem turns into:

$$\min_{\theta} \quad \sum_{x} \mathcal{L}\left(x, \mathcal{U}\left(\mathcal{D}(x; \theta); \theta\right)\right),$$

$$\text{s.t.} \quad \|\mathcal{D}(x; \theta) - y_{\text{guide}}(x)\| < \epsilon. \tag{5}$$

In Sect. 3.2.2, this constraint will be further relaxed and formulate a guidance loss in practice.

Now we have described the basic settings of image degradation–restoration. The problem formulation under our invertible framework will be illustrated in the following sections.

3.2 Specification of Invertible Modeling

3.2.1 Formulation of Invertible Framework

As described in the Introduction, we model the bidirectional degradation and restoration from the perspective of invertible bijective transformation. Figure 1 illustrates the sketch of our invertible framework. To model lost information during degradation, we introduce an auxiliary latent random variable z, and leverage an invertible neural network to bijectively transform the distribution of x to the joint distribution of a pre-specified distribution $p(z)$ and the distribution of model-degraded image y. Then the distribution of lost contents is transformed to $p(z)$ together with the generation of y. As described in the introduction, we note that for any random vector with a density (i.e. $z' \sim p(z'|y)$), there exists a bijection f_y such that $f_y(z') \sim N(0, I)$ (Hyvärinen & Pajunen, 1999); therefore for easier modeling and sampling of $p(z)$ without handling conditions, we choose image-agnostic $z \sim p(z)$ as an additional desideratum, which will be enforced by distribution matching. In this way, the distribution of lost contents is captured by our model without preserving image-specific lost contents or z, and a random sample of z' from $p(z)$ in company with the degraded image y could reconstruct a image x' with reasonable lost contents by the inverse function of our invertible model. Let f_{θ} denote the parameterized bijective transformation. Then the degradation procedure of our model is expressed as $(y, z) = f_{\theta}(x)$, where y is the output degraded image. Correspondingly, the restoration procedure is $x' = f_{\theta}^{-1}(y, z')$, where $z' \sim p(z)$. As z' is random, the restored image x' is also random. This defines the restoration distribution $p_{\theta}(x|y)$, representing the uncertainty over all possible original images that could yield y. The randomness of z corresponds to the randomness of reasonable x in $p_{\theta}(x|y)$. Note that this inverse transformation will mix y and z' so that the generation process is still dependent on the image-specific information.

The invertible modeling framework is particularly suitable for the degradation–restoration problem under a measure-theoretic point of view, in that it has the unique advantage of being *cyclically compatible* (Liu et al. 2021, Def. 2.1). This means the model-defined restoration distribution $p_{\theta}(x|y)$ and degradation distribution $p_{\theta}(y|x)$ always come from the same joint distribution of (x, y). Since the degradation distribution $p_{\theta}(y|x) = \delta_{f_{\theta}^{y}(x)}(y)$ ($f_{\theta}^{y}(x)$ denotes the y-part of the output of $(y, z) = f_{\theta}(x)$) is a Dirac delta distribution, the restoration distribution $p_{\theta}(x|y)$ is compatible with it if and only if it is supported within the preimage set of the degradation transformation f_{θ}^{y}, i.e. $(f_{\theta}^{y})^{-1}(\{y\}) := \{x \mid f_{\theta}^{y}(x) = y\}$ (Liu et al. 2021, Thm. 2.6). Due to the invertibility of f_{θ}, for any $z' \in \mathbb{R}^{K}$, the restored image $f_{\theta}^{-1}(y, z')$ is always in the preimage set since $f_{\theta}^{y}(f_{\theta}^{-1}(y, z')) = y$. In this way, the

Fig. 1 Illustration of the invertible modeling framework for the degradation–restoration problem. In the forward degradation procedure, the image x is transformed to a valid degraded image y and an image-agnostic latent variable z through a parameterized invertible function $f_\theta(\cdot)$; in the inverse upscaling procedure, a randomly drawn z combined with y are transformed to restore image x through the inverse function $f_\theta^{-1}(\cdot)$

Forward Degradation f_θ

$z \sim N(0, I_{K=d-m})$

Invertible Neural Network

$x \sim q(x)$

$x \in \mathbb{R}^d$

Inverse Restoration f_θ^{-1}

$y = f_\theta^y(x)$

$y \in \mathbb{R}^m$
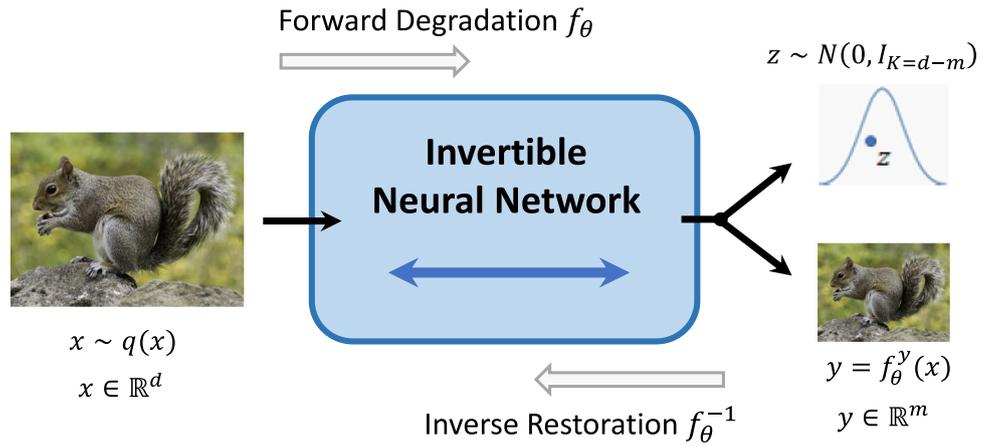
model only needs to focus on learning the distribution over all possible original images without worrying about conflicting with the degradation process.

With invertible modeling, the problem formulation is described as:

$$\min_\theta \quad \sum_x \mathbb{E}_{z \sim p(z)}\left[\mathcal{L}\left(x, f_\theta^{-1}\left([f_\theta^y(x), z]\right)\right)\right],$$

$$\text{s.t.} \quad \|f_\theta^y(x) - y_{\text{guide}}\| < \epsilon,$$
$$\{f_\theta^z(x)\}_x \sim p(z),$$

(6)

where f_θ^y and f_θ^z denote the transformations whose outputs correspond to y and z of the output of $f_\theta(x)$ respectively. In Sect. 3.2.2, the constraint regarding distributions will be relaxed and formulate a distribution loss in practice.

3.2.2 Realization of Invertible Framework

Our invertible framework specifies a correspondence between the distributions of the original image x and the degraded image y, as well as the image-agnostic distribution $p(z)$ of the latent variable z. To realize this framework, we should train the invertible model denoted by f_θ. This subsection introduces the general training objectives for our invertible models, while some adaptions will be detailed for specific tasks in Sects. 3.3 and 3.4. The training objectives are to drive the above relations and match our requirements, i.e. solve (6). We will make the constrained optimization problem (6) practical by reforming it as jointly optimizing three objective terms as introduced below.

Reconstruction As described in Sect. 3.2, our invertible framework is under the context of distribution. Therefore it is not for the correspondence between the point x and y if z is not specified. Given a image $x^{(n)}$, the model-degraded image $f_\theta^y(x^{(n)})$ will be restored by our model with the image-agnostic latent variable $z \sim p(z)$, resulting in $f_\theta^{-1}(f_\theta^y(x^{(n)}), z)$ which also follows a distribution. We hope to restrict this distribution around the original image so that

the image can be validly recovered by the model using any sample of z from $p(z)$. This arbitrariness would inversely encourage the disentanglement between z and y in the forward process as well. To achieve this, we encourage the reconstructed image with any random sample z to match the original $x^{(n)}$, leading to the reconstruction loss which minimizes the expected difference over all original images:

$$L_{\text{recon}}(\theta) := \sum_{n=1}^N \mathbb{E}_{z \sim p(z)}[\ell_{\mathcal{X}}(x^{(n)}, f_\theta^{-1}(f_\theta^y(x^{(n)}), z))],$$

(7)

where $\ell_{\mathcal{X}}$ is a difference metric on $\mathcal{X}$, e.g. the L_1 or L_2 loss. We estimate the expectation w.r.t z by one random sample from $p(z)$ each evaluation in practice. This loss corresponds to the objective in (6).

Guidance As described in Sect. 3.1, we hope to generate a valid degraded image belonging to a target set, whose expression is not explicitly known, but we can instantiate it as a constraint w.r.t. the distance to guiding degraded images. We relax this constraint as a loss added in the objective, which encourages the model-degraded images to resemble guiding images. Let $y_{\text{guide}}^{(n)}$ denote this guiding image (for example, an LR image generated by the Bicubic interpolation for the image rescaling task). The guidance loss is expressed as:

$$L_{\text{guide}}(\theta) := \sum_{n=1}^N \ell_{\mathcal{Y}}(y_{\text{guide}}^{(n)}, f_\theta^y(x^{(n)})),$$

(8)

where $\ell_{\mathcal{Y}}$ is a difference metric on $\mathcal{Y}$, e.g. the L_1 or L_2 loss. This kind of objective was also adopted in literatures (Kim et al., 2018; Sun and Chen, 2020).

Distribution Matching The third part of the training objective is to match the distribution of latent variable z and original images. We first describe our notations for the distributions. We denote the data distribution of original images as $q(x)$, which is available through the sample cloud $\{x^{(n)}\}_{n=1}^N$.

Note that when traversing over this sample cloud, $\{y^{(n)}\}_{n=1}^{N}$ generated by our model also form a sample cloud of a distribution. We use the push-forward distribution $f_{\theta\,\#}^{y}[q](y)$ to denote this distribution of y, which represents the distribution of the transformed random variable $y = f_{\theta}^{y}(x)$ with $x \sim q(x)$. We define the push forward distribution $f_{\theta\,\#}^{z}[q](z)$ in the same way. Similarly, the inversely reconstructed images compose a sample cloud $\{f_{\theta}^{-1}(y^{(n)}, z^{(n)})\}_{n=1}^{N}$ following a distribution, where $z^{(n)} \sim p(z)$ is a randomly drawn latent variable. As $z \sim p(z)$ is to be independent from y, we have $(y^{(n)}, z^{(n)}) \sim f_{\theta\,\#}^{y}[q](y)\,p(z)$. Therefore, we can denote the distribution of reconstructed images as $f_{\theta\,\#}^{-1}\big[f_{\theta\,\#}^{y}[q](y)\,p(z)\big](x)$.

Our model should enforce $z \sim p(z)$ to be image-agnostic and match the model-reconstructed distribution towards data distribution. This corresponds to the constraint on the distribution in (6). Therefore we relax the constraint as a loss added in the objective as well, and introduce the distribution matching loss to achieve these two goals:

$$L_{\mathrm{distr}}(\theta) := L_{\mathcal{P}}\big(f_{\theta\,\#}^{-1}\big[f_{\theta\,\#}^{y}[q](y)\,p(z)\big](x), q(x)\big), \qquad (9)$$

where $L_{\mathcal{P}}$ is a difference metric of distributions. The distribution matching loss directly pushes the model-reconstructed images to lie on the manifold of true original images, which matches the distribution and enables the recovered images to be more realistic (note that the reconstruction loss only restrict them around the original images). At the same time, it drives the independence of $z \sim p(z)$ from y in the forward transformation. This is because if f_{θ} is invertible, the distribution matching holds on $\mathcal{X}$ if and only if it holds on $\mathcal{Y} \times \mathcal{Z}$ in the asymptotic case, i.e. $f_{\theta\,\#}^{-1}\big[f_{\theta\,\#}^{y}[q](y)\,p(z)\big](x) = q(x)$ is equivalent to $f_{\theta\,\#}^{y}[q](y)\,p(z) = f_{\theta\#}[q](y, z)$. In this way, the loss also drives the coupled distribution $f_{\theta\#}[q](y, z)$ from the forward transformation towards the decoupled distribution $f_{\theta\,\#}^{y}[q](y)\,p(z)$, realizing the matching of independent $z \sim p(z)$.

