Enhancing Image Rescaling using Dual Latent Variables in Invertible Neural Network

Min Zhang
University of Southern California
Los Angeles, California, USA
zhan980@usc.edu

Zhihong Pan
C.-C. Jay Kuo
University of Southern California
Los Angeles, California, USA
zhihongpan,zhouxin16@baidu.com
cckuo@sipi.usc.edu

ABSTRACT
Normalizing flow models have been used successfully for generative image super-resolution (SR) by approximating complex distribution of natural images to simple tractable distribution in latent space through Invertible Neural Networks (INN). These models can generate multiple realistic SR images from one low-resolution (LR) input using randomly sampled points in the latent space, simulating the ill-posed nature of image upsampling where multiple high-resolution (HR) images correspond to the same LR. Lately, the invertible process in INN has also been used successfully by bidirectional image rescaling models like IRN and HCFlow for joint optimization of downsampling and inverse upsampling, resulting in significant improvements in upscaled image quality. While they are optimized for image downscaling too, the ill-posed nature of image downsampling, where one HR image could be downsized to multiple LR images depending on different interpolation kernels and resampling methods, is not considered. A new downscaling latent variable, in addition to the original one representing uncertainties in image upsampling, is introduced to model variations in the image downsampling process. This dual latent variable enhancement is applicable to different image rescaling models and it is shown in extensive experiments that it can improve image upsampling accuracy consistently without sacrificing image quality in downscaled LR images. It is also shown to be effective in enhancing other INN-based models for image restoration applications like image hiding.

CCS CONCEPTS
• Computing methodologies → Image processing: Reconstruction.

KEYWORDS
image rescaling, latent variable, invertible neural network

ACM Reference Format:
Min Zhang, Zhihong Pan, Xin Zhou, and C.-C. Jay Kuo. 2022. Enhancing Image Rescaling using Dual Latent Variables in Invertible Neural Network. In Proceedings of the 30th ACM International Conference on Multimedia (MM ’22), Oct. 10–14, 2022, Lisbon, Portugal. ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3503161.3548376

1 INTRODUCTION
Recent deep learning based image super-resolution (SR) methods have advanced the performance of image upscaling significantly. These methods are only optimized for the unidirectional upscaling process where the LR inputs are synthesized from a predefined downsampling kernel. To take advantage of the potential mutual beneficiary reinforcement between downsampling and the inverse upscaling, some image rescaling models [14, 26, 32] are developed to optimize the downscaling and the inverse upscaling processes jointly and they yield significant improvements in accuracy of the upscaling task comparing to unidirectional SR models of the same scale factors. The state-of-the-art (SOTA) for learning based bidirectional image rescaling is set by the invertible rescaling net (IRN) as proposed by Xiao et al. [32]. As shown in Fig. 1, it is able to achieve the best performance so far as both the Haar transformation and the invertible neural network (INN) [2] backbone are invertible processes which fit naturally with the downsampling and inverse upscaling process.

Here the downscaling and inverse upscaling process of IRN can be described as \( y, z = f(x) \) and \( x = f^{-1}(y, z) \) respectively. When latent variable \( z \) is preserved, \( x \) can be perfectly restored as it is an invertible process. This ideal inverse process does not exist in real applications when only LR output \( y \) is saved. When the network is optimized to store as much information as allowed in \( y \) and transform \( z \) as a random variable independent of \( y \), the restoration of \( x \) can be calculated as \( f^{-1}(y, z) \) where \( z \) is randomly sampled from a distribution like multivariate Gaussian, illustrated as a colored area surrounding \( z \) in Fig. 1. After training, IRN is optimized to estimate the same \( x \) from multiple \( z \) samples. For a given pair of \( x \) and \( y \), it is ideal to have the perfect restoration from multiple samples like

\[ x = f^{-1}(y, \tilde{z}), i \in \mathbb{I}, \]  

where \( \mathbb{I} \) is a set with more than one element. However, this is not feasible with IRN as it is self-conflicting with the invertible process. Assume \( |\mathbb{I}| > 1 \), there must exist two different \( \tilde{z} \) and \( \tilde{z} \) that satisfy \( f^{-1}(y, \tilde{z}) = f^{-1}(y, \tilde{z}) = x \). We can then see that \( y, \tilde{z} = y, \tilde{z} \) after applying \( f(\cdot) \) to both outputs. Consequently, \( \tilde{z} \) and \( \tilde{z} \) are identical and the size of set \( \mathbb{I} \) must be 1. In other words, for given \( x \) and \( y \), the average restoration error of \( x \) from randomly sampled \( z \) must be larger than zero as the zero-error prediction is only valid for one sample. For reference, the distribution of \( \hat{x} \) is illustrated as an area surrounding \( x \) in Fig. 1.

Aiming to resolve this self-conflicting problem, we introduce a new latent variable \( \tilde{w} \) during the forward downsampling process and the enhanced invertible rescaling network with dual latent variables,
Invertible Neural Network (IRN) is used for feature mixing from where...

2.1 Invertible Neural Network

The INN [2, 6, 9, 10, 16] has an architecture of $f_0$ whose inverse function $f_0^{-1}$ share the same parameters, leading to a cheaper inference. Specifically, given an input $x$, INN generates $z = f_0(x)$ in the forward pass, and $x$ can be recovered by $x = f_0^{-1}(z)$. In practice, for a complex distribution $p(x)$, $z$ is commonly designed as an unobserved latent variable with a predefined tractable distribution. As a result, the generative or reconstruction process in $x$ can be modeled as random sampling in $z$ and this model can be optimized using standard SGD-based techniques as the negative log-likelihood (NLL) can be computed exactly.

