

Hierarchical Conditional Flow: A Unified Framework for Image Super-Resolution and Image Rescaling

Jingyun Liang¹ Andreas Lugmayr¹ Kai Zhang^{1,*} Martin Danelljan¹ Luc Van Gool^{1,2} Radu Timofte¹

¹Computer Vision Lab, ETH Zurich, Switzerland ² KU Leuven, Belgium

{jinliang, andreas.lugmayr, kai.zhang, martin.danelljan, vangool, timofte}@vision.ee.ethz.ch

<https://github.com/JingyunLiang/HCFLOW>

Abstract

Normalizing flows have recently demonstrated promising results for low-level vision tasks. For image super-resolution (SR), it learns to predict diverse photo-realistic high-resolution (HR) images from the low-resolution (LR) image rather than learning a deterministic mapping. For image rescaling, it achieves high accuracy by jointly modelling the downscaling and upscaling processes. While existing approaches employ specialized techniques for these two tasks, we set out to unify them in a single formulation. In this paper, we propose the hierarchical conditional flow (HCFLOW) as a unified framework for image SR and image rescaling. More specifically, HCFLOW learns a bijective mapping between HR and LR image pairs by modelling the distribution of the LR image and the rest high-frequency component simultaneously. In particular, the high-frequency component is conditional on the LR image in a hierarchical manner. To further enhance the performance, other losses such as perceptual loss and GAN loss are combined with the commonly used negative log-likelihood loss in training. Extensive experiments on general image SR, face image SR and image rescaling have demonstrated that the proposed HCFLOW achieves state-of-the-art performance in terms of both quantitative metrics and visual quality.

1. Introduction

Normalizing flows [7, 8, 22, 12, 16, 36] are powerful deep generative probabilistic models that allow for efficient and exact likelihood calculation and sampling. They have been used in the generation of image [8, 22], blur kernel [28], and audio [20] data. Recently, in the low-level vision community, normalizing flows have attracted much interest and have achieved promising progress for image super-resolution (SR) [32] and image rescaling [47].

*Corresponding author.

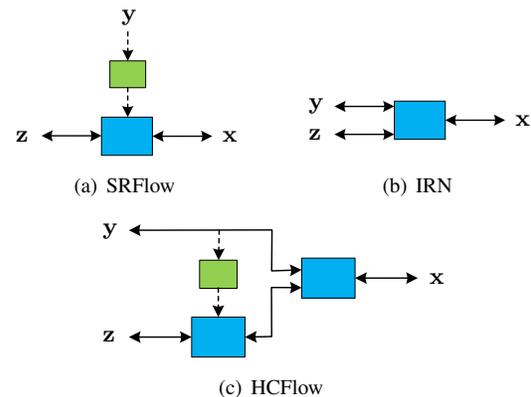


Figure 1: The comparison between SRFlow [32], IRN [47] and the proposed HCFLOW. x , y and z denote HR image, LR image and the latent variable, respectively. Blue boxes are invertible neural networks, while green ones are non-invertible models (e.g., CNN). Solid bi-directional arrows denote bijective mappings, while dashed arrows represent conditional relations.

SRFlow [32] is a seminal flow-based model for image SR. Unlike previous CNN-based models that learn a deterministic mapping from the low-resolution (LR) image to the high-resolution (HR) image, SRFlow learns the distribution of HR images and is able to generate diverse photo-realistic HR images. However, as shown in Fig. 1(a), it treats the LR image as an external conditional prior and thus is not fully invertible between HR and LR image pairs, making it hard to be used for image rescaling. Another work IRN [47] employs an invertible neural network to learn downscaling and upscaling for image rescaling. Since the model is bijective, it can recover the input HR image with high accuracy after downscaling. Nevertheless, as shown in Fig. 1(b), it assumes the high-frequency and low-frequency components of the image are independent of each other and thus lacks the ability to exploit their dependency for image SR.

In this paper, we propose a hierarchical conditional flow (HCFLOW) as a unified framework for both image SR and rescaling. As shown in Fig. 1(c), HCFLOW is an invertible flow-based model for modelling the HR-LR relationship,

in which the high-frequency component is hierarchically conditional on the low-frequency component of the image. More specifically, in the forward propagation, HCFlow learns to decompose the input HR image into the LR image and a latent variable. In the inverse propagation, it generates HR images based on the LR input and random samples of the latent variable. The modelling of the latent variable (high-frequency component) is conditional on the generated LR image (low-frequency component) in a hierarchical manner.

When trained for image SR, HCFlow is optimized by minimizing the negative log-likelihood loss on the basis of tractable Jacobian determinant computation. To further improve visual quality, we integrate a pixel loss, perceptual loss, and GAN loss in the inverse propagation to constrain the learned HR space. Moreover, HCFlow can be used for the image rescaling task. It can decompose the HR image to a visually-pleasing LR image and a latent variable that follows a simple distribution. In this case, HCFlow is trained as an encoder-decoder framework, in which the forward and inverse processes are jointly optimized. As HCFlow is bijective, it can recover the HR image faithfully by sampling from the latent space given the generated LR image.