As for the probability metric $L_{\mathcal{P}}$, we can employ the JS divergence due to the high-dimensionality and unknown density function in our problem. We estimate the loss as:

$$
\begin{aligned}
L_{\mathrm{distr}}(\theta) =\,& \mathrm{JS}(f_{\theta\,\#}^{-1}\big[f_{\theta\,\#}^{y}[q](y)\,p(z)\big](x), q(x)) \\
=\,& \frac{1}{2}\max_{T}\Big\{\mathbb{E}_{q(x)}\big[\log\sigma(T(x))\big] \\
& + \mathbb{E}_{x' \sim f_{\theta\,\#}^{-1}\big[f_{\theta\,\#}^{y}[q](y)\,p(z)\big](x')}\big[\log\big(1 - \sigma(T(x'))\big)\big]\Big\} \\
& + \log 2 \\
=\,& \frac{1}{2}\max_{T}\Big\{\mathbb{E}_{q(x)}\big[\log\sigma(T(x))\big] \\
& + \mathbb{E}_{(y,z) \sim f_{\theta\,\#}^{y}[q](y)\,p(z)}\big[\log\big(1 - \sigma(T(f_{\theta}^{-1}(y, z)))\big)\big]\Big\} \\
& + \log 2 \\
\approx\,& \frac{1}{2N}\max_{T}\sum_{n}\Big\{\log\sigma(T(x^{(n)}))
\end{aligned}
$$

$$+ \log\Big(1 - \sigma(T(f_{\theta}^{-1}(f_{\theta}^{y}(x^{(n)}), z^{(n)})))\Big)\Big\} + \log 2, \qquad (10)$$

where σ is the sigmoid function, $T : \mathcal{X} \to \mathbb{R}$ is a function on $\mathcal{X}$ and $\sigma(T(\cdot))$ is regarded as the discriminator in GAN literatures (Goodfellow et al., 2014). The "$\approx$" is due to Monte Carlo estimation: $\{z^{(n)}\}_{n=1}^{N}$ are i.i.d. samples from $p(z)$ and $\{x^{(n)}\}_{n=1}^{N} \sim q(x)$. In practice, we can parameterize the function T with a neural network T_{ϕ}, and thus $\max_{T}$ amounts to $\max_{\phi}$. We can follow the same way as GANs to optimize θ and ϕ so that the JS divergence is minimized.

3.3 Model for Image Rescaling

As for specific models, we start from image rescaling in this section. We develop Invertible Rescaling Network (IRN) as the instantiation model of our inverible modeling framework for image rescaling, and we will describe the specific invertible architecture and training methods of IRN. We also present the algorithms for downscaling and upscaling in our IRN model in Algorithms 1, 2 as an example to better demonstrate the input, output, and procedure of our invertible framework. Note that in practice the HR image x and LR image y will be quantized to 8-bit representation, as will be indicated in Sect. 3.3.1. We omit this detail in the algorithm description and treat the domain as $\mathbb{R}$.

Algorithm 1 Downscaling of IRN

Input: HR image $x \in \mathbb{R}^{H \times W \times C}$, scale size s, model $f_{\theta,s}$
Output: LR image $y \in \mathbb{R}^{\frac{H}{s} \times \frac{W}{s} \times C}$
1: Calculate $(y, z) = f_{\theta,s}(x)$
2: **return** y

Algorithm 2 Upscaling of IRN

Input: LR image $y \in \mathbb{R}^{H \times W \times C}$, scale size s, model $f_{\theta,s}$
Output: HR image $x \in \mathbb{R}^{sH \times sW \times C}$
1: Randomly sample $z \sim p(z)$, $z \in \mathbb{R}^{H \times W \times (s^2-1)C}$
2: Calculate $x = f_{\theta,s}^{-1}(y, z)$
3: **return** x

3.3.1 Invertible Architecture

Figure 2 illustrates the architecture of our proposed IRN, which is basically composed of stacked *Downscaling Modules* consisting of one *Haar Transformation* and several *InvBlocks*. Each *Downscaling Module* will reduce the spatial resolution by $2\times$. The overall architecture is invertible given that each component is invertible.

The Haar Transformation In each *Downscaling Module*, a Haar Transformation is first applied to equip the model with a certain inductive bias for splitting low- and
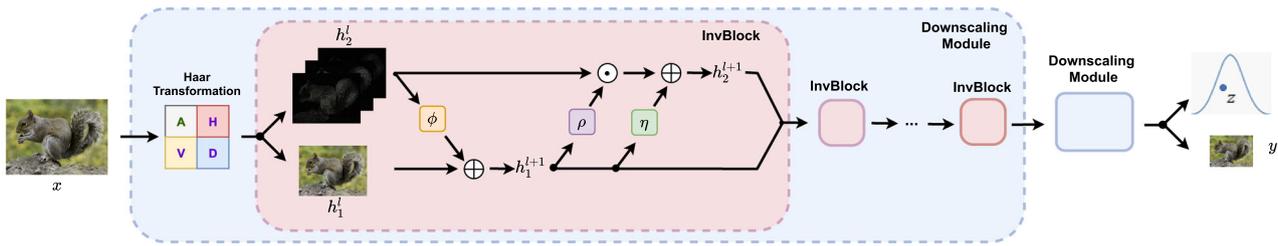
Fig. 2 Illustration of our Invertible Rescaling Network (IRN) as the instantiation model of our invertible modeling framework. The invertible architecture is composed of Downscaling Modules, in which

InvBlocks are stacked after a Haar Transformation. Each Downscaling Module reduces the spatial resolution by 2×. The exp(·) of ρ is omit

high-frequency contents, which are approximately preserved and lost contents during image downscaling respectively. The Haar Transformation, which is an invertible wavelet transformation, will decompose the input into a low-pass representation and three directions of high-frequency coefficients (Ardizzone et al., 2019). Specifically, given the input raw image or feature maps with height H, width W and channel C, a tensor of shape $(\frac{1}{2}H, \frac{1}{2}W, 4C)$ is produced, where the first C slices are the low-pass representation equivalent to the Bilinear interpolation downscaling, and the other three groups of C slices correspond to the high-frequency residual in the vertical, horizontal and diagonal directions respectively. With the help of the Haar Transformation, the model could effectively separate low- and high-frequency information, which benefits the following generation of y and transformation from x_H to z. And the spatial resolution is reduced by 2× after the Haar Transformation.

InvBlock InvBlocks are the main components for the target invertible transformations. Given that the input has been split into low- and high-frequency components by the Haar Transformation, we introduce InvBlocks based on the coupling layer architecture described in Eqs. (1,2,3), whose two branches (i.e. the split of h_1^l and h_2^l in Eq. (1)) correspond to these two components respectively. The transformation would further polish the input representations for the generation of a suitable LR image as well as an independent and properly distributed latent representation for lost information. As for the detailed computation, considering the importance of shortcut connection in image scaling tasks (Lim et al., 2017; Wang et al., 2018), we employ the additive transformation (Eq. 1) for the low-frequency part h_1^l, and the enhanced affine transformation (Eq. 3) for the high-frequency part h_2^l to enhance the model capacity. This also equips the model with a certain inductive bias for the generation of LR images with the low-frequency part going straight through, and could stabilize the training of IRN. The details of the InvBlock architecture are illustrated in Fig. 2, except that the exp(·) operation after function ρ is omitted here.
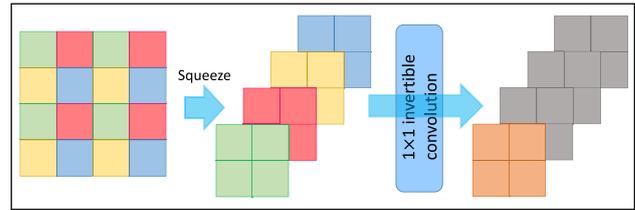
Fig. 3 Illustration of the learnable downsampling module (2× example). It consists of a squeeze operation to downscale the spatial resolution by N times and a 1×1 invertible convolution to transform the squeezed $N \times N$ elements

We employ a densely connected convolutional block, which has demonstrated its effectiveness for image scaling tasks in (Wang et al., 2018), to parameterize the transformation functions $\phi(\cdot)$, $\eta(\cdot)$, $\rho(\cdot)$. To avoid numerical explosion due to the exp(·) function, we employ a centered sigmoid function and a scale term after function $\rho(\cdot)$.

Quantization The outputs of our model are floating-point values, while the common image formats such as RGB are quantized to 8-bit representation. To enable storage compatibility, we adopt a rounding operation as the quantization module on the generated LR image. The quantized LR image is saved by PNG format and used for upscaling. However, the nondifferentiable property of quantization poses challenges for training with back-propagation. To overcome the obstacle, we apply the Straight-Through Estimator (Bengio et al., 2013) to calculate the gradients for the quantization module. The notation for quantization is omitted in the following for simplicity.

3.3.2 Scale-Flexible and Efficient Implementation

There could be further improvements over the architecture to adapt IRN to more scales or more computation efficiency. Specifically, we will introduce the learnable downsampling module and improvement on computational efficiency to enable scale-flexible and efficient implementation.

Learnable Downsampling Although the Haar Wavelet Transformation is able to serve for downsampling and splitting

high- and low-frequency contents well, stacking multiple transformations can only rescale images by the scales that are the power of two. This largely restricts the rescaling scope for our model. To enable more scales, such as $3\times$, we propose to leverage a learnable downsampling layer to replace Haar Transformation in the architecture. It consists of a squeeze operation and one 1×1 invertible convolution.

As shown in Fig. 3, the squeeze operation downscales the spatial resolution for a certain scale N by squeezing spatial elements into channels. Then, a 1×1 invertible convolution is applied to transform the squeezed $N \times N$ elements before InvBlocks. 1×1 invertible convolution is first proposed in GLOW (Kingma & Dhariwal, 2018) for channel permutation. Different from their purpose, we expect it to learn to split low- and high-frequency contents under arbitrary scales and adapt the following InvBlocks better. The Haar Transformation can be viewed as a special case of this downsampling module under $2\times$ scale, as it provides a fixed rather than learnable prior. For this module, we provide a prior for extracting low-frequency in initialization by setting parameters of the 1×1 invertible convolution in order that the first channel after transformation is the average of $N \times N$ elements, while the other channels are the identity transformation to enable the invertibility.

We denote the IRN model with learnable downsampling as IRN_{LD}.

Fractional scaling factors In real applications, there would be fractional scaling factors. We can deal with them by combining IRN and traditional interpolation methods. Specifically, for the scaling factor s_1, we choose IRN with scaling factor $s_2 = [s_1]$ and rescale HR images with interpolation (e.g. Bicubic) by scale $\frac{s_2}{s_1}$ and $\frac{s_1}{s_2}$ before and after passing them into IRN respectively. This has been demonstrated in recent work as well (Xing et al., 2022).

Improving Computation Efficiency We note that the architecture that stacks multiple Downscaling Modules containing one downsampling module and multiple InvBlocks suffers from much-increased FLOPs during computation. This is because InvBlocks in the previous Downscaling Modules other than the last one will apply convolution operations on tensors with larger spatial resolution, which significantly increases computational cost. To further improve computation efficiency, we propose to modify the architecture to first apply downsampling modules (e.g. multiple Haar Transformation or learnable downsampling) and then go through multiple InvBlocks. This enables the convolution operations to be applied on smaller resolutions, which could largely reduce the FLOPs and runtime under a similar amount of parameters.

We denote the IRN model under this architecture as IRN_E. It differs from IRN only when IRN stacks multiple Downscaling Modules.

3.3.3 Training Objectives

The training objectives of IRN mainly follow the reconstruction (Eq. 7), guidance (Eq. 8), and distribution matching (Eq. 9) to realize the invertible framework as described in Sect. 3.2.2. For image rescaling, the reconstruction and guidance is adapted as HR reconstruction and LR guidance correspondingly, which means calculating L_{recon} between reconstructed and original HR images and calculating L_{guide} between model-generated LR images and LR images generated by the Bicubic interpolation methods, respectively. Based on the above objectives, we can optimize our IRN model by minimizing the combination of the three losses, which relaxes the constrained problem (6) into an unconstrained one. However, as an issue in practice, we find it difficult to directly do the optimization due to the unstable training process of GANs (Arjovsky & Bottou, 2017). Therefore, we propose to adopt a weakened but more stable surrogate loss for the distribution matching as a pre-training stage, forming a two-stage training procedure.