INN was first proposed in NICE [9], a flow model by stacking non-linear additive coupling and other transformation layers. RealNVP [16] introduced multi-scale layers and substitute the non-linear additive coupling with affine coupling for lower computational cost and better regularization ability. Furthermore, the fixed permutation layer is replaced with $1 \times 1$ convolutional layers in Glow [16]. In contrast to previous unconditional generative models, lately INN have been applied to various conditional generative models like image SR [21] and image colorization [3].

2.2 Image Rescaling

Image rescaling includes image downscaling and upsampling. Image downscaling resizes the HR image to a lower resolution which is visually pleasing. Frequency-based kernels [23], such as Bilinear and Bicubic, are commonly used for image downscaling. Image upsampling reconstructs promising HR image from the downscaled LR image, which is also known as image SR. While powerful deep learning techniques have led to developments of many image SR...
models [11, 15, 19, 35, 36] with impressive results, they rely on LR inputs generated from predefined downsampling settings.

Recently, an encoder-decoder architecture [14] is utilized as the first to jointly optimize the downsampling and upsampling process in bidirectional image rescaling. Later a new content adaptive-resampler based image downsampling module [26] was proposed to train with existing differentiable SR models jointly. However, these methods still suffer from the ill-posed problem, a visually plausible downsampled image may not be optimal for inverse upsampling. More recently, IRN [32] was proposed to use the bijective process in INN to model downsampling and upsampling according to their reciprocal nature. The proposed IRN is capable of generating a visually pleasing LR image and a latent variable z as well as restoring HR accurately from the saved LR and randomly sampled z. While latent variable in IRN is regulated to be independent of the LR image, HCFlow [18] proposed a hierarchical conditional framework so that the high frequency components are conditional on the LR image hierarchically. Later FGRN [17] proposed an encoder-decoder architecture to model downsampling and upsampling while using a separate invertible flow module, without using latent variables, as a guidance to learn the optimal image downsampling in conjunction with image upsampling.

### 2.3 Steganography and Image Hiding

Steganography is the practice of hiding one message into a carrier, such as audio, image or video. Image hiding, specifically, is to unobtrusively place a whole image, i.e., secret image, within another image of the same size, i.e., cover image, and the secret image should be recovered from the stego image at the receiver end. Least Significant Bit (LSB) [27] is a classic spatial domain method which uses the n most significant bits of the secret image to replace the n least significant bits of the cover image. Newer methods use a similar but describe information in frequency domains such as discrete wavelet transform (DWT) domain [5] so it is more undetectable.

Baluja [4] proposed the first deep learning solution for image hiding, which contains two sub-networks. The concealing network hides the secret image into the cover image, outputting a stego image. Then the revealing network reconstructs the secret image from the stego image. The two sub-networks work as a pair and are trained simultaneously, but they do not share parameters so that the connection is loose, causing texture-copying artifacts and color distortion. Recently, HiNet [13] utilized INN to solve the image hiding problem, sharing the same set of network parameters for both image concealing and revealing, which improved the reconstruction accuracy significantly. Besides, the secret information is hidden in the wavelet domain rather than pixel domain for high invisibility in the stego image.

### 3 PROPOSED METHOD

#### 3.1 Preliminaries

Flow-based models, which aim to learn a bijective mapping between the target space and the latent space, have been investigated for various applications. For image generation models, the target space of HR images is modeled as a high-dimensional random variable x with a distribution of x ~ p(x). The key aspect of flow models is the employment of an invertible neural network (INN) \( f_\theta \) that transforms x to a latent variable z with simple tractable distribution z ~ p(z) (e.g. Gaussian distribution). Here \( \theta \) represents the parameters of the invertible network which could be learned from training samples. With the invertible network \( f_\theta \), an HR image can be transformed to a latent variable as z = \( f_\theta(x) \), and it can also be restored from a latent variable as x = \( f_\theta^{-1}(z) \). Another key aspect of flow models is that, according to the change of variable formula, the probability density \( p(x|\theta) \) can be calculated as

\[
p(x|\theta) = p(f_\theta(x)) \left| \frac{\partial f_\theta}{\partial x}(x) \right|.
\]

This allows the exact negative log-likelihood (NLL) \( -\log p(x|\theta) \) to be computed and the network can then be trained by directly minimizing it using standard SGD-based techniques.

Lately, there have been investigations of conditional flow models, like using class [25] or image [3,49] as conditions for image generation. More recently, SRFlow is proposed to generate realistic images from the condition of an LR image y. Similar to Equation 5, the conditional probability density of x is calculated as

\[
p(x|y, \theta) = p(f_\theta(x|y)) \left| \frac{\partial f_\theta}{\partial x}(x|y) \right|.
\]

Using a large set of HR-LR training pairs \( \{(x_i, y_i)\}_{i=1}^M \), the invertible network can be trained by minimizing the NLL loss using a data-driven process. After training, the conditional distribution \( p(x|y, \theta) \) captures the nature of all possible realistic HR images corresponding to a known y and multiple SR images can be generated from one LR reference y and randomly sampled latent variable z.

Most recently, HCFlow aims to use the bijective transformation of flow models to model two different modes: SR generation from LR and bidirectional image rescaling. In the latter case, y is not the known LR ground-truth (GT) used as a condition, but part of outputs in the latent space as x ↔ (y, a) = \( f_\theta(x) \) where a is an intermediate variable representing decomposed high-frequency components from x. To explain the role of the flow model in transforming the complex distribution of natural images x to tractable distribution in latent space, a GT LR y is introduced as a condition as below

\[
p(x|y) \leftrightarrow p(y, a|y) = p(a|y) p(y|y).
\]

Ideally, the model is trained to generate the LR image y exactly as the GT LR y, so the first factor of the above calculation is simplified as \( p(a|y) \), which is further mapped to a standard multivariate Gaussian distribution in latent space: \( p(z) = N(z|0, I) \). And the second factor could also be formulated as \( \delta(y - \tilde{y}) \), which can be further approximated by a multivariate Gaussian distribution as \( \lim_{\Sigma \to 0} N(x|y, \Sigma) \). As a result, the complex distribution \( p(x|y) \) becomes the product of two tractable Gaussian distributions. Note that a is introduced to make the transformation from and to latent variable x conditional to y for generative SR models as they needed to be trained by maximum likelihood estimation (MLE).