Our contributions can be summarized as follows:

- 1) We propose a unified framework for image SR and image rescaling. It learns to model the LR image and the residual high-frequency component simultaneously. The high-frequency component is hierarchically conditional on the generated LR image.
- 2) We propose additional losses to train normalizing flows, including pixel, perceptual, and GAN losses, which effectively enhances the HR image quality.
- 3) We perform extensive experiments on three tasks: general image SR, face image SR and image rescaling. HCFlow achieves state-of-the-art results on all tasks in terms of both quantitative metrics and visual quality.

2. Related Work

In this section, we will briefly review image SR and image rescaling with a particular focus on two highly related flow-based methods, *i.e.*, SRFlow [32] and IRN [47].

2.1. Image SR

Image SR aims to reconstruct the HR image given the LR image. Since the pioneer work SRCNN [9], many CNN-based models have been proposed in recent years [9, 19, 24, 43, 41, 40, 42, 15, 6, 6, 54, 55, 30, 49, 11, 5, 44, 26, 51, 27, 50]. Most of them focus on delicate feature extraction module design and generate over-smoothed images when trained with the pixel loss. To remedy this, the perceptual loss [17, 43] and GAN loss [10, 24, 43, 53] are introduced to improve the perceptual quality. Despite of above progresses, they usually learn a deterministic mapping between

the LR image and HR image, which is unnatural for image SR since one LR image may correspond to multiple HR images.

SRFlow [32]. Normalizing flows [7, 8, 22, 12, 16, 36, 46] provide a new possible solution for image SR. SRFlow designs a conditional flow to model the distribution of HR images, conditional on LR images. It can generate diverse photo-realistic images by sampling different latent variables. Our proposed HCFlow differs from SRFlow in two main aspects: First, SRFlow uses the LR image as an external conditional prior and maps the HR distribution to a simple latent distribution. Therefore, it cannot generate LR image and thus is not applicable for image rescaling. In contrast, HCFlow models the LR image and treats it as part of the latent space. Second, SRFlow basically follows the flow framework proposed in [8], while HCFlow proposes a new framework with hierarchical conditional mechanism.

2.2. Image Rescaling

Image rescaling aims to downscale the HR image to a visually meaningful LR image, and then recover the HR image plausibly. Different from image SR that works on a given LR image space, image rescaling tries to maintain as much information from the HR image as possible for a better subsequent reconstruction, for the purpose of reducing the storage and bandwidth cost. In other words, it can define its own LR image space which is expected to be more informative than that by simple downscaling such as bicubic downscaling. In general, in image rescaling, the downscaling and upscaling processes are jointly modelled by an encoder-decoder framework [18, 25, 38], so that the downscaling model is optimized for the later upscaling operation.

IRN [47]. Recently, IRN proposes to use a bijective invertible neural network to model the downscaling and upscaling processes. High-frequency component is well-captured and transformed to a structured latent space in training. In testing, the HR image can be recovered by inputting the generated LR image and a randomly sampled latent variable. In particular, IRN assumes the LR image and high-frequency component is independent to each other. These two components are divided apart and learned separately. By contrast, HCFlow assumes the removed high-frequency component is dependent on the LR image and thus employs a hierarchical conditional framework to model the LR image and the conditional distribution of the high-frequency component. Besides, although IRN designs a bijective mapping between HR and LR image pairs, it can only be trained by Monte Carlo simulation rather than maximum likelihood estimation (MLE). HCFlow can be trained in the same way for image rescaling, but it further models the LR image distribution and allows for tractable Jacobian determinant computation, making it possible for probabilistic modelling of HR and LR images when trained by MLE.