As explained in Sect. 3.2.2, the distribution matching on $\mathcal{X}$ has the same asymptotic effect as on $\mathcal{Y} \times \mathcal{Z}$, i.e. $L_{\mathcal{P}}(f^y_{\theta\#}[q](y)\, p(z), f_{\theta\#}[q](y, z))$. Our surrogate loss considers partial distribution matching on $\mathcal{Z}$, i.e. $L_{\mathcal{P}}(p(z), f^z_{\theta\#}[q](z))$, which is more flexible as the density function of $p(z)$ is available. We choose cross entropy (CE) as a more stable distribution metric for minimization:

$$L'_{distr}(\theta) := CE(f^z_{\theta\#}[q](z), p(z))$$
$$= -\mathbb{E}_{f^z_{\theta\#}[q](z)}[\log p(z)] = -\mathbb{E}_{q(x)}[\log p(z = f^z_\theta(x))]. \tag{11}$$

Note that the maximum likelihood estimation (MLE) $\max_\theta \mathbb{E}_{q(x)}[\log f^{-1}_{\theta\#}[p_{y,z}](x)]$ commonly used in related INN-based generative models (Ardizzone et al., 2019; Dinh et al., 2015, 2017; Kingma & Dhariwal, 2018) is however not applicable to our model, since it requires a joint distribution $p(y, z)$ with tractable density function on the (y, z) end, while we only have a distribution $p(z)$ on z.[4] Therefore we can only leverage a stable but weakened surrogate loss.

Our pre-training stage will minimize the following total objective, and we call IRN as this trained model:

$$L_{IRN} := \lambda_1 L_{recon} + \lambda_2 L_{guide} + \lambda_3 L'_{distr}, \tag{12}$$

where $\lambda_1, \lambda_2, \lambda_3$ are coefficients for balancing different loss terms.

---

[4] MLEs corresponding to minimizing $KL(q(x|y), f^{-1}_\theta(y, \cdot)_{\#}[p_z](x))$ or $KL\left(q(x), \left(\mathbb{E}_{f^y_{\theta\#}[q](y)}[f^{-1}_\theta(y, \cdot)]\right)_{\#}[p_z](x)\right)$ are also impossible, since the pushed-forward distributions are only supported on a lower-dimensional manifold (dimension of z) in $\mathcal{X}$ so their densities are not well-defined (i.e., the densities are a.e. zero in $\mathcal{X}$ and are infinite on the manifold).

After the pre-training, we adopt the trained model as the initialization and restore the full distribution matching loss L_{distr} based on JS divergence for the training objective. Additionally, as L_{distr} encourages reconstructed HR images to be more realistic, we also add a perceptual loss (Johnson et al., 2016) L_{percp} on $\mathcal{X}$ to further enhance the perceptual quality. Instead of pixels, the perceptual loss measures the difference between two images on their semantic features, which are extracted by pre-trained deep learning models (e.g. VGG). There are several variants of the perceptual loss which mainly differ from the feature positions (Ledig et al., 2017; Wang et al., 2018), and we adopt the variant proposed in Wang et al. (2018).

Therefore, the second stage minimizes the following total objective, and we call the model as IRN+:

$$L_{\text{IRN+}} := \lambda_1 L_{\text{recon}} + \lambda_2 L_{\text{guide}} + \lambda_3 L_{\text{distr}} + \lambda_4 L_{\text{percp}}. \quad (13)$$

3.4 Model for Image Decolorization–Colorization

Image decolorization–colorization is a commonly seen task (Xia et al., 2018; Ye et al., 2020) and is another instantiation of bidirectional degradation–restoration problem, in which color information in the channel dimension is lost. The core idea of our problem formulation is the same as Fig. 1, which transforms the distribution of image-specific lost information into an image-agnostic Gaussian distribution. Some adaptation of the specific model to fit this task is illustrated as the following.

3.4.1 Architecture

The basic architecture is similar to Fig. 2. Different from splitting low- and high-frequency contents as image rescaling, we need to split grayscale and color information, and produce a grayscale image while capturing the distribution of color information here. Therefore, we need to replace the downsampling module with a graying module. We directly leverage the YCbCr color space representation of the image to split the information in the channel. Then these two branch of information (i.e. Y and CbCr) go through InvBlocks as introduced previously. We denote this model as IRN$_{\text{color}}$.

3.4.2 Training Objectives

We also leverage the three components for the objective, i.e. guidance loss (Eq. 8), reconstruction loss (Eq. 7), and distribution matching loss (Eq. 9). In particular, for the guidance loss, we adapt it as a Grayscale Guidance, in which the Y channel under YCbCr representation of the image is leveraged as the guidance. The reconstruction loss is to compute the difference between reconstructed images and original ones. For distribution matching, we choose the sta-

ble cross entropy introduced in Sect. 3.3.3 here, because the human perception of color is less sensitive and the unstable perceptual-driven loss is not necessary for good results. Besides, because colorization has more diverse results than upscaling, to stabilize and improve our training for the reconstruction of original color images, we will consider an alternative choice to only encourage the most probable point of latent variable z in its distribution rather than the whole distribution to perfectly reconstruct original images. That is, when z follows the standard Gaussian distribution, we set $z = 0$ rather than a random sample in the inverse computation. For more discussion about this please refer to Sect. 4.2.5.

3.5 Combination of Image Rescaling and Compression

Our invertible framework jointly models degradation and restoration as an invertible bijective transformation. In real applications, some parts of degradation operations are not always available to adapt with restoration, e.g. for convenience. For example, the widely used image compression follows general standards, and formats such as PNG and JPEG are the most commonly used ones with well-established support in most digital devices. Therefore, we propose the combination of our invertible framework and restoration from existing degradation methods for wider applications.

Specifically, we consider the instantiation of the combination between image rescaling and compression, which is also a common method for a higher compression rate of high-resolution images (Bruckstein et al., 2003), because direct image compression methods perform poorly under an extremely large compression rate. In this work, we demonstrate the combination between IRN and lossless as well as lossy compression methods for better lossy compression performance.

Note that it is also possible to directly generalize the invertible framework for image compression with some additional efforts. Please refer to (Wang et al., 2020) for the preliminary attempt.

3.5.1 Methods

For lossless image compression methods, LR images can be encoded without information loss, therefore IRN can be directly combined with them, i.e. directly compress the downscaled LR images generated by IRN.

For existing lossy image compression methods, there would be inevitable information loss during encoding, i.e. additional degradation caused by the lossy compression. So directly combining IRN with them , e.g. first compress LR images of IRN and then directly pass compressed images to IRN, may go against the principle of modeling lost infor-

Table 1 Quantitative evaluation results (PSNR/SSIM) of different downscaling and upscaling methods for image reconstruction on benchmark datasets: Set5, Set14, BSD100, Urban100, and DIV2K validation set

Downscaling and upscaling	Scale	Param	Set5	Set14	BSD100	Urban100	DIV2K
Bicubic & Bicubic	2×	/	33.66/0.9299	30.24/0.8688	29.56/0.8431	26.88/0.8403	31.01/0.9393
Bicubic & SRCNN (Dong et al., 2015)	2×	57.3K	36.66/0.9542	32.45/0.9067	31.36/0.8879	29.50/0.8946	35.60/0.9663
Bicubic & EDSR (Lim et al., 2017)	2×	40.7M	38.20/0.9606	34.02/0.9204	32.37/0.9018	33.10/0.9363	35.12/0.9699
Bicubic & RDN (Zhang et al., 2018b)	2×	22.1M	38.24/0.9614	34.01/0.9212	32.34/0.9017	32.89/0.9353	–
Bicubic & RCAN (Zhang et al., 2018a)	2×	15.4M	38.27/0.9614	34.12/0.9216	32.41/0.9027	33.34/0.9384	–
Bicubic & SAN (Dai et al., 2019)	2×	15.7M	38.31/0.9620	34.07/0.9213	32.42/0.9028	33.10/0.9370	–
TAD & TAU (Kim et al., 2018)	2×	–	38.46/–	35.52/–	36.68/–	35.03/–	39.01/–
CNN-CR & CNN-SR (Li et al., 2018)	2×	–	38.88/–	35.40/–	33.92/–	33.68/–	–
CAR & EDSR (Sun and Chen, 2020)	2×	51.1M	38.94/0.9658	35.61/0.9404	33.83/0.9262	35.24/0.9572	38.26/0.9599
IRN (ours)	2×	1.66M	**43.99/0.9871**	**40.79/0.9778**	**41.32/0.9876**	**39.92/0.9865**	**44.32/0.9908**
Bicubic & Bicubic	4×	/	28.42/0.8104	26.00/0.7027	25.96/0.6675	23.14/0.6577	26.66/0.8521
Bicubic & SRCNN (Dong et al., 2015)	4×	57.3K	30.48/0.8628	27.50/0.7513	26.90/0.7101	24.52/0.7221	–
Bicubic & EDSR (Lim et al., 2017)	4×	43.1M	32.62/0.8984	28.94/0.7901	27.79/0.7437	26.86/0.8080	29.38/0.9032
Bicubic & RDN (Zhang et al., 2018b)	4×	22.3M	32.47/0.8990	28.81/0.7871	27.72/0.7419	26.61/0.8028	–
Bicubic & RCAN (Zhang et al., 2018a)	4×	15.6M	32.63/0.9002	28.87/0.7889	27.77/0.7436	26.82/0.8087	30.77/0.8460
Bicubic & ESRGAN (Wang et al., 2018)	4×	16.3M	32.74/0.9012	29.00/0.7915	27.84/0.7455	27.03/0.8152	30.92/0.8486
Bicubic & SAN (Dai et al., 2019)	4×	15.7M	32.64/0.9003	28.92/0.7888	27.78/0.7436	26.79/0.8068	–
TAD & TAU (Kim et al., 2018)	4×	–	31.81/–	28.63/–	28.51/–	26.63/–	31.16/–
CAR & EDSR (Sun and Chen, 2020)	4×	52.8M	33.88/0.9174	30.31/0.8382	29.15/0.8001	29.28/0.8711	32.82/0.8837
IRN (ours)	4×	4.35M	**36.19/0.9451**	**32.67/0.9015**	**31.64/0.8826**	**31.41/0.9157**	**35.07/0.9318**
Bicubic & Bicubic	8×	/	24.40/0.6580	23.10/0.5660	23.67/0.5480	20.74/0.5160	23.70/0.6387
Bicubic & SRCNN (Dong et al., 2015)	8×	57.3K	25.33/0.6900	23.76/0.5910	24.13/0.5660	21.29/0.5440	–
Bicubic & EDSR (Lim et al., 2017)	8×	–	26.96/0.7762	24.91/0.6420	24.81/0.5985	22.51/0.6221	25.50/–
Bicubic & RCAN (Zhang et al., 2018a)	8×	15.8M	27.31/0.7878	25.23/0.6511	24.98/0.6058	23.00/0.6452	–
Bicubic & SAN (Dai et al., 2019)	8×	15.8M	27.22/0.7829	25.14/0.6476	24.88/0.6011	22.70/0.6314	–
TAD & TAU (Kim et al., 2018)	8×	–	–	–	–	–	26.77/–
IRN (ours)	8×	11.1M	**31.20/0.8736**	**28.40/0.7698**	**27.49/0.7239**	**26.67/0.7947**	**30.29/0.8280**

Bold italic values indicate the best results under different scales

For our method, differences on average PSNR/SSIM from different z samples are less than 0.02. We report the mean result over 5 draws

mation in the proposed invertible framework. Additional restoration for such degradation is required for good performance.

To mitigate this problem, we propose to leverage an additional module to partially restore the lost information by lossy compression methods. Specifically, downscaled images of IRN will first be compressed by lossy compression methods, e.g. JPEG, and the compressed image will go through a Compression Restore Module (CRM) before being passed to IRN. CRM is taken as a neural network model, whose input is the compressed LR image with degradation and output is the LR image restored from the degradation caused by lossy compression. This module is trained to restore lost information of the given compression method, which is similar to many methods considering the unidirectional restoration task. We will elaborate on the detailed architecture and evaluate the compression performance in the next section. The combination of IRN and CRM is the instantiation model of our proposed combination of invertible framework and restoration from existing degradation methods.