#### 3.2 Dual Latent Variables

While the newly introduced intermediate variable a in HCFlow makes it possible to share the same invertible network architecture for both generative image SR and bidirectional image rescaling, it is not really needed for image rescaling as the NLL loss is not used in actual training. Alternatively, IRN works on the bidirectional...
image rescaling problem exclusively. It uses the INN backbone to transform the input image \( x \) to an LR output \( y \) and latent variable \( z \) without intermediate \( a \). By training the network to make \( z \) a Gaussian distribution independent of \( y \), an accurate restoration of \( x \) is made possible using randomly sampled \( \hat{z} \) during the inverse upsampling process. Similar to Equation 7, for IRN, the distribution transformation process becomes \( p(x) \leftrightarrow p(y)p(z) \).

While both SRFlow and IRN use similar INN backbones to model bijective transformation between the image space and the latent space and achieve great performance in generative image SR and image rescaling respectively, there is a distinctive difference in the purpose of random sampling in the latent space. For SRFlow, the random sampling of \( z \) is beneficial to generate diverse versions of SR output from one LR image. This is also aligned with one aspect of the ill-posed nature in image SR application, as there are multiple HR images corresponding to one LR image. In other words, \( z \) is a latent variable for image upsampling, which represents high-frequency components of HR images that can only be observed after the image upsampling process of \( f_\theta^{-1}(\cdot) \) is applied. While for IRN in ideal situations, as its goal is to restore HR image exactly, the random sampling of \( z \) works against the goal of restoring the same HR result \( x \) when \( y \) is known. When \( x \) is given, \( y \) and \( z \) are also determined as \( (y, z) = f_\theta(x) \). However, since \( z \) is invisible for the inverse upsampling process, a randomly sampled \( \hat{z} \) is used instead. As explained in the introduction, different sampling of \( \hat{z} \) must map with different \( \hat{x} \) due to the invertible nature of the network. That is, if \( \hat{z}_i = \hat{z} \), then for any \( j \neq i \), we can see that \( \hat{x}_j \neq \hat{x}_i = \hat{x} \) so \( \|x - \hat{x}_j\|_2 > 0 \). For a given \( x \) and any \( N > 1 \), the expected restoration error of \( N \) randomly sampled \( \hat{x} \) has a positive lower boundary as

\[
E[\|x - \hat{x}\|_2] > 0.
\]

To alleviate this self-conflicting nature between random sampling of \( \hat{z} \) and exact restoration of \( x \), we propose a second latent variable \( w \) as additional input besides \( x \). The forward downsampling and backward upsampling process become

\[
y, z = f_\theta(x, w),
\]

\[
x, \hat{w} = f_\theta^{-1}(y, \hat{z}),
\]

where \( w \) is randomly sampled from an independent normal distribution for downsampling while \( \hat{z} \) is randomly sampled for subsequent upsampling. Using the same optimization objective of minimizing \( E[\|x - \hat{x}\|_2] \) for this random downsampling and upsampling process, we can show that, assuming the learning capability of \( f_\theta(\cdot) \) is not limited, its lower boundary is zero without conflicting with the invertible network characteristics. For any given \( x \), we can generate \( y \) as its corresponding LR output. For any set of \( N \) unique values of \( z \): \( Z = \{z_i|i = 1, 2, \ldots, N\} \) and another set of \( N \) unique elements \( W = \{w_i|i = 1, 2, \ldots, N\} \), there exists a invertible network \( f_{\theta_N}(\cdot) \) that satisfies

\[
y, z_i = f_{\theta_N}(x, w_i),
\]

\[
x, w_i = f_{\theta_N}^{-1}(y, z_i)
\]

for any \( i \in [1..N] \). Due to the unlimited learning ability of \( f_\theta(\cdot) \), we can see that

\[
\lim_{N \to \infty} E[\|x - \hat{x}\|_2] = 0.
\]

In summary, with the introduction of \( w \), the lower boundary of the restoration error is reduced from a positive value to zero theoretically for any big \( N \). While the learning ability of the invertible network is limited in practice, it is shown in extensive experiments that it helps reducing the restoration error for different applications. Alternatively, the introduction of \( w \) allows the modeling of the downsampling aspect of the ill-posed nature of image rescaling. That is, due to variations in blur kernels and resampling methods, there exist various LR outputs from a single HR input. With two latent variables, \( w \) for image downsampling and \( z \) for upsampling, our proposed method can be applied to different INN based image rescaling models to become an enhanced dual latent variable (DLV) version for improved performance.

### 3.3 Model Architecture

The introduction of the downsampling latent variable \( w \) only causes minor changes in model architecture of the baseline model. Using IRN as the primary baseline model here the model architecture of the enhanced DLV-IRN is illustrated in Fig. 2. In addition to
the original HR input \( \mathbf{x} \), the new latent variable \( \mathbf{w} \) is introduced for each pixel. After both are applied with Haar transformation, the low-frequency components of \( \mathbf{x} \) are split and preserved as the low-frequency branch while the remaining high-frequency components are concatenated with all channels from \( \mathbf{w} \) as the mixture branch. Other than increased channel numbers, the transformation networks between the low-frequency and the mixture branches are kept the same as the baseline model. Depending on the scaling factor, there could be more than one InvBlock cascaded to build the full pipeline, with the split and concatenating step applied once at the beginning of each InvBlock. At the end, the output of the low-frequency branch is \( \hat{y} \) and the one from the mixture branch is \( \mathbf{z} \). The image and feature dimensionality at different stages are included in Fig. 2 for reference.