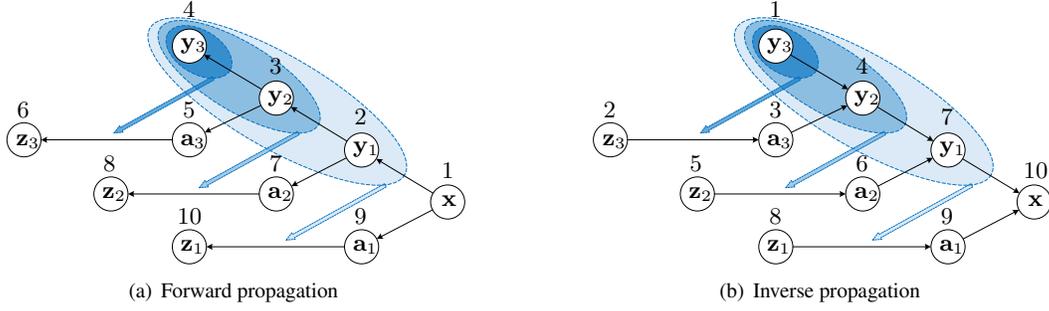


Figure 2: Schematic computational graphs of the hierarchical conditional flow (HCFlow) with 3 flow levels. On level l , y_{l-1} (note that $y_0 = \mathbf{x}$) is decomposed to low-frequency component y_l and high-frequency component \mathbf{a}_l . The transformation between \mathbf{a}_l and \mathbf{z}_l is conditional on $[y_L, y_{L-1}, \dots, y_l]$, as indicated by the blue arrows. The computation orders in forward and inverse propagation are shown on the top of each node.

3. Methodology

3.1. Preliminaries

Flow-based models [7, 8, 23, 37, 13, 22, 2, 16, 12, 36, 28] aim to learn a bijective mapping between the target space and the latent space. For a high-dimensional random variable (e.g., an image) \mathbf{x} with distribution $\mathbf{x} \sim p(\mathbf{x})$ and a latent variable \mathbf{z} with simple tractable distribution $\mathbf{z} \sim p(\mathbf{z})$ (e.g., multivariate Gaussian distribution), flow models generally use an invertible neural network f_θ to transform \mathbf{x} to \mathbf{z} : $\mathbf{z} = f_\theta(\mathbf{x})$. Conversely, \mathbf{x} can be recovered from \mathbf{z} by the inverse mapping $\mathbf{x} = f_\theta^{-1}(\mathbf{z})$.

Generally, f_θ is composed of a series of invertible transformations: $f_\theta = f_\theta^1 \circ f_\theta^2 \circ \dots \circ f_\theta^K$. The intermediate variables are defined as $\mathbf{h}^k = f_\theta^k(\mathbf{h}^{k-1})$ for $k \in \{1, \dots, K\}$. The input \mathbf{h}^0 and output \mathbf{h}^N of f_θ are \mathbf{x} and \mathbf{z} , respectively. Concretely, f_θ^k are flow layers such as squeeze layer, batch normalization layer, affine coupling layer, etc.

According to the change of variable formula and the chain rule, for a sample \mathbf{x} , the log probability $\log p(\mathbf{x})$ can be calculated as

$$\log p(\mathbf{x}) = \log p(f_\theta(\mathbf{x})) + \sum_{k=1}^K \log \left| \det \frac{\partial f_\theta^k(\mathbf{h}^{k-1})}{\partial \mathbf{h}^{k-1}} \right|, \quad (1)$$

where $\log \left| \det \frac{\partial f_\theta^k(\mathbf{h}^{k-1})}{\partial \mathbf{h}^{k-1}} \right|$ is the logarithm of the absolute value of the determinant of the Jacobian of f_θ^k at \mathbf{h}^{k-1} . The flow model can thereby be optimized by minimizing the negative log-likelihood loss.

3.2. Model Specification

Both image SR and image rescaling try to reconstruct the HR image \mathbf{x} given a LR image. Since the image degradation process (or image downscaling) is the inverse of image super-resolution (or image upscaling), we can model these two processes with an invertible bijective transformation: $\mathbf{x} \leftrightarrow [\mathbf{y}, \mathbf{a}]$, where \mathbf{y} and \mathbf{a} are the generated LR image and

the rest high-frequency component, respectively. As modelling the probability of natural images is a non-trivial task, it is reasonable to design a flow model conditional on the ground-truth LR image \mathbf{y}^* as,

$$p(\mathbf{x}|\mathbf{y}^*) \leftrightarrow p(\mathbf{y}, \mathbf{a}|\mathbf{y}^*) = p(\mathbf{y}|\mathbf{y}^*)p(\mathbf{a}|\mathbf{y}, \mathbf{y}^*). \quad (2)$$

Ideally, we hope the model can generate exactly the same LR image as the ground-truth LR image. This can be formulated as a Dirac delta function $\delta(\mathbf{y} - \mathbf{y}^*)$ and further approximated by a multivariate Gaussian distribution as,

$$\begin{aligned} p(\mathbf{y}|\mathbf{y}^*)p(\mathbf{a}|\mathbf{y}, \mathbf{y}^*) &= \delta(\mathbf{y} - \mathbf{y}^*)p(\mathbf{a}|\mathbf{y}) \\ &= \lim_{\Sigma \rightarrow \mathbf{0}} \mathcal{N}(\mathbf{y}|\mathbf{y}^*, \Sigma)p(\mathbf{a}|\mathbf{y}), \end{aligned} \quad (3)$$

where Σ is a diagonal covariance matrix with all diagonal elements close to zero. Note that \mathbf{y} is nearly equal to \mathbf{y}^* in this case. By further mapping $p(\mathbf{a}|\mathbf{y})$ to a standard multivariate Gaussian distribution $p(\mathbf{z}) = \mathcal{N}(\mathbf{z}|\mathbf{0}, \mathbf{I})$, the flow model is defined as,

$$p(\mathbf{x}|\mathbf{y}^*) \leftrightarrow \lim_{\Sigma \rightarrow \mathbf{0}} \mathcal{N}(\mathbf{y}|\mathbf{y}^*, \Sigma)\mathcal{N}(\mathbf{z}|\mathbf{0}, \mathbf{I}). \quad (4)$$