4 Experiments

4.1 Datasets and Settings

Our experiments include three parts: image rescaling, image decolorization–colorization, as well as the combination between image rescaling and compression. For the training of all tasks, we employ the widely used DIV2K (Agustsson & Timofte, 2017) image restoration dataset to train our models. It contains 800 high-quality 2K resolution training images and 100 validation images. Besides, for the first two tasks, we evaluate our model on 4 additional standard datasets, i.e. the Set5 (Bevilacqua et al., 2012), Set14 (Zeyde et al., 2010), BSD100 (Martin et al., 2001), and Urban100 (Huang et al., 2015); and for the third task, we also evaluate our model on the widely used Kodak dataset (Franzen, 1999). For image rescaling, following the setting in (Lim et al., 2017), we quantitatively evaluate the peak noise-signal ratio (PSNR) and SSIM (Wang et al., 2004) on the Y channel of images represented in the YCbCr (Y, Cb, Cr) color space. We also evaluate LPIPS (Zhang et al., 2018b), PI (Blau et al., 2018), and FID (Heusel et al., 2017) as quantitative metrics of perceptual evaluation. For the other two tasks, we evaluate PSNR and SSIM on the RGB three-channel color space.

For image rescaling, we train and compare our IRN model in $2\times$, $4\times$ and $8\times$ downscaling scale with 1, 2, and 3 downscaling modules respectively. Each downscaling module has 8 InvBlocks and downscales the original image by $2\times$. The transformation functions $\phi(\cdot)$, $\eta(\cdot)$, $\rho(\cdot)$ in InvBlocks are parameterized by a densely connected convolutional block, which is referred to as Dense Block in Wang et al. (2018).

Table 2 Quantitative evaluation results (PSNR/SSIM) of different $3\times$ image downscaling and upscaling methods on benchmark datasets: Set5, Set14, BSD100, Urban100, and DIV2K validation set

Downscaling and upscaling	Scale	Param	Set5	Set14	BSD100	Urban100	DIV2K
Bicubic & Bicubic	$3\times$	/	30.39/0.8682	27.55/0.7742	27.21/0.7385	24.46/0.7349	26.95/0.8556
Bicubic & SRCNN (Dong et al., 2015)	$3\times$	57.3K	32.75/0.9090	29.30/0.8215	28.41/0.7863	26.24/0.7989	30.48/0.9117
Bicubic & EDSR (Lim et al., 2017)	$3\times$	43.7M	34.65/0.9280	30.52/0.8462	29.25/0.8093	28.80/0.8653	34.17/0.9476
Bicubic & RDN (Zhang et al., 2018b)	$3\times$	22.3M	34.71/0.9296	30.57/0.8468	29.26/0.8093	28.80/0.8653	34.13/0.9484
Bicubic & RCAN (Zhang et al., 2018a)	$3\times$	15.6M	34.74/0.9299	30.65/0.8482	29.32/0.8111	29.09/0.8702	34.44/0.9499
Bicubic & SAN (Dai et al., 2019)	$3\times$	15.7M	34.75/0.9300	30.59/0.8476	29.33/0.8112	28.93/0.8671	34.30/0.9494
IRN$_{LD}$ (ours)	$3\times$	3.14M	*37.94/0.9586*	*34.64/0.9313*	*33.80/0.9306*	*33.45/0.9470*	*37.33/0.9586*

Bold italic values indicate the best results

For our model, differences on average PSNR/SSIM of different samples for z are less than 0.02. We report the mean result

Table 3 Quantitative perceptual evaluation results of different 4× image downscaling and upscaling methods on benchmark datasets: Set5, Set14, BSD100, Urban100, and DIV2K validation set

(Metrics)	PSNR/SSIM LPIPS/PI/FID	PSNR/SSIM LPIPS/PI/FID	PSNR/SSIM LPIPS/PI/FID	PSNR/SSIM LPIPS/PI/FID	PSNR/SSIM LPIPS/PI/FID
Downscaling & Upscaling	Set5	Set14	BSD100	Urban100	DIV2K
Bicubic & ESRGAN	32.74/0.9012	29.00/0.7915	27.84/0.7455	27.03/0.8152	30.92/0.8486
	0.169/6.095/53.87	0.273/5.342/74.75	0.358/5.190/93.1	0.198/5.041/24.41	0.256/5.274/15.91
Bicubic & ESRGAN+	30.57/0.8561	26.39/0.7054	25.52/0.6618	24.48/0.7420	28.17/0.7759
	0.076/3.842/27.61	0.133/*2.944*/55.17	0.165/*2.494*/49.00	0.126/*3.740*/20.75	0.115/*3.202*/13.56
IRN (ours)	***36.19/0.9451***	***32.67/0.9015***	***31.64/0.8826***	***31.41/0.9157***	***35.07/0.9318***
	0.078/4.195/33.88	*0.123*/3.635/35.96	0.166/3.069/42.11	0.084/4.021/***9.13***	0.119/3.804/***5.78***
IRN+ (ours)	33.59/0.9147	29.97/0.8444	28.94/0.8189	28.24/0.8684	32.24/0.8921
	0.031/3.382/11.15	***0.067/2.952/32.38***	***0.074/2.398/22.06***	***0.055/3.541/13.00***	***0.054/3.240/7.90***

For LPIPS, PI, and FID, lower is better. The best result is in bold italic, and the second best result is in italic

For experiments of IRN$_{LD}$ model in 3× scale, we use one downscaling module with learnable downsampling and 12 InvBlocks. For experiments of IRN$_E$ model in 4× scale, we use one downscaling module with 16 InvBlocks (downscaling first). We use Adam optimizer (Kingma & Ba, 2015) with $\beta_1 = 0.9, \beta_2 = 0.999$ to train our model. The mini-batch size is set to 16. The input HR image is randomly cropped into 144×144 and augmented by applying random horizontal and vertical flips. In the pre-training stage, the total number of iteration is $500K$, and the learning rate is initialized as 2×10^{-4} where halved at $[100k, 200k, 300k, 400k]$ mini-batch updates. The hyper-parameters in Eq. (12) are set as $\lambda_1 = 1, \lambda_2 = s^2, \lambda_3 = 1$, where s denotes the scale. After pre-training, we finetune our model for another $200K$ iterations as described in Sect. 3.3.3. The learning rate is initialized as 1×10^{-4} and halved at $[50k, 100k]$ iterations. We set $\lambda_1 = 0.01, \lambda_2 = s^2, \lambda_3 = 1, \lambda_4 = 0.01$ in Eq. (13) and pre-train the discriminator for 5000 iterations. The discriminator is similar to Ledig et al. (2017), which contains eight convolutional layers with 3×3 kernels, whose numbers increase from 64 to 512 by a factor of 2 every two layers, and two dense layers with 100 hidden units.

For image decolorization–colorization, the graying module has 8 InvBlocks. The hyper-parameters are set as $\lambda_1 = 1, \lambda_2 = 9, \lambda_3 = 1$. Other optimizers and iteration settings are the same as image rescaling.

For combination with image compression, we leverage the IRN$_{2\times}$ model trained in image rescaling task and further finetune it for $100K$ iterations in the rescaling task by adding a random noise on the generated LR images during upscaling in training, in order to make the model more robust to possible changes on LR images due to compression and restoration. The model for Kodak is additionally finetuned for $2.5K$ iterations on Kodak. We train a compression restore module (CRM) for each compression ratio of JPEG. The CRM contains 8 residual in residual dense blocks (RRDB) proposed in (Wang et al., 2018), and is trained by a L_2 loss on reconstructed LR images and LR images before compression. The optimizer and iteration settings are the same as IRN.

4.2 Image Rescaling

4.2.1 Evaluation on Reconstructed HR Images

In this section, we present the quantitative and qualitative performance of HR images reconstructed by our model and other downscaling and upscaling methods. Two kinds of baselines are considered: (1) downscaling with Bicubic interpolation and upscaling with state-of-the-art SR models trained with this downscaling kernel (Dai et al., 2019; Dong et al., 2015; Lim et al., 2017; Wang et al., 2018; Zhang et al., 2018b, a); (2) downscaling with upscaling-optimal models (Kim et al., 2018; Li et al., 2018; Sun and Chen, 2020) and

Fig. 4 Qualitative results of upscaling the $4\times$ downscaled images. IRN recovers rich details, leading to both visually pleasing performance and high similarity to the original images. IRN+ produces even sharper and more realistic details. See the appendix for more results
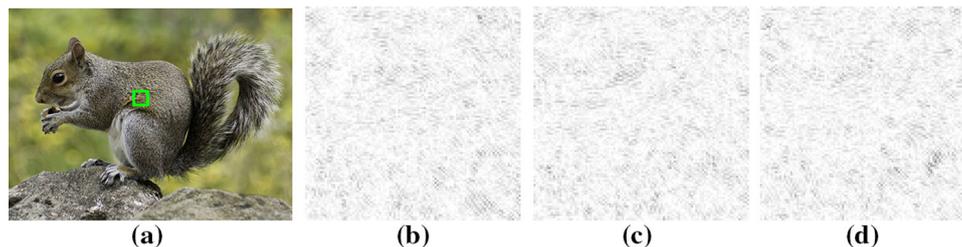
Fig. 5 Visualisation of the difference of upscaled HR images from multiple draws of z. **a** original image; **b–d** HR image differences of three z samples from another common z sample. Darker color means larger difference. It shows that the differences are high-frequency noises in high-frequency regions without a typical texture

upscaling with corresponding SR models. For the notations, we identify the downscaling and upscaling methods respectively for baselines while use IRN or IRN+ as a whole to denote our invertible model for the bidirectional tasks; and following our notation, we use ESRGAN to represent the pre-trained PSNR-driven model of Wang et al. (2018) while ESRGAN+ for their GAN-based perceptual-driven model. In addition, the influence of different samples of z on our reconstructed HR images and the effectiveness of different types of loss in the pre-training stage are investigated.

Quantitative Results As shown in Table 1, IRN significantly outperforms the state-of-the-art baseline models regarding quantitative evaluation PSNR and SSIM in all datasets. Although upscaling-optimal downscaling methods largely enhance the reconstruction performance of SR models compared with Bicubic interpolation due to the unification of bidirectional tasks, they still suffer from the ill-posed problem caused by information loss and therefore the results are hardly satisfying. Contrarily, by modeling the lost informa-

tion with invertibility, IRN significantly boosts the PSNR with about 4–5 dB, 2–3 dB, and 3–4 dB on each dataset under $2\times$, $4\times$, and $8\times$ scale respectively compared with the state-of-the-art results, where the improvement is up to 5.94 dB. The PSNR results indicate an exponential reduction of information loss due to its logarithmic computation, which is consistent with the significant improvement of SSIM. The results of IRN+ are in the appendix because it is visual perception oriented. IRN_{LD} extends IRN to more flexible downscaling and upscaling scales. Table 2 demonstrates the significant improvement of IRN_{LD} on $3\times$ scale as well, with about 3-5 dB improvement on the PSNR metric compared with other methods.

It is noteworthy that IRN achieved the best results with a relatively small amount of parameters. When upscaling with SR models, it requires more than 15M parameters for better results, while the model sizes of our IRN are only 1.66M, 4.35M, and 11.1M in the scale $2\times$, $4\times$, and $8\times$. It indicates

the lightweight property and high efficiency of our proposed invertible model.

We also quantitatively evaluate the perceptual performance as shown in Table 3. LPIPS and PI are full-reference and no-reference methods for perceptual evaluation of each image respectively, and FID is the metric for the perceptual similarity between two groups of images. We compare IRN and IRN+ with the representative PSNR-driven model ESRGAN and perceptual-driven model ESRGAN+, and the results demonstrate significant improvements of our models. Particularly, IRN+ with full distribution matching and perceptual loss achieves the best result considering both PSNR/SSIM and perceptual indexes, which also accords with the qualitative results below.

Qualitative Results When it comes to qualitative evaluation, we visually demonstrate the details of the upscaled images by different methods. Figure 4 demonstrates the better visual quality and fidelity of our IRN and IRN+ model compared with previous state-of-the-art methods. IRN could recover richer details, while IRN+ further produces sharper and more realistic images, leading to their pleasing visual quality. For instance, IRN and IRN+ are the only models that are able to reconstruct the 'Comic' image with the complicated textures on the headwear and necklace, as well as the sharp and realistic fingers. Previous perceptual-driven models such as ESRGAN+, however, would produce unreasonable and unpleasing details, leading to great dissimilarity. The better results of our models owe to the modeling of information loss, as well as the distribution matching and perceptual loss for IRN+. More visual results are in the appendix.