### 3.4 Training Objectives

To better demonstrate the effectiveness of the innovative dual latent variable module itself, it is desired to keep the training objective of original baseline model unchanged for fair comparison. In the case of newly introduced latent variable \( \mathbf{w} \), as it is randomly sampled during the forward downscaling process and could be safely discarded after upscaling, it is possible to use the same overall loss as IRN for training DLV-IRN, denoted as

\[
L = \lambda_1 L_r + \lambda_2 L_g + \lambda_3 L_d + \lambda_4 L_i. \tag{12}
\]

Here \( L_r \) is the L1 reconstruction loss for upscaled HR output \( \hat{\mathbf{x}} \) and \( L_g \) is the L2 guidance loss for downscaled LR output \( y \) in reference to a downsampled LR reference \( \hat{y} \) using bicubic interpolation. For \( L_d \), same as in IRN [33], the partial distribution matching loss \(-\log p(z)\) is used for stable training. These three are similar to the baseline IRN and their weights are also kept unchanged for consistency. The last term \( L_i \) is an LR invariance loss introduced for the newly proposed downsampling later variable \( \mathbf{W} \). As the reverse upsampling output \( \hat{\mathbf{w}} \) has no impact on model performance, the LR invariance loss is introduced to the forward downsampling process only. When \( \mathbf{x} \) is given and \( \mathbf{w}, j \in [1, m] \) is randomly sampled, \( L_i \) is designed to make the output LR \( y_j \) invariant to \( \mathbf{w} \) and it is calculated as

\[
L_i = \frac{1}{m-1} \sum_{j=1}^{m} (y_j - \hat{y})^2 \tag{13}
\]

where \( \hat{y} \) is the average LR output. This loss does not rely on any supervision from LR references and works together with \( L_g \) as an enhanced semi-supervised learning in LR output. In our experiments, \( m \) is set as 3 and \( \lambda_4 = s^2/4 \) where \( s \) is the scaling factor.

## 4 EXPERIMENTS

### 4.1 Experimental Setup

**Dataset and metrics.** Following the IRN baseline, we train our DLV-IRN model on the DIV2K [1] dataset, which contains 800 2K resolution training images. For the DLV-HCFlow, a combined DF2K dataset including both DIV2K and Flickr2K [28] is also used to compare with the HCFlow baseline. For image rescaling, we evaluate our method on five standard benchmark datasets, i.e., the Set5 [7], Set14 [34], BSD100 [22], Urban100 [12] and the validation set of DIV2K. For image hiding, the testing datasets include the validation set of DIV2K with 100,1024 × 1024 images, ImageNet [25] with 50,000 256 × 256 images, and the COCO [20] dataset with 5,000 256 × 256 images. Following IRN [32], PSNR and SSIM [31] on the Y channel of the YCbCr color space are used for assessing upsampled image quality. Since the downscaled LR images do not have ground-truth, we employ NIQE [24] and PIQE [29] which are non-reference metrics in addition to SSIM. For image hiding, we also compare MAE and RMSE by following HiNet [13]. The larger value of PSNR, SSIM and the smaller value of NIQE, PIQE, MAE and RMSE represent better image quality.

**Image rescaling.** Two baseline models, IRN and HCFlow, are used to assess the effectiveness of dual latent variable enhancement. For both DLV-IRN and DLV-HCFlow, the downsampling latent variable \( \mathbf{w} \) is added as a 2-channel pixel-wise variable. The model size change introduced by this channel addition is insignificant. To verify that the performance improvements brought by DLV enhancement is not caused by simple model size change, IRN and HCFlow are also augmented in model depth, denoted as IRN\(^2\) and HCFlow\(^2\), to match their DLV counterparts in parameter numbers for fair comparison. For both the augmented-depth and DLV variants, they are trained using the same settings as their corresponding baselines, including loss weights, learning rate and number of iterations.

**Image hiding.** We follow most of the experimental settings in HiNet [13] except for the following aspects. First, observing that the original HiNet has gradient explosion during training, we change the learning rate to be \( 2 \times 10^{-4} \), halved every 1,000 epochs, and add gradient clipping. Second, the lack of quantization in HiNet causes unreliable results so we add a quantization step before the revealing process. Third, with the above changes, we find that the model can learn more after 80K iterations so the total number of epochs is set to be 5,000. For DLV-HiNet, the 2-channel pixel-wise downsampling latent variable \( \mathbf{w} \) is split to the cover and secret branch separately, each with 1-channel. For fair comparison, we retrained the model of Baluja [4] based on a third party implementation \(^1\) using the same settings mentioned above for consistency.

### 4.2 Ablation Study

For the image rescaling models studied here, while the HR image can consistently restored from random samples of \( \mathbf{z} \), we only need one specific \( \mathbf{z} \) to restoration during testing. In fact, while the random sampling of \( \mathbf{z} \) is critical for diversity in generative models, it is not beneficial to have uncertainty in restored HR image. As the distribution matching loss \( L_d \) is applied to \( \mathbf{z} \) and there is no direct constraint on selection of \( \mathbf{z} \), one practical choice is to keep it consistent across training and testing. As validated in FGRN [17],

---

1https://github.com/arnoweng/PyTorch-Deep-Image-Steganography
Table 2: Quantitative results of upscaled HR image quality from different rescaling methods on benchmark datasets.