As we can see, part of the latent space is constrained to be the LR image space. In particular, decomposed high-frequency component \mathbf{a} is conditional on another decomposed component \mathbf{y} . Once trained, following the forward direction, HCFlow can decompose the HR image \mathbf{x} into LR image \mathbf{y} and latent variable \mathbf{z} that follows a simple distribution. Following the inverse direction, HCFlow can generate \mathbf{x} given the LR image input \mathbf{y}^* and a random sample \mathbf{z} from the latent distribution, as it is an invertible bijective model.

Note that this model regards \mathbf{y}^* as an input or output, rather than as an external conditional prior. Therefore, it is not explicitly conditional on \mathbf{y}^* and is fully invertible between HR and LR image pairs. Besides, by approximating the distribution of \mathbf{y} with a multi-variate Gaussian distribution, it allows for tractable Jacobian determinant computation, so that the model can be optimized by maximum likelihood estimation (MLE).

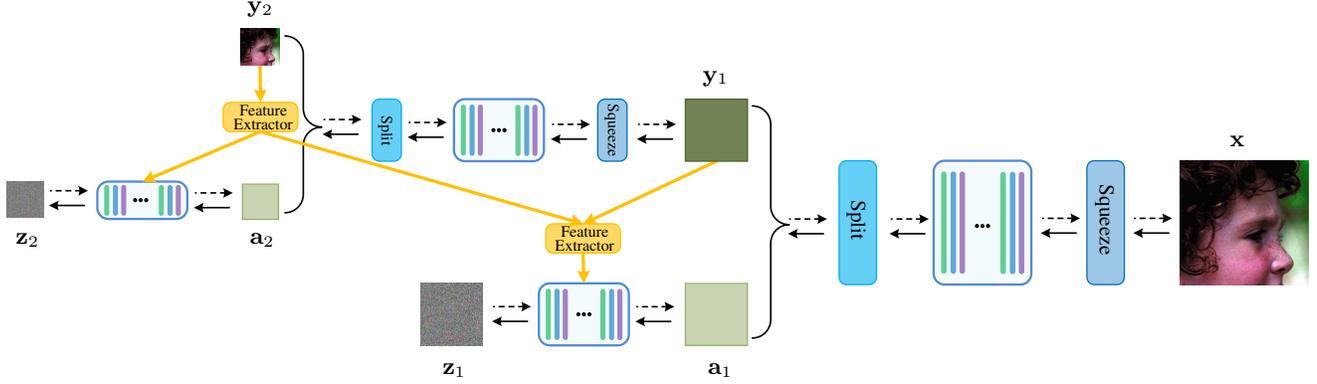


Figure 3: The architecture of the hierarchical conditional flow (HCFlow) with 2 flow levels. For a HR image \mathbf{x} , we first squeeze, transform and split it to low-frequency component \mathbf{y}_1 and high-frequency component \mathbf{a}_1 . Similarly, \mathbf{y}_1 is decomposed to \mathbf{y}_2 (*i.e.*, the LR image in this case) and \mathbf{a}_2 in the next level. \mathbf{a}_1 and \mathbf{a}_2 are transformed to latent variables \mathbf{z}_1 and \mathbf{z}_2 , conditional on $\phi_1([\phi_2(\mathbf{y}_2), \mathbf{y}_1])$ and $\phi_2(\mathbf{y}_2)$ (note that ϕ_1 and ϕ_2 are feature extractors, *e.g.*, CNN) respectively, in a hierarchical manner. The model is trained by negative log-likelihood loss, and can be further enhanced by pixel loss, perceptual loss and GAN loss.

3.3. Model Architecture

The multi-scale architecture proposed in RealNVP [8] is a popular normalizing flow architecture [32, 22, 12]. It consists of L levels and at the end of each level, half of the dimensions are factored out. Generally, the factored out dimensions are directly Gaussianized for the computation of negative log-likelihood loss, lacking sufficient modelling of these dimensions. Therefore, based on the multi-scale architecture, we take a further step to model factored out dimensions conditional on the reserved dimensions.

As illustrated in Fig. 2, at each level l , \mathbf{y}_{l-1} is decomposed to low-frequency component \mathbf{y}_l and high-frequency component \mathbf{