Visualisation on the Influence of z We further investigate the influence of random z. As described in Sect. 3.2, different samples of $z \sim p(z)$ aim to only focus on the randomness of reasonable high-frequency contents. Visually, we calculate and visualize the difference between different draws of z in Fig. 5. It shows that only a tiny noisy distinction without typical textures is observed in high-frequency regions, which are almost imperceptible if combined with low-frequency contents. Quantitatively, different samples of z result in the PSNR difference that is less than 0.02 dB for each image, which also indicates that the randomness mainly lies in high-frequency noise. These results indicate that our models have learned the knowledge to restore meaningful lost high-frequency contents while embedding imperceptible noises into the randomness of distribution.

Additionally, we test our model with out-of-distribution samples to verify its effectiveness and sensitivity. Our models are trained with $p(z)$ being an isotropic Gaussian distribution, and we test IRN and IRN+ by inversely passing $(y, \alpha z)$ to obtain x_α with the control of the scale α of sampled $z \sim p(z)$. Note that the probability density for samples with $\alpha < 1$ is still large for the Gaussian distribution, e.g. the point of $z = 0$ has the largest probability density, and therefore the

reconstruction should still be valid if distribution matching is fully realized. As shown in Fig. 6, IRN+ could validly reconstruct HR images when the sampled z lie in areas with a large probability density or with small disturbance, and more noisy textures and degradations would appear when there is a larger deviation from the original distribution. This indicates that IRN+ fully realizes the distribution matching for $p(z)$ and is robust to mild deviation. On the other hand, IRN without the full distribution matching objective fails to validly reconstruct HR images when the scale $\alpha \neq 1$, which indicates that it only learns to validly reconstruct images by z around the areas with a large density of training samples rather than the full distribution. This demonstrates the effectiveness of our full distribution matching objective.

Analysis on the Losses We also conduct analysis experiments for the losses of Eqs. (8, 7, 11), which is shown in Tables 4, 5 and 6. We can see from Table 4 that when the LR guidance takes the L_2 loss while the HR reconstruction is the L_1 loss, IRN gets the best training performance. The underlying explanation is that our forward procedure aims to learn a valid downscaling transformation that is beneficial to the inverse upscaling, rather than exactly the Bicubic downscaling, so the L_2 loss that is less sensitive to minor changes from the guidance would be more suitable; while the goal of our inverse procedure is to accurately reconstruct the original HR image, thus the L_1 loss encouraging more pixelwise similarity is profitable. The results also demonstrate the improvement brought by our surrogate partial distribution matching loss (Eq. (11)), which acts on the marginal distribution on $\mathcal{Z}$ to encourage the forward distribution learning.

As described in Sect. 4.1, our default weights for HR reconstruction and LR guidance loss are $\lambda_1 = 1$ and $\lambda_2 = s^2$ in order to keep the losses on the same scale. To further justify the choice, we study the weights with different scales of ratios. We conduct analysis experiments with IRN in $4\times$ scale. The original weights are $\lambda_1 = 1$, $\lambda_2 = 16$, we largely increase or decrease the weight for LR guidance, i.e. $\lambda_2 = 160$ or $\lambda_2 = 1.6$. The evaluation results on image reconstruction are shown in Table 5. It shows that the reconstruction quality is quite robust to the ratio between HR reconstruction and LR guidance, and the original weights that keep the losses on the same scale achieve the best results. We also compare the images downscaled by IRN trained by different loss weights with those downscaled by Bicubic to verify the validity of LR images. The results are in Table 6. It shows that the LR similarity is strongly correlated with the ratio of LR guidance loss, and the larger the loss is, the more similar LR images are. When $\lambda_2 = 16$, it is enough to keep the LR images valid due to the strong similarity (PSNR$>$40, SSIM$>$0.99), and setting $\lambda_2 = 160$ could improve the LR similarity but not HR reconstruction quality. When $\lambda_2 = 1.6$, however, the LR similarity is significantly dropped, and there could be slight artifacts on the LR images on the validation

(a) IRN

(b) IRN+
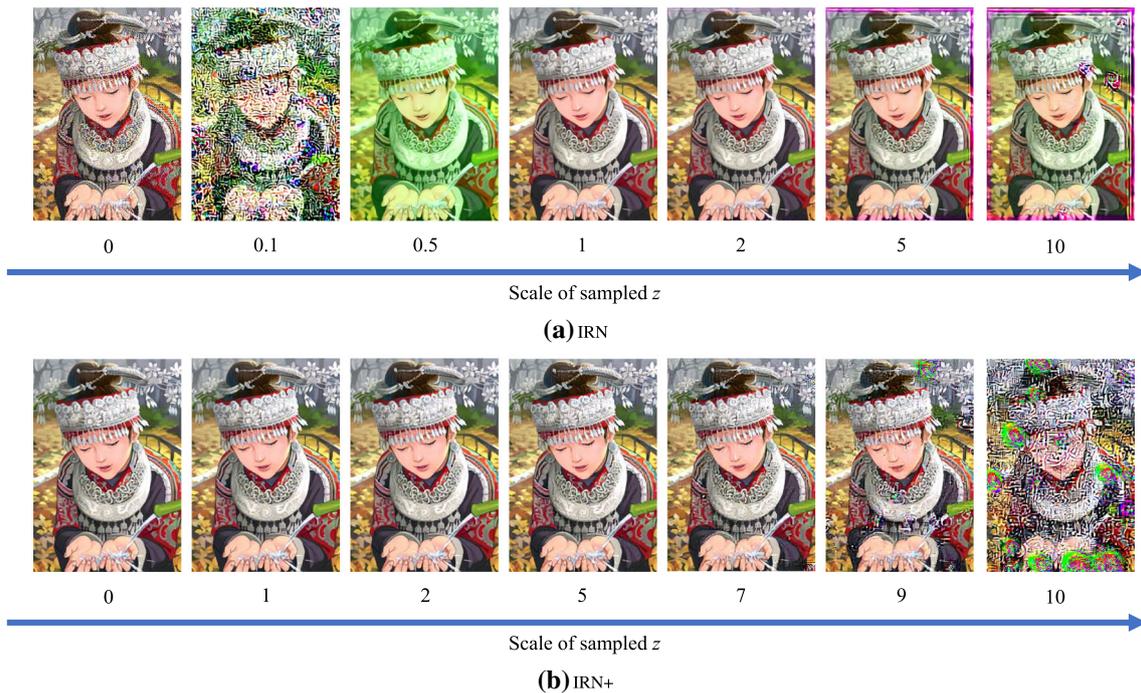
Fig. 6 Results of HR images by IRN and IRN+ with out-of-distribution samples of z. We train z with an isotropic Gaussian distribution, and illustrate upscaling results when scaling z sampled from the isotropic Gaussian distribution

Table 4 Analysis results (PSNR/SSIM) of training IRN with L_1 or L_2 LR guide and HR reconstruction loss, with/without partial distribution matching loss, on Set5, Set14, BSD100, Urban100 and DIV2K validation sets with scale $4\times$

L_{guide}	L_{recon}	$L_{distr'}$	Set5	Set14	BSD100	Urban100	DIV2K
L_1	L_1	Yes	34.75/0.9296	31.42/0.8716	30.42/0.8451	30.11/0.8903	33.64/0.9079
L_1	L_2	Yes	34.93/0.9296	31.76/0.8776	31.01/0.8562	30.79/0.8986	34.11/0.9116
L_2	L_1	Yes	*36.19*/0.9451	*32.67/0.9015*	*31.64/0.8826*	*31.41/0.9157*	*35.07/0.9318*
L_2	L_2	Yes	35.93/0.9402	32.51/0.8937	31.64/0.8742	31.40/0.9105	34.90/0.9308
L_2	L_1	No	36.12/*0.9455*	32.18/0.8995	31.49/0.8808	30.91/0.9102	34.90/0.9308

Bold italic values indicate the best results

Table 5 Analysis results (PSNR/SSIM) of training IRN with different loss weights for HR reconstruction and LR guidance loss, for image reconstruction on Set5, Set14, BSD100, Urban100 and DIV2K validation sets with scale $4\times$

λ_1	λ_2	Set5	Set14	BSD100	Urban100	DIV2K
1	16	*36.19/0.9451*	*32.67/0.9015*	*31.64/0.8826*	*31.41/0.9157*	*35.07/0.9318*
1	160	35.94/0.9439	32.32/0.8961	31.40/0.8757	31.26/0.9121	34.81/0.9276
1	1.6	35.72/0.9391	32.06/0.8863	31.14/0.8676	30.52/0.8992	34.47/0.9221

Bold italic values indicate the best results

Table 6 Analysis results (PSNR/SSIM) between the LR images downscaled by IRN trained by different loss weights and by Bicubic on Set5, Set14, BSD100, Urban100 and DIV2K validation sets with scale $4\times$

λ_1	λ_2	Set5	Set14	BSD100	Urban100	DIV2K
1	16	44.60/0.9964	42.47/0.9928	43.24/0.9923	41.28/0.9916	44.37/0.9933
1	160	50.14/0.9988	47.57/0.9977	48.62/0.9976	47.46/0.9977	50.06/0.9980
1	1.6	34.25/0.9820	34.00/0.9764	35.59/0.9755	33.40/0.9720	35.59/0.9782

Table 7 SSIM results between the images downscaled by IRN and by Bicubic on the Set5, Set14, BSD100, Urban100 and DIV2K validation sets

Scale	Set5	Set14	BSD100	Urban100	DIV2K
$2\times$	0.9957	0.9936	0.9936	0.9941	0.9945
$4\times$	0.9964	0.9927	0.9923	0.9916	0.9933
$8\times$	0.9958	0.9926	0.9918	0.9879	0.9919

datasets, which hamper the HR reconstruction. As a result, the reconstruction performance of $\lambda_2 = 1.6$ is the worst. Therefore, keeping the losses on the same scale as the original setting is the best choice for our model.

4.2.2 Evaluation on Downscaled LR Images

To verify the validity of our downscaling, we evaluate the quality of IRN-downscaled LR images. Table 7 demonstrates the similarity index SSIM between our LR images and Bicubic-based LR images. It quantitatively shows that the images are extremely similar to each other. More figures in the appendix illustrate the visual similarity between the images, demonstrating the proper and valid visual perception of our LR images similar to Bicubic-based ones. Therefore, the downscaling of IRN can perform as well and valid as the guidance Bicubic interpolation.

4.2.3 Ablation on Invertibility

To further demonstrate the effectiveness of the proposed invertible framework, we conduct ablation comparisons by simply leveraging IRN architecture to upscale Bicubic-downscaled images (we denote the model as IRN-U), by training existing SR models to upscale IRN-downscaled images (IRN model is pre-trained and we denote it as IRN-D* here), and by joint training separate IRN-D and IRN-U models in an encoder-decoder framework.

For the first experiment, we pad z by 0 to keep the dimension in order to train the model. As shown in Table 8, simply training the architecture of IRN on Bicubic-downscaled images fails to reach a satisfactory performance. This illustrates that our improvement is not from network architecture or capacity.

For the second experiment, we train the ESRGAN model (Wang et al., 2018) (one of the state-of-the-art SR models with codes, we use its PSNR-driven model) on LR images downscaled by pre-trained IRN. We train a small model with similar parameters with IRN (we denote it as ESRGAN$_s$), and a model with original capacity. As shown in Table 8, without our invertible framework, the performance will drop much even if more parameters are used.

For the third experiment, we train IRN-D & IRN-U and IRN under different amount of parameters. As shown in Table 8, without invertibility, separate IRN-D & IRN-U models achieve much lower performance, especially when the amount of parameters is small. This illustrates the improvement by our invertible framework, as well as the highly efficient utilization of parameters that enables lightweight models.

4.2.4 Computation Efficiency

The previous results demonstrate the lightweight property of IRN considering parameters. We further compare detailed computation efficiency between IRN and other methods with available open-source code. We demonstrate the results of $2\times$ and $4\times$ here.

We calculate the FLOPs and RunTime for models to downscale or upscale images, setting the size of high-resolution images as 1920×1080, and running on one Tesla-P100 GPU. All methods are implemented in PyTorch, except CAR (Sun and Chen, 2020) which is partially in CUDA code. As shown in Table 9, IRN demonstrates overall computation efficiency.

IRN$_E$ could improve computation efficiency for larger scales that require multiple downscaling modules in IRN. As shown in Table 9, in $4\times$ scale, IRN$_E$ could reduce about 50% of FLOPS and RunTime. Table 10 shows the performance of IRN$_E$. There might exists a balance between computation efficiency and performance.