| Type | Downscaling & Upscaling | Scale | Param | PSNR | SSIM↑ | PSNR | SSIM↑ | PSNR | SSIM↑ | PSNR | SSIM↑ | PSNR | SSIM↑ | PSNR | SSIM↑ |
|------|-------------------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| I    | Bicubic & Bicubic       | -     |       | 33.66 | 0.9299 | 30.24 | 0.8688 | 29.56 | 0.8431 | 26.88 | 0.8403 | 31.01 | 0.9393 |       |       |
|      | Bicubic & EDSR [19]     | × 2   |       | 40.7M | 38.20 | 0.9606 | 34.02 | 0.9204 | 32.37 | 0.9018 | 33.10 | 0.9363 | 35.12 | 0.9699 |       |       |
|      | Bicubic & RCAN [35]     |       |       | 15.4M | 38.27 | 0.9614 | 34.12 | 0.9216 | 32.41 | 0.9027 | 33.34 | 0.9384 | 35.04 | 0.9405 |       |       |
|      | Bicubic & SAN [8]       |       |       | 15.7M | 38.31 | 0.9620 | 34.07 | 0.9213 | 32.42 | 0.9028 | 33.10 | 0.9370 | 36.73 | 0.9497 |       |       |
|      | CAR & EDSR [26]         |       |       | 51.1M | 38.94 | 0.9658 | 33.61 | 0.9404 | 33.83 | 0.9262 | 35.24 | 0.9572 | 38.26 | 0.9599 |       |       |
|      | IRN†                    |       |       | 1.66M | 43.99 | 0.9871 | 40.79 | 0.9778 | 41.32 | 0.9876 | 39.92 | 0.9865 | 44.32 | 0.9908 |       |       |
|      | FGRN [17]               |       |       | 1.33M | 44.15 | 0.9902 | 42.28 | 0.9840 | 41.87 | 0.9887 | 41.71 | 0.9904 | 45.08 | 0.9917 |       |       |
|      | DLV-IRN (ours)          |       |       | 2.08M | 43.91 | 0.9871 | 40.34 | 0.9777 | 41.25 | 0.9875 | 39.80 | 0.9867 | 44.23 | 0.9908 |       |       |
|      |                         |       |       | 1.89M | 45.42 | 0.9910 | 42.16 | 0.9839 | 42.91 | 0.9916 | 41.29 | 0.9904 | 45.58 | 0.9934 |       |       |
| II   | Bicubic & Bicubic       | -     |       | 28.42 | 0.8104 | 26.00 | 0.7027 | 25.96 | 0.6675 | 23.14 | 0.6577 | 26.66 | 0.8521 |       |       |
|      | Bicubic & EDSR [19]     | × 4   |       | 43.1M | 32.62 | 0.8984 | 28.94 | 0.7901 | 27.79 | 0.7437 | 26.86 | 0.8080 | 29.38 | 0.9032 |       |       |
|      | Bicubic & RCAN [35]     |       |       | 15.6M | 32.63 | 0.9002 | 28.87 | 0.7889 | 27.77 | 0.7436 | 26.82 | 0.8087 | 30.77 | 0.8460 |       |       |
|      | Bicubic & SAN [8]       |       |       | 15.7M | 32.64 | 0.9003 | 28.92 | 0.7888 | 27.78 | 0.7436 | 26.79 | 0.8068 | 31.14 | 0.8510 |       |       |
|      | CAR & EDSR [26]         |       |       | 52.8M | 33.88 | 0.9174 | 30.31 | 0.8382 | 29.15 | 0.8004 | 29.28 | 0.8711 | 32.82 | 0.8837 |       |       |
|      | IRN†                    |       |       | 4.33M | 36.19 | 0.9451 | 32.67 | 0.9015 | 31.64 | 0.8826 | 31.41 | 0.9157 | 35.07 | 0.9318 |       |       |
|      | HCFlow [18]             |       |       | 4.40M | 36.29 | 0.9468 | 33.02 | 0.9065 | 31.74 | 0.8864 | 31.62 | 0.9206 | 35.23 | 0.9346 |       |       |
|      | FGRN [17]               |       |       | 3.35M | 36.97 | 0.9505 | 33.77 | 0.9183 | 31.83 | 0.8907 | 31.91 | 0.9253 | 35.15 | 0.9322 |       |       |
|      | IRN†                    |       |       | 5.44M | 36.20 | 0.9445 | 32.33 | 0.8986 | 31.64 | 0.8808 | 31.51 | 0.9152 | 35.07 | 0.9308 |       |       |
|      | DLV-IRN (ours)          |       |       | 5.49M | 36.62 | 0.9484 | 33.26 | 0.9093 | 32.05 | 0.8893 | 32.26 | 0.9253 | 35.55 | 0.9363 |       |       |

IRS\(\dagger\) is a variant of IRN with increased model depth for fair comparison with DLV-IRN in terms of number of parameters.

4.3 Experiments on Image Rescaling

Dual latent variable enhancement of IRN. For quantitative comparison in restored HR image quality in Table 2, IRN is used as the primary baseline model for dual latent variable enhancement. The Type I category includes image SR models optimized for up-scaling only. They are separately listed from bidirectional models in Type II category for fair comparison, as the latter ones have the advantage of jointly optimizing downsampling and upsampling which is demonstrated in the big difference in PSNR and SSIM between two categories. For Type II models, our enhanced DLV-IRN is consistently better than other INN based models which has only the default upsampling latent variable z, including the retrained IRN\(\dagger\) which is equivalent with DLV-IRN in model size. Comparing to the latest FGRN which does not have latent variables so that our enhanced DLV-IRN is still better in large test sets like BSD100, Urban100 and DIV2K overall, while trailing behind in smaller test sets. From visual examples shown in Fig. 3,

Figure 3: Visual comparisons of upsampling the ×4 downsampled images.