4.2.5 Discussion on Randomness of z

In this subsection, we would like to have some discussions on the randomness of z and the current implementation of our model.

First, when there is information loss, restoration would certainly contain randomness due to the uncertainty. To fully model the information loss from the perspective of statistical modeling, we have to leverage a random latent variable z and learn the bijective distribution transformation between the distribution of x and the joint distribution of y and z, and the randomness of z corresponds to randomness of reasonable lost contents.

As for our IRN model, which is in the pre-training stage without the full distribution matching objective and is different from IRN+, it does not fully model the full distribution, but only around the density of training samples of z (see the paragraph **Visualisation on the Influence of** z in Section 4.2.1). So for this model, an alternative to not consider the randomness, e.g. taking $z = 0$ which has the largest probability density in the Gaussian distribution, may be still valid considering the density on this point, as shown in Table 11. Note that this only encourages the point with the largest probability density to recover an HR image, and it degrades the

Table 8 Ablation study on the invertibility

Downscaling and upscaling	Param	Set5	Set14	BSD100	Urban100	DIV2K
IRN	4.35M	36.19/0.9451	32.67/0.9015	31.64/0.8826	31.41/0.9157	35.07/0.9318
Bicubic & IRN-U	4.35M	32.03/0.8930	28.54/0.7800	27.52/0.7336	25.97/0.7801	30.37/0.8358
IRN-D* & ESRGAN$_s$	4.35+4.47M	35.14/0.9365	31.47/0.8807	30.61/0.8588	29.62/0.8903	33.71/0.9150
IRN-D* & ESRGAN	4.35+16.3M	35.87/0.9432	32.31/0.8963	31.37/0.8775	30.98/0.9116	34.75/0.9288
IRN-D & IRN-U (tiny)	1.09M	34.87/0.9283	31.34/0.8721	30.47/0.8510	29.39/0.8790	33.49/0.9061
IRN (tiny)	1.09M	**35.64/0.9402**	**32.00/0.8891**	**31.12/0.8698**	**30.36/0.8994**	**34.41/0.9230**
IRN-D & IRN-U (small)	2.18M	35.88/0.9432	32.31/0.8959	31.31/0.8755	30.65/0.9060	34.63/0.9267
IRN (small)	2.18M	**36.04/0.9432**	**32.49/0.8955**	**31.45/0.8764**	**31.13/0.9102**	**34.84/0.9279**
IRN-D & IRN-U	4.35M	35.93/0.9418	32.57/0.8974	31.41/0.8750	31.31/0.9124	34.77/0.9265
IRN	4.35M	**36.19/0.9451**	**32.67/0.9015**	**31.64/0.8826**	**31.41/0.9157**	**35.07/0.9318**
IRN-D & IRN-U (large)	8.70M	36.21/0.9450	32.84/0.9008	31.57/0.8772	31.59/0.9169	35.05/0.9297
IRN (large)	8.70M	**36.32/0.9461**	**32.86/0.9032**	**31.74/0.8845**	**31.59/0.9179**	**35.18/0.9330**

Bold values indicate the better results under the same amount of parameters
Quantitative results (PSNR/SSIM) for 4× scale on the Set5, Set14, BSD100, Urban100 and DIV2K validation sets are reported

Table 9 Computation efficiency results of different methods for downscaling or upscaling images by different scales, with the HR image size 1920×1080

Downscaling and upscaling method	Scale	Param (Down+Up)	FLOPs (Down)	FLOPS (Up)	RunTime (ms) (Down)	RunTime (ms) (Up)
Bicubic & RCAN (Zhang et al., 2018a)	2×	15.4M	/	7.96×10^{12}	/	2188
Bicubic & ESRGAN (Wang et al., 2018)	2×	16.7M	/	9.31×10^{12}	/	2251
CAR & EDSR (Sun and Chen, 2020)	2×	10.7M + 40.73M	2.12×10^{12}	2.11×10^{13}	228	2476
IRN (ours)	2×	1.67M	8.66×10^{11}	8.66×10^{11}	344	347
Bicubic & RCAN (Zhang et al., 2018a)	4×	15.6M	/	2.07×10^{12}	/	633
Bicubic & ESRGAN (Wang et al., 2018)	4×	16.7M	/	2.33×10^{12}	/	593
CAR & EDSR (Sun and Chen, 2020)	4×	9.89M + 43.09M	8.97×10^{11}	6.52×10^{12}	107	706
IRN (ours)	4×	4.36M	1.21×10^{12}	1.21×10^{12}	515	521
IRN$_E$ (ours)	4×	5.37M	6.97×10^{11}	6.97×10^{11}	264	269

Table 10 Quantitative results (PSNR/SSIM) of IRN and IRN$_E$ for 4× scale on the Set5, Set14, BSD100, Urban100 and DIV2K validation sets

Downscaling and upscaling	Param	Set5	Set14	BSD100	Urban100	DIV2K
IRN	4.35M	36.19/0.9451	32.67/0.9015	31.64/0.8826	31.41/0.9157	35.07/0.9318
IRN$_E$	5.37M	35.52/0.9393	32.14/0.8935	31.17/0.8777	30.65/0.9107	34.53/0.9282

Table 11 Quantitative evaluation results (PSNR/SSIM) of IRN and IRN ($z=0$) on benchmark datasets: Set5, Set14, BSD100, Urban100, and DIV2K validation set

Downscaling and upscaling	Scale	Param	Set5	Set14	BSD100	Urban100	DIV2K
IRN	4×	4.35M	36.19/0.9451	32.67/0.9015	31.64/0.8826	31.41/0.9157	35.07/0.9318
IRN ($z=0$)	4×	4.35M	36.23/0.9463	32.70/0.9019	31.63/0.8832	31.22/0.9137	35.04/0.9321

Table 12 Quantitative results (PSNR) of different decolorization–colorization methods for image reconstruction on the Set5, Set14, BSD100, Urban100 and DIV2K validation sets

Method	Set5	Set14	BSD100	Urban100	DIV2K
Baseline (Kim et al., 2018)	19.12	21.14	24.21	23.29	21.10
TAD-G & TAU-C	35.22	32.67	32.73	30.98	36.63
IRN$_{color}$ (ours)	*40.86*	*36.78*	*42.43*	*38.77*	*42.65*

Bold italic values indicate the best results

bijective transformation between two distributions into the bijective transformation between two points (i.e. it does not model the distribution or consider randomness by choosing only one preferred point in the distribution). In this setting, the losses for IRN may correspond to the losses to match data points. The results show that our invertible model is valid for this degraded condition as well.

However, our general goal is to model the full distribution as IRN+, which is a more general case and has more potential. For example, the reconstructed HR images should have many different possible realistic high-frequency details, and our general framework has the potential to model such diversity according to the randomness of z.

In our current experiments, because the training dataset does not contain enough such diversity information, e.g. different perceptible high-frequency textures of similar low-frequency contents, and one of our main training objectives during pre-training is to encourage the pixel level similarity of reconstructed and original HR images, the diversity with different z mainly lies in the randomness of imperceptible high-frequency details, and the PSNR scores are similar. In potential future applications, it is possible for realistic diversities with proper datasets.

In this work, we present our general invertible framework that can model the full distribution of lost information, which may have more potential future applications.

4.3 Invertible Image Decolorization–Colorization

As described in Sect. 3.4, the proposed invertible framework and model can be extended to other bidirectional tasks, such as image decolorization–colorization. In this section, we present experiments of the extended model under this task, to illustrate the generalization ability of our model.

We compare our model with TAD Gray & TAU Color (Kim et al., 2018) and invertible grayscale (Xia et al., 2018), which all follow the encoder-decoder framework. Because Xia et al. (2018) has different training settings and datasets, we train and test their model under a similar setting as theirs on the DIV2K dataset that is rescaled to 256×256. We also test our model that is trained on the original DIV2K dataset on this rescaled dataset.

As shown in Table 12, IRN$_{color}$ can perfectly reconstruct the original color images from grayscale ones, with most RGB PSNR results above 40 dB, which indicates that the

Table 13 Quantitative results (PSNR/SSIM) of different decolorization–colorization methods for image reconstruction on the DIV2K validation set that is rescaled to 256×256

Method	Param	DIV2K_256×256
Invertible Grayscale	7.42M	31.52/0.9475
IRN$_{color}$ (ours)	1.41M	*37.27/0.9800*

Bold italic values indicate the best results

img_012 from Urban100 — Grayscale — Reconstructed (31.80 / 0.9640)

img_0882 from DIV2K — Grayscale — Reconstructed (44.79 / 0.9826)
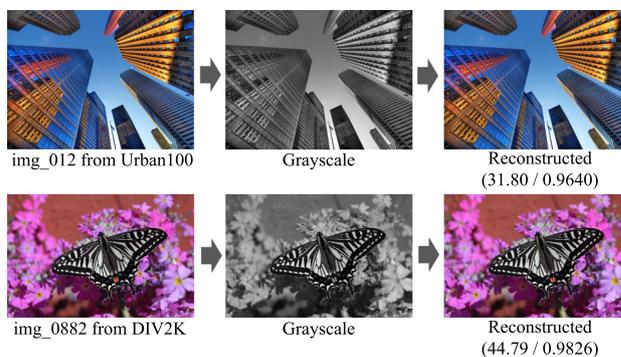
Fig. 7 Qualitative demonstration of decolorization–colorization by IRN$_{color}$

img_0846 from DIV2K — Invertible Grayscale (28.97 / 0.9344) — IRN$_{color}$ (ours) (40.35 / 0.9954)

img_0833 from DIV2K — Invertible Grayscale (29.34 / 0.9449) — IRN$_{color}$ (ours) (38.77 / 0.9923)
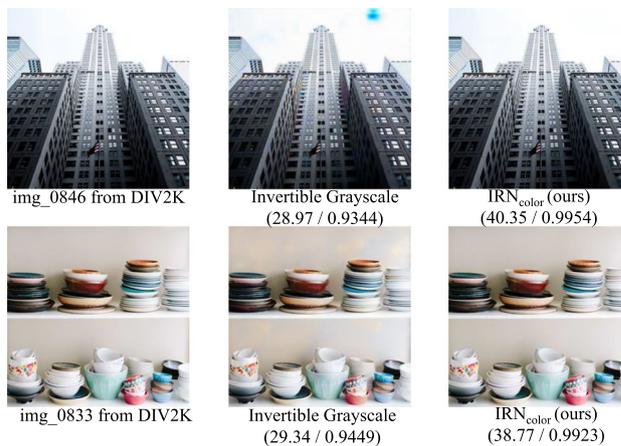
Fig. 8 Qualitative comparison of colorization reconstruction for grayscale images between different methods

reconstructed images are almost the same as original ones. And compared with TAD Gray & TAD Color (Kim et al., 2018), IRN$_{color}$ demonstrates the significant improvement of the quality of reconstructed images, indicating the advantage of our invertible framework.