for IRN, fixing \(\hat{z}\) as a constant zero \(\hat{z}_0\) for both training and testing could achieve equivalent performance. In the case of our DLV-IRN, using constant \(\hat{z}_0\) is also beneficiary for stable training as random sampling of both \(\hat{z}\) and \(w\) could cause oscillation in joint optimization. For comparison, an ablation study is conducted to compare the effects of different settings in both \(\hat{z}\) and \(w\), where \(\hat{z}_0\) and \(w_0\) refer to the constant zero and \(\hat{z}_N\) and \(w_N\) represent random sampling from a normal distribution. All model variants here are trained for 250k iterations with an initial learning rate of \(2 \times 10^{-4}\), halved after every 50K iterations. As shown in Table 1, using constant \(\hat{z}_0\) is clearly better than \(\hat{z}_N\), while random sampling of \(w_N\) is slightly better than \(w_0\). Additionally, for different latent variable channel number \(C_w\), it is shown that there is consistent performance gain when \(C_w\) increases from 1 to 2, but the overall performance drops when it is further increased to 3. Lastly, the addition of \(L_3\) loss, which is only implemented for DLV-IRN, is shown to further improve HR reconstruction performance. Due to limited space, results with \(L_3\) included in Table 1 are for the default setting of \(w_N\), \(\hat{z}_0\) and \(C_w = 2\) only.
As shown in Table 3, for certain training set, there is marginal improvement of visual examples in Fig. 4, it is also clear that DLV-HCFlow restores images are of high qualities, a worst-case example from DIV2K reference image quality assessment like SSIM and no reference this, LR outputs from different methods are compared in both full downsampling LR and free of obvious visual artifacts. Considering this, the SSIM metric is calculated twice for Y channel only and RGB channels respectively. As shown in Table 4, the difference between two SSIM values is minimum for CAR, which is consistent with visual examples in Fig. 5 where no false color artifacts noticed for CAR. In contrast, this difference is very large for the HCFlow family, as their false color effects are the worst as demonstrated in Fig. 5. Results of the HCFlow family are also worse than the IRN ones in both NIQE and P IQE. For DLV-IRN and DLV-HCFlow, the difference between them and the corresponding IRN$^7$ and HCFlow$^7$ are minor, as demonstrated in both qualitative results in Fig. 5 and quantitative values in Table 4. In summary, the performance enhancement in HR restoration brought by dual latent variables is achieved without sacrificing image quality in downscaled LR. In contrast, the improvement of HCFlow over IRN is accompanied with significant degradation in LR. While FGRN is the best in PIQE, there are no SSIM values and downscales images available for more comprehensive comparison.

### Dual latent variable enhancement of HCFlow.

The dual latent variable enhancement is also applied to HCFlow to train a new model DLV-HCFlow. In addition to validating the general effectiveness of DLV-enhancement on different models, this experiment is summarized separately from the main one for a couple of reasons. First, the baseline HCFlow is trained from DF2K, a much larger dataset than the DIV2K as used in IRN. To study the effects of different training set sizes, all three models, including HCFlow, its variant HCFlow$^7$ with increased model depth and the enhanced DLV-HCFlow, are all trained with DF2K and DIV2K respectively. As shown in Table 3, for certain training set, there is marginal improvement from HCFlow to HCFlow$^7$ due to increased model depth. For comparison between the two training sets, it shows that results from DIV2K are at least as good as those from DF2K. This is in contrast to what is commonly observed from other image restoration models like ESRGAN [30] where increased training set size lead to performance improvement in general. Secondly, it is found out that HCFlow, and accordingly HCFlow$^7$ and DLV-HCFlow, uses a smaller weight for guidance loss $l_g$, which leads to poor image quality in downscaled LR images, as demonstrated later in Fig. 5. In other words, the performance improvement in HCFlow over IRN for upscaled HR images is accompanied with sacrifices in LR image quality. Nevertheless, the DLV enhancement is shown to be effective for HCFlow too. For DLV-HCFlow, there are additional improvements across different test sets comparing to HCFlow$^7$, although both are similar in model size. For the challenging case of Urban100, the additional increase in PSNR is as large as 0.31. From visual examples in Fig. 4, it is also clear that DLV-HCFlow restores fine details better and has less visible artifacts.

### Quality assessment of downscaled LR images.

Although there is no GT reference for downscaled LR in this study, it is important to generate LR images which are visually consistent with bicubicly downsampled LR and free of obvious visual artifacts. Considering this, LR outputs from different methods are compared in both full reference image quality assessment like SSIM and no reference ones like NIQE and PIQE. While the majority of generated LR images are of high qualities, a worst-case example from DIV2K validation set is selected here to demonstrate potential severity of visual artifacts. As shown in Fig. 5, when viewed in its native resolution, generated LR from IRN$^7$ and DLV-IRN look very similar to bicubic reference without obvious artifacts. For CAR, it is visibly brighter than normal overall. For the HCFlow family, false color and Moiré-like artifacts are noticeable even without zoomed-in. In magnified views, both false color and Moiré-like artifacts are very obvious, and the false color ones are more severe in general. Considering this, the SSIM metric is calculated twice for Y channel only and RGB channels respectively. As shown in Table 4, the difference between two SSIM values is minimum for CAR, which is consistent with visual examples in Fig. 5 where no false color artifacts noticed for CAR. In contrast, this difference is very large for the HCFlow family, as their false color effects are the worst as demonstrated in Fig. 5. Results of the HCFlow family are also worse than the IRN ones in both NIQE and PIQE. For DLV-IRN and DLV-HCFlow, the difference between them and the corresponding IRN$^7$ and HCFlow$^7$ are minor, as demonstrated in both qualitative results in Fig. 5 and quantitative values in Table 4. In summary, the performance enhancement in HR restoration brought by dual latent variables is achieved without sacrificing image quality in downscaled LR. In contrast, the improvement of HCFlow over IRN is accompanied with significant degradation in LR. While FGRN is the best in PIQE, there are no SSIM values and downscales images available for more comprehensive comparison.