Table 14 Comparison results of combination between image rescaling and lossless image compression methods on average RGB PSNR and total storage size of DIV2K validation set

Method	Scale	PSNR (dB)	Storage (MB)
PNG	/	/	470
FLIF	/	/	294
JPEG (Q=20)	/	29.59	16.2
Bicubic&ESRGAN+PNG	4×	29.47	32.4 (+100.0%)
Bicubic&ESRGAN+FLIF	4×	29.47	22.4 (+38.3%)
JPEG (Q=32)	/	31.11	21.7
CAR&EDSR+PNG	4×	31.09	30.2 (+39.2%)
CAR&EDSR+FLIF	4×	31.09	21.3 (– 1.8%)
JPEG (Q=57)	/	32.94	31.4
IRN+PNG	4×	32.95	34.9 (+11.1%)
IRN+FLIF	4×	32.95	*28.7 (– 8.6%)*
JPEG (Q=96)	/	40.70	122
IRN+PNG	2×	40.87	131 (+7.3%)
IRN+FLIF	2×	40.87	*108 (– 11.5%)*
JPEG (Q=14)	/	28.36	13.07
IRN+PNG	8×	28.50	*9.16 (– 29.9%)*
IRN+FLIF	8×	28.50	*7.68 (– 41.2%)*

Bold italic means our results that are better than JPEG with the similar PSNR
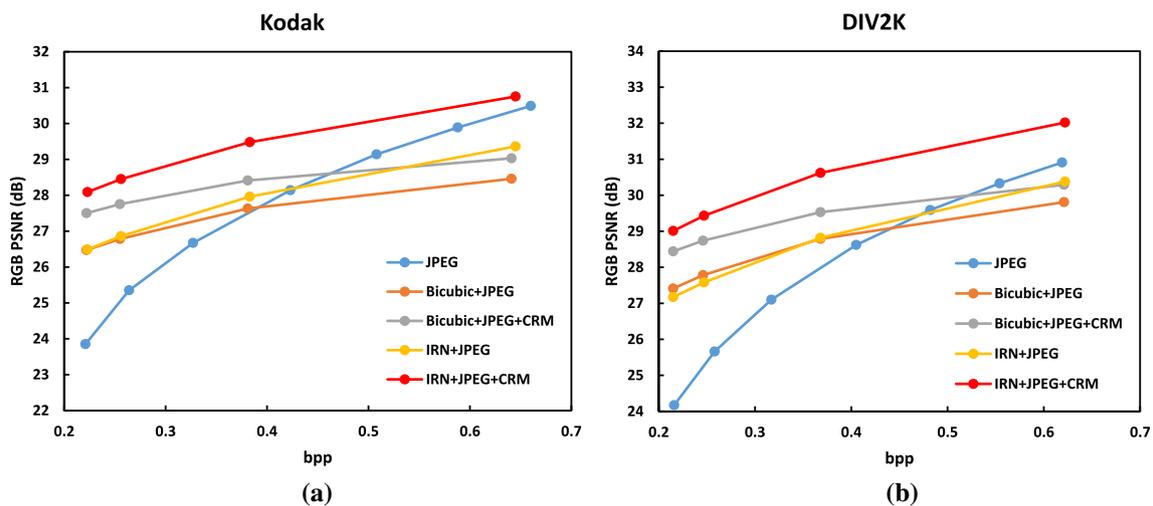
Fig. 9 Results of combination between image rescaling and lossy image compression methods on different datasets. The rescaling scale is 2×. We tune the quality of JPEG algorithm for different compression ratios. RGB PSNR and bit rate (bit per pixel, bpp) are evaluated

Table 13 also demonstrates the significant improvement of IRN_{color} compared with Xia et al. (2018). Note that under this test setting, the distribution of images could be inconsistent with training images for IRN_{color} due to the degradation by rescaling images to the size 256×256. Despite this, IRN_{color} still outperforms Xia et al. (2018) by 5.75 dB with much fewer parameters, further indicating the effectiveness and high efficiency of the proposed model.

Figures 7 and 8 illustrate the visual quality of the grayscale and reconstructed images, as well as the comparison with other methods. It shows that the reconstructed images could have almost the same perception as the original ones. And

compared with Xia et al. (2018), whose reconstructed images may contain some noise or strange variegation, IRN_{color} achieves more fidelity and better visual perception.

4.4 Combination with Image Compression

In this section, we evaluate the combination of image rescaling and image compression methods as described in Sect. 3.5.

For the combination with lossless image compression, we choose two representative methods, i.e. PNG and FLIF (Sneyers and Wuille, 2016), as an example. PNG is a classical lossless image compression algorithm, while FILF is a more

Ground Truth
img_03 from Kodak

JPEG (quality 5)
(RGB) PSNR: 25.16 / 0.7146; (Y) PSNR: 29.27 / 0.7879
bpp: 0.179

Bicubic + JPEG (quality 40)
(RGB) PSNR: 29.19 / 0.8064; (Y) PSNR: 31.65 / 0.8501
bpp: 0.162

IRN + JPEG (quality 40) + CRM
(RGB) PSNR: 31.75 / 0.8536; (Y) PSNR: 33.58 / 0.8817
bpp: 0.163

Ground Truth
img_03 from Kodak

JPEG (quality 35)
(RGB) PSNR: 33.38 / 0.8967
(Y) PSNR: 36.26 / 0.9286
bpp: 0.493

Bicubic + JPEG (quality 90)
(RGB) PSNR: 31.62 / 0.8795
(Y) PSNR: 33.68 / 0.9092
bpp: 0.482

IRN + JPEG (quality 90) + CRM
(RGB) PSNR: 34.62 / 0.9126
(Y) PSNR: 36.44 / 0.9323
bpp: 0.486

Fig. 10 Qualitative results of image compression methods

recent one based on machine learning algorithms. We choose the popular JPEG lossy image compression method as the comparison standard for the trade-off between compression ratio and image quality. Because there is no hyper-parameter for image rescaling and lossless image compression to control the compression ratio, we tune the quality of JPEG to compare the compression performance with different rescaling methods under similar image quality respectively. We evaluate the total storage size for the DIV2K validation set, which contains 100 images, as compression performance, and average RGB PSNR as image quality.

As shown in Table 14, when compared with other image downscaling and upscaling methods, IRN always shows its advantage in the trade-off between compression ratio and image quality. When compared with classical lossy image compression methods, IRN with advanced lossless compression methods can directly outperform JPEG. IRN could get promising results, especially under the condition that high compression performance is required.

For the combination with lossy image compression, we choose the classical JPEG algorithm as an example. As described in Sect. 3.5, we train a Compression Restore Module (CRM) to restore the lost information in compression, which is a neural network consisting of eight residual in residual dense blocks (RRDB) introduced in the ESRGAN model (Wang et al., 2018). We tune the quality of JPEG, and the R-D curves are shown in Fig. 9. As explained in Sect. 3.5, directly combining IRN and JPEG may not perform well because JPEG introduces additional information loss which goes against our invertible framework. This problem is mitigated by CRM. Results demonstrate that IRN combined with JPEG and CRM achieves satisfactory compression performance compared with traditional image rescaling and compression methods. Also, the ablation experiments of Bicubic+JPEG, Bicubic+JPEG+CRM, and IRN+JPEG illustrate that the performance improvement is not majorly owed to CRM, but the effectiveness of our proposed combination between the invertible framework and restoration from existing degradation methods. Additionally, we present qualitative visual results in Fig. 10. It demonstrates the improvement of our proposed model for clearer details under similar compression ratios.

5 Conclusion

In this paper, we propose a novel invertible framework for the bidirectional image degradation–restoration task, which models degradation and restoration from the perspective of invertible transformation to largely mitigate the ill-posed problem. By bijectively transforming the distribution of image-specific lost contents into a pre-specified image-agnostic distribution together with the generation of degraded images, the proposed invertible framework can model lost information and keep the knowledge of distribution transformation in the invertible model. In the inverse restoration, an easily sampled latent variable in company with the generated degraded image is able to reconstruct images through the inverse transformation. Our deliberately designed architecture and effective training objectives enable the proposed IRN model to achieve the goals of the invertible framework in the image rescaling scenario, and it is easily adapted to similar tasks such as image decolorization–colorization. Further, we propose the combination between our invertible framework and restoration from existing degradation methods for wider applications, with an instantiation of the combination of image rescaling and compression. Our extensive experiments demonstrate the significant improvement of our model both quantitatively and qualitatively, as well as the lightweight property and high efficiency of our model. More ablation and extension experiments further provide detailed analysis and illustrate the generalization ability of the proposed method.

Author Contributions MX, SZ, and CL conceptualized the work and designed the methodology. MX and CL formulated the mathematical formulation. MX conducted the experiments. MX, SZ, and CL analyzed the results. ZL and TY. CL supervised the work. All authors wrote and revised the manuscript.

Availability of data and materials All the datasets used in the paper are publicly available.

Code availability Our code is available at https://github.com/pkuxmq/Invertible-Image-Rescaling. We also provide the full code in the supplementary materials.

Declarations

Competing interests The authors have no competing interests to declare that are relevant to the content of this article.

References

Agustsson, E., & Timofte, R. (2017). Ntire 2017 challenge on single image super-resolution: Dataset and study. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*.

Agustsson, E., Tschannen, M., Mentzer, F., Timofte, R., & Gool, L. V. (2019). Generative adversarial networks for extreme learned image

compression. In *Proceedings of the IEEE international conference on computer vision*.

Ardizzone, L., Kruse, J., Wirkert, S., Rahner, D., Pellegrini, E. W., Klessen, R. S., & Köthe, U. (2019). Analyzing inverse problems with invertible neural networks. In *Proceedings of the international conference on learning representations*.

Ardizzone, L., Lüth, C., Kruse, J., Rother, C., & Köthe, U. (2019). Guided image generation with conditional invertible neural networks. Preprint arXiv:1907.02392

Arjovsky, M., & Bottou, L. (2017). Towards principled methods for training generative adversarial networks. In *Proceedings of the international conference on learning representations*.

Asim, M., Daniels, M., Leong, O., Ahmed, A., & Hand, P. (2020). Invertible generative models for inverse problems: Mitigating representation error and dataset bias. In *Proceedings of the international conference on machine learning*.

Bala, R., & Eschbach, R. (2004). Spatial color-to-grayscale transform preserving chrominance edge information. In *Color and imaging conference*.

Ballé, J., Laparra, V., & Simoncelli, E. P. (2017). End-to-end optimized image compression. In *Proceedings of the international conference on learning representations*.

Ballé, J., Minnen, D., Singh, S., Hwang, S. J., & Johnston, N. (2018). Variational image compression with a scale hyperprior. In *Proceedings of the international conference on learning representations*.

Behrmann, J., Grathwohl, W., Chen, R. T., Duvenaud, D., & Jacobsen, J.-H. (2019). Invertible residual networks. In *Proceedings of the international conference on machine learning*.

Bengio, Y., Léonard, N., & Courville, A. (2013). Estimating or propagating gradients through stochastic neurons for conditional computation. Preprint arXiv:1308.3432

Berg, R. V. D., Hasenclever, L., Tomczak, J. M., & Welling, M. (2018). Sylvester normalizing flows for variational inference. In *Proceedings of the conference on uncertainty in artificial intelligence*.

Bevilacqua, M., Roumy, A., Guillemot, C., & Morel, M.-L.A. (2012). Low-complexity single-image superresolution based on nonnegative neighbor embedding. In *British machine vision conference (BMVC)*.

Blau, Y., Mechrez, R., Timofte, R., Michaeli, T., & Zelnik-Manor, L. (2018). The 2018 pirm challenge on perceptual image super-resolution. In *European conference on computer vision workshops (ECCVW)*.

Bruckstein, A. M., Elad, M., & Kimmel, R. (2003). Downscaling for better transform compression. *IEEE Transactions on Image Processing, 12*(9), 1132–1144.

Chen, R. T., Behrmann, J., Duvenaud, D. K., & Jacobsen, J.-H. (2019). Residual flows for invertible generative modeling. *Advances in Neural Information Processing Systems*.

Chen, Y., Xiao, X., Dai, T., & Xia, S.-T. (2020). Hrnet: Hamiltonian rescaling network for image downscaling. In *Proceedings of the IEEE international conference on image processing (ICIP)*.

Cheng, K. L., Xie, Y., & Chen, Q. (2021). IICNet: A generic framework for reversible image conversion. In *Proceedings of the IEEE international conference on computer vision*.

Cheng, Z., Sun, H., Takeuchi, M., & Katto, J. (2020). Learned image compression with discretized gaussian mixture likelihoods and attention modules. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Dai, T., Cai, J., Zhang, Y., Xia, S.-T., & Zhang, L. (2019). Second-order attention network for single image super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Deshpande, A., Lu, J., Yeh, M.-C., Jin Chong, M., & Forsyth, D. (2017). Learning diverse image colorization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Dinh, L., Krueger, D., & Bengio, Y. (2015). NICE: Nonlinear independent components estimation. In *Workshop of the international conference on learning representations*.

Dinh, L., Sohl-Dickstein, J., & Bengio, S. (2017). Density estimation using real NVP. In *Proceedings of the international conference on learning representations*.

Dong, C., Loy, C. C., He, K., & Tang, X. (2015). Image super-resolution using deep convolutional networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 38*(2), 295–307.

Franzen, R. (1999). Kodak lossless true color image suite. source: http://r0k.us/graphics/kodak

Freedman, G., & Fattal, R. (2011). Image and video upscaling from local self-examples. *ACM Transactions on Graphics (TOG), 30*(2), 12.

Giachetti, A., & Asuni, N. (2011). Real-time artifactfree image upscaling. *IEEE Transactions on Image Processing, 20*(10), 2760–2768.