### 4.4 Experiments on Image Hiding

Table 5 compares the results of DLV-HiNet with 4bit-LSB, HiD-DeN [37], Baluja [4], HiNet [13] and HiNet$^*$ quantitatively. For both cover/stego and secret/recovery image pairs of the three datasets, i.e., DIV2K, COCO and ImageNet, DLV-HiNet outperforms other methods in all four metrics. Note that the numerical numbers of HiNet are worse than the ones in the original paper because we add quantization step in the training phase. The visual comparisons are shown in Fig. 6. For better visual quality, we draw the heatmaps of the mean squared error between the ground truth and prediction. As can be seen, our DLV-HiNet can effectively reduce the errors in the edges and corners.

### 5 Conclusion

For bidirectional image rescaling models like IRN and HCFlow, a new latent variable is introduced to model the natural variations in image downsampling. Combining with the original latent variable that models the variations of high-frequency details in image upsampling, the dual latent variable enhancement is capable of further reducing expected restoration error in image upsampling. With minimum impact on model complexity, this newly proposed enhancement is shown to be able to improve restored HR image quality.
Table 4: Quantitative results of LR (×4) image quality by different downsampling methods.

| Methods      | Y-Set5 | Y-Set14 | BSD500 | Urban100 | DIV2K  |
|--------------|--------|---------|---------|----------|--------|
|              | SSIM   | NIQE    | PSNR    | SSIM     | NIQE   | PSNR   |
| Bicubic -    | -      | -       | 18.75   | 32.04    | -      | -      |
| CAR          | 0.9028 | 0.9352  | 18.875  | 59.782   | 0.9388 | 0.9303 |
| HCFlow       | 0.9846 | 0.9200  | 18.875  | 49.174   | 0.9766 | 0.8799 |
| DLV-HCFlow   | 0.9977 | 0.9293  | 18.875  | 43.697   | 0.9609 | 0.8877 |
| IRN†         | 0.9960 | 0.9823  | 18.875  | 44.920   | 0.9926 | 0.9705 |
| DLV-IRN      | 0.9961 | 0.9833  | 18.875  | 48.822   | 0.9937 | 0.9770 |
| YRGBIRN†     | -      | 0.9676  | 18.768  | 49.627   | -      | -      |

Table 5: Image hiding results of different methods on benchmark datasets.

| Methods     | Param  | Cover/Stego image pair | DIV2K | COCO | ImageNet |
|-------------|--------|------------------------|-------|------|----------|
|              |        |                        | SSIM  | MAE  | RMSE     | SSIM  | MAE  | RMSE     |
| 4bit-LSB    | 24.6M  | -                      | 0.9995| 1.32 | 1.93     | 0.9965| 1.40 | 2.01     |
| HiDDeN [37] | -      | -                      | 0.9965| 1.40 | 1.47     | 0.9769| 3.49 | 4.85     |
| Baluja [4]   | 42.4M  | -                      | 0.9965| 1.40 | 1.47     | 0.9769| 3.49 | 4.85     |
| HiNet [13]   | 4.1M   | -                      | 0.9965| 1.40 | 1.47     | 0.9769| 3.49 | 4.85     |
| HCFlow†      | 0.9517 | -                      | 0.9965| 1.40 | 1.47     | 0.9769| 3.49 | 4.85     |
| DLV-HiNet    | 4.5M   | -                      | 0.9965| 1.40 | 1.47     | 0.9769| 3.49 | 4.85     |

Figure 5: Visual examples of downscaled (×4)LR images, selected to demonstrate visual artifacts in worst cases.

Figure 6: Visual comparison of MSE heatmap images between the ground truth and predictions.
REFERENCES

[1] Eirikur Agustsson and Radu Timofte. 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. 126–135.

[2] Lynton Ardizzone, Carsten Lüth, Jakob Kruse, Carsten Rother, and Ullrich Kühne. 2018. Analyzing inverse problems with invertible neural networks. arXiv preprint arXiv:1808.04750 (2018).

[3] Lynton Ardizzone, Carsten Lüth, Jakob Kruse, Carsten Rother, and Ullrich Kühne. 2019. Guided image generation with conditional invertible neural networks. arXiv preprint arXiv:1907.02932 (2019).

[4] Shumeet Baluja. 2017. Hiding images in plain sight: Deep steganography. Advances in neural information processing systems 30 (2017).

[5] Mauro Barni, Franco Bartolini, and Alessandro Piva. 2001. Improved wavelet-based watermarking through pixel-wise masking. IEEE transactions on image processing 10, 5 (2001), 783–791.

[6] Jens Behrmann, Will Gratwohl, Ricky TQ Chen, David Duvenaud, and Jörn-Henrik Jacobsen. 2019. Invertible residual networks. In International Conference on Machine Learning. PMLR, 573–582.

[7] Marco Bevilacqua, Aline Roumy, Christine Guilleumot, and Marie line Alberi Morel. 2012. Low–Complexity Single-Image Super-Resolution based on Nonnegative Neighbor Embedding. In Proceedings of the British Machine Vision Conference. BMVA Press, 135.1–135.16.

[8] Tao Dai, Jianrui Cai, Yongbing Zhang, Shu-Tao Xia, and Lei Zhang. 2019. Second-order attention network for image super-resolution. In Proceedings of the IEEE conference on computer vision and pattern recognition. 11065–11074.

[9] Laurent Dinh, David Krueger, and Yoshua Bengio. 2014. Nice: Non-linear indepen-dent components estimation. arXiv preprint arXiv:1412.8506 (2014).

[10] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. 2016. Density estimation using real nvp. arXiv preprint arXiv:1605.08803 (2016).

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

[12] Jia-Bin Huang, Abhishek Singh, and Narendra Ahuja. 2015. Single image super-resolution using very deep residual channel attention networks. In International conference on curves and surfaces. Springer, 126–144.