Glasner, D., Bagon, S., & Irani, M. (2009a). Super-resolution from a single image. In *Proceedings of the IEEE international conference on computer vision*.

Glasner, D., Bagon, S., & Irani, M. (2009b). Super-resolution from a single image. In *Proceedings of the IEEE international conference on computer vision*.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., & Bengio, Y. (2014). Generative adversarial nets. *Advances in Neural Information Processing Systems*.

Grathwohl, W., Chen, R. T., Betterncourt, J., Sutskever, I., & Duvenaud, D. (2019). FFJORD: Free-form continuous dynamics for scalable reversible generative models. In *Proceedings of the international conference on learning representations*.

Guo, Y., Chen, J., Wang, J., Chen, Q., Cao, J., Deng, Z., & Tan, M. (2020). Closed-loop matters: Dual regression networks for single image super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., & Hochreiter, S. (2017). Gans trained by a two timescale update rule converge to a local nash equilibrium. *Advances in Neural Information Processing Systems*.

Huang, J.-B., Singh, A., & Ahuja, N. (2015). Single image super-resolution from transformed self-exemplars. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Huang, Y.-C., Chen, Y.-H., Lu, C.-Y., Wang, H.-P., Peng, W.-H., & Huang, C.-C. (2021). Video rescaling networks with joint optimization strategies for downscaling and upscaling. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Hyvärinen, A., & Pajunen, P. (1999). Nonlinear independent component analysis: Existence and uniqueness results. *Neural Networks, 12*(3), 429–439.

Jacobsen, J.-H., Smeulders, A. W., & Oyallon, E. (2018). irevnet: Deep invertible networks. In *Proceedings of the international conference on learning representations*.

Jing, J., Deng, X., Xu, M., Wang, J., Guan, Z. (2021). Hinet: Deep image hiding by invertible network. Proceedings of the IEEE International Conference on Computer Vision.

Johnson, J., Alahi, A., & Fei-Fei, L. (2016). Perceptual losses for real-time style transfer and super-resolution. In *Proceedings of the European conference on computer vision (ECCV)*.

Kim, H., Choi, M., Lim, B., & Mu Lee, K. (2018). Task-aware image downscaling. In *Proceedings of the European conference on computer vision (ECCV)*.

Kim, K. I., & Kwon, Y. (2010). Single-image superresolution using sparse regression and natural image prior. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 32*(6), 1127–1133.

Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization. In *Proceedings of the international conference on learning representations*.

Kingma, D. P., & Dhariwal, P. (2018). Glow: Generative flow with invertible 1x1 convolutions. *Advances in Neural Information Processing Systems*.

Kingma, D. P., Salimans, T., Jozefowicz, R., Chen, X., Sutskever, I., & Welling, M. (2016). Improved variational inference with inverse autoregressive flow. *Advances in Neural Information Processing Systems*.

Kobyzev, I., Prince, S., Brubaker, M. (2020). Normalizing flows: An introduction and review of current methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Kopf, J., Shamir, A., & Peers, P. (2013). Content-adaptive image downscaling. *ACM Transactions on Graphics (TOG), 32*(6), 173.

Kumar, M., Babaeizadeh, M., Erhan, D., Finn, C., Levine, S., Dinh, L., & Kingma, D. (2020). Videoflow: A conditional flow-based model for stochastic video generation. In *Proceedings of the international conference on learning representations*.

Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., Acosta, A., et al. (2017). Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Levin, A., Lischinski, D., & Weiss, Y. (2004). Colorization using optimization. ACM SIGGRAPH.

Li, M., Zuo, W., Gu, S., You, J., & Zhang, D. (2020). Learning content-weighted deep image compression. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Li, Y., Liu, D., Li, H., Li, L., Li, Z., & Wu, F. (2018). Learning a convolutional neural network for image compactresolution. *IEEE Transactions on Image Processing, 28*(3), 1092–1107.

Li, Z., Li, S., Zhang, N., Wang, L., & Xue, Z. (2019). Multiscale invertible network for image super-resolution. *Proceedings of the ACM Multimedia Asia*.

Lim, B., Son, S., Kim, H., Nah, S., & Mu Lee, K. (2017). Enhanced deep residual networks for single image super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*.

Lin, W., & Dong, L. (2006). Adaptive downsampling to improve image compression at low bit rates. *IEEE Transactions on Image Processing, 15*(9), 2513–2521.

Liu, C., Tang, H., Qin, T., Wang, J., & Liu, T.-Y. (2021). On the generative utility of cyclic conditionals. *Advances in Neural Information Processing Systems*.

Liu, J., He, S., & Lau, R. W. (2017). $l_{\{0\}}$-regularized image downscaling. *IEEE Transactions on Image Processing, 27*(3), 1076–1085.

Liu, Q., Liu, P. X., Xie, W., Wang, Y., & Liang, D. (2015). Gcsdecolor: Gradient correlation similarity for efficient contrast preserving decolorization. *IEEE Transactions on Image Processing, 24*(9), 2889–2904.

Liu, Y., Qin, Z., Anwar, S., Ji, P., Kim, D., Caldwell, S., & Gedeon, T. (2021). Invertible denoising network: A light solution for real noise removal. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Lu, C., Chen, J., Li, C., Wang, Q., & Zhu, J. (2021). Implicit normalizing flows. In *International conference on learning representations*.

Lu, S.-P., Wang, R., Zhong, T., & Rosin, P. L. (2021). Largecapacity image steganography based on invertible neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Lugmayr, A., Danelljan, M., Van Gool, L., & Timofte, R. (2020). Srflow: Learning the super-resolution space with normalizing flow. In *Proceedings of the European conference on computer vision (ECCV)*.

Martin, D., Fowlkes, C., Tal, D., Malik, J., et al. (2001). A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proceedings of the IEEE international conference on computer vision*.

Minnen, D., Ballé, J., & Toderici, G. D. (2018). Joint autoregressive and hierarchical priors for learned image compression. *Advances in Neural Information Processing Systems*.

Mitchell, D. P., & Netravali, A. N. (1988). Reconstruction filters in computer-graphics. ACM Siggraph Computer Graphics (Vol. 22-4, pp. 221–228).

Oeztireli, A. C., & Gross, M. (2015). Perceptually based downscaling of images. *ACM Transactions on Graphics (TOG), 34*(4), 77.

Ren, S., Padilla, W., & Malof, J. (2020). Benchmarking deep inverse models over time, and the neural-adjoint method. *Advances in Neural Information Processing Systems*.

Rezende, D., & Mohamed, S. (2015). Variational inference with normalizing flows. In *Proceedings of the international conference on machine learning*.

Rippel, O., & Bourdev, L. (2017). Real-time adaptive image compression. In *Proceedings of the international conference on machine learning*.

Schulter, S., Leistner, C., & Bischof, H. (2015). Fast and accurate image upscaling with super-resolution forests. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Shannon, C. E. (1949). Communication in the presence of noise. *Proceedings of the IRE, 37*(1), 10–21.

Shen, M., Xue, P., & Wang, C. (2011). Down-sampling based video coding using super-resolution technique. *IEEE Transactions on Circuits and Systems for Video Technology, 21*(6), 755–765.

Sneyers, J., & Wuille, P. (2016). Flif: Free lossless image format based on maniac compression. In *Proceedings of the IEEE international conference on image processing (ICIP)*.

Sullivan, G. J., Ohm, J. R., Han, W. J., & Wiegand, T. (2013). Overview of the high efficiency video coding (hevc) standard. *IEEE Transactions on Circuits and Systems for Video Technology, 22*(12), 1649–1668.

Sun, W., & Chen, Z. (2020). Learned image downscaling for upscaling using content adaptive resampler. *IEEE Transactions on Image Processing, 29,* 4027–4040.

Teshima, T., Ishikawa, I., Tojo, K., Oono, K., Ikeda, M., & Sugiyama, M. (2020). Coupling-based invertible neural networks are universal diffeomorphism approximators. *Advances in Neural Information Processing Systems*.

Tian, Y., Lu, G., Min, X., Che, Z., Zhai, G., Guo, G., & Gao, Z. (2021). Self-conditioned probabilistic learning of video rescaling. In *Proceedings of the IEEE international conference on computer vision*.

van der Ouderaa, T. F., & Worrall, D. E. (2019). Reversible gans for memory-efficient image-to-image translation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Wang, X., Yu, K., Wu, S., Gu, J., Liu, Y., Dong, C., & Loy, C. C. (2018). Esrgan: Enhanced super-resolution generative adversarial networks. In *European conference on computer vision workshops (ECCVW)*.

Wang, Y., Xiao, M., Liu, C., Zheng, S., & Liu, T.-Y. (2020). Modeling lost information in lossy image compression. Preprint arXiv:2006.11999

Wang, Z., Bovik, A. C., Sheikh, H. R., Simoncelli, E. P., et al. (2004). Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing, 13*(4), 600–612.

Weber, N., Waechter, M., Amend, S. C., Guthe, S., & Goesele, M. (2016). Rapid, detail-preserving image downscaling. *ACM Transactions on Graphics (TOG), 35*(6), 205.

Wu, X., Zhang, X., & Wang, X. (2009). Low bit-rate image compression via adaptive down-sampling and constrained least squares upconversion. *IEEE Transactions on Image Processing, 18*(3), 552–561.

Xia, M., Liu, X., & Wong, T.-T. (2018). Invertible grayscale. *ACM Transactions on Graphics (TOG), 37*(6), 1–10.

Xiao, M., Zheng, S., Liu, C., Wang, Y., He, D., Ke, G., & Liu, T.-Y. (2020). Invertible image rescaling. In *Proceedings of the European conference on computer vision (ECCV)*.

Xie, Y., Cheng, K. L., & Chen, Q. (2021). Enhanced invertible encoding for learned image compression. In *Proceedings of the 29th ACM international conference on multimedia*.

Xing, J., Hu, W., & Wong, T.-T. (2022). Scale-arbitrary invertible image downscaling. Preprint arXiv:2201.12576.

Xing, Y., Qian, Z., & Chen, Q. (2021). Invertible image signal processing. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Yang, J., Wright, J., Huang, T. S., & Ma, Y. (2010). Image super-resolution via sparse representation. *IEEE Transactions on Image Processing, 19*(11), 2861–2873.

Ye, T., Du, Y., Deng, J., & He, S. (2020). Invertible grayscale via dual features ensemble. *IEEE Access, 8,* 89670–89679.

Yeo, H., Do, S., & Han, D. (2017). How will deep learning change internet video delivery? Proceedings of the 16th ACM Workshop on Hot Topics in Networks.

Yeo, H., Jung, Y., Kim, J., Shin, J., & Han, D. (2018). Neural adaptive content-aware internet video delivery. In *13th USENIX Symposium on Operating Systems Design and Implementation (OSDI 18)*.

Zeyde, R., Elad, M., & Protter, M. (2010). On single image scale-up using sparse-representations. In *International conference on curves and surfaces*.

Zhang, R., Isola, P., & Efros, A. A. (2016). Colorful image colorization. In *Proceedings of the European conference on computer vision (ECCV)*.

Zhang, R., Isola, P., Efros, A. A., Shechtman, E., & Wang, O. (2018). The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Zhang, R., Zhu, J.-Y., Isola, P., Geng, X., Lin, A. S., Yu, T., & Efros, A. A. (2017). Real-time user-guided image colorization with learned deep priors. *ACM Transactions on Graphics (TOG), 36*(4), 1–11.

Zhang, Y., Li, K., Li, K., Wang, L., Zhong, B., & Fu, Y. (2018). Image super-resolution using very deep residual channel attention networks. *Proceedings of the European conference on computer vision (ECCV)*.

Zhang, Y., Tian, Y., Kong, Y., Zhong, B., & Fu, Y. (2018). Residual dense network for image super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Zhao, R., Liu, T., Xiao, J., Lun, D. P., & Lam, K.-M. (2021). Invertible image decolorization. *IEEE Transactions on Image Processing, 30,* 6081–6095.

Zhong, Z., Shen, T., Yang, Y., Lin, Z., & Zhang, C. (2018). Joint sub-bands learning with clique structures for wavelet domain super-resolution. *Advances in Neural Information Processing Systems*.

Zhu, X., Li, Z., Zhang, X.-Y., Li, C., Liu, Y., & Xue, Z. (2019). Residual invertible spatio-temporal network for video super-resolution. In *Proceedings of the AAAI conference on artificial intelligence*.