[13] Mingqing Xiao, Shuxin Zheng, Chang Liu, Yaolong Wang, Di He, Guolin Ke, Jiang Biao, Zhouchen Lin, and Tie-Yan Liu. 2017. NTIRE 2017 challenge on single image super-resolution: Methods and results. In Proceedings of the IEEE conference on computer vision and pattern recognition workshops. 114–125.

[14] Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. 2015. ImageNet large scale visual recognition challenge. International journal of computer vision 115, 3 (2015), 211–252.

[15] Junpeng Jing, Xin Deng, Mai Xu, Jianyi Wang, and Zhenyu Guan. 2021. HiNet: Deep Image Hiding by Invertible Network. In Proceedings of the IEEE/CVF International Conference on Computer Vision. 4733–4742.

[16] Heewon Kim, Myungsub Choi, Bee Lim, and Kyoung Mu Lee. 2018. Task-aware image downscaling. In Proceedings of the European Conference on Computer Vision (ECCV). 399–414.

[17] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. 2016. Deeply-recurrent convolutional network for image super-resolution. In Proceedings of the IEEE conference on computer vision and pattern recognition. 1637–1645.

[18] Durk P Kingma and Prafulla Dharwadker. 2018. Glow: Generative flow with invertible 1x1 convolutions. Advances in neural information processing systems 31 (2018).

[19] Shah Li, Guixuan Zhang, Zhengxiong Luo, Jie Liu, Zhi Shen, and Shuwu Zhang. 2021. Approaching the Limit of Image Rescaling via Flow Guidance. arXiv preprint arXiv:2111.05133 (2021).

[20] Jingyun Liang, Andreas Lugmayr, Martin Danelljan, Luc Van Gool, and Radu Timofte. 2020. SFlow: Learning the Super-Resolution Space with Normalizing Flow. In ECCV.

[21] David Martin, Charles Fowlkes, Damin Bavishi, Wren Lavren, et al. 2015. Imagenet large scale visual recognition challenge. International journal of computer vision 115, 3 (2015), 211–252.

[22] Anish Mittal, Rajiv Soundararajan, and Alan C Bovik. 2012. Making a completely blind image quality analyzer. IEEE Signal processing letters 20, 3 (2012), 209–212.

[23] David Martin, Charles Fowlkes, Damin Bavishi, Wren Lavren, et al. 2015. Imagenet large scale visual recognition challenge. International journal of computer vision 115, 3 (2015), 211–252.

[24] Anish Mittal, Rajiv Soundararajan, and Alan C Bovik. 2012. Making a completely blind image quality analyzer. IEEE Signal processing letters 20, 3 (2012), 209–212.

[25] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. 2015. ImageNet large scale visual recognition challenge. International journal of computer vision 115, 3 (2015), 211–252.

[26] Tian Sun and Zhehong Chen. 2020. Learned image downscaling for upscaling using content adaptive resampler. IEEE Transactions on Image Processing 29 (2020), 4027–4040.

[27] Abdel-Fattah A Tamimi, Ayman M Abdalla, and Omaima Al-Allaf. 2013. Hiding an image inside another image using variable-rate steganography. International Journal of Advanced Computer Science and Applications (IJACSA) 4, 10 (2013).

[28] Radu Timofte, Eirikur Agustsson, Luc Van Gool, Ming-Hsuan Yang, and Lei Zhang. 2017. NTIRE 2017 challenge on single image super-resolution: Methods and results. In Proceedings of the IEEE conference on computer vision and pattern recognition workshops. 114–125.

[29] H. V. Thi, Ph. Tran, F. Fleuret, Mathis Chalain, Madhurima Chandrasekhar Bl, Sumo Hernandez, S. Chan-nappaya, and Swaparup S. Medasani. 2015. Blind image quality evaluation using perception based features. In Twenty First National Conference on Communications (NCC). IEEE, 1–6.

[30] Xintao Wang, Ke Yu, Shixiang Wu, Jinjin Gu, Yihao Liu, Chao Dong, Yu Qiao, and Chen Change Loy. 2018. ESRGAN: Enhanced super-resolution generative adversarial networks. In Proceedings of the European Conference on Computer Vision (ECCV). 63–79.

[31] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. 2004. Image quality assessment: from error visibility to structural similarity. IEEE transactions on image processing 13, 4 (2004), 600–621.

[32] Mingming Xiao, Shuxin Zheng, Chang Liu, Yaolong Wang, Di He, Guolin Ke, Jiang Biao, Zhouchen Lin, and Tie-Yan Liu. 2017. Invertible image rescaling. In Proceedings of the IEEE conference on computer vision and pattern recognition. 2472–2481.

[33] Mingming Xiao, Shuxin Zheng, Chang Liu, Yaolong Wang, Di He, Guolin Ke, Jiang Biao, Zhouchen Lin, and Tie-Yan Liu. 2020. Invertible image rescaling. In European Conference on Computer Vision. Springer, 126–144.

[34] Charless Fowlkes, Doron Tal, and Jitendra Malik. 2001. A database of image objects for image data mining. In Proceedings Eighth IEEE International Conference on Computer Vision. IEEE, 1–6.

[35] Roman Zeyde, Michael Elad, and Matan Protter. 2010. On single image scale-up using sparse-representations. In International conference on curves and surfaces. Springer, 711–730.

[36] Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. 2015. ImageNet large scale visual recognition challenge. International journal of computer vision 115, 3 (2015), 211–252.

[37] Radu Timofte, Eirikur Agustsson, Luc Van Gool, Ming-Hsuan Yang, and Lei Zhang. 2017. NTIRE 2017 challenge on single image super-resolution: Methods and results. In Proceedings of the IEEE conference on computer vision and pattern recognition workshops. 114–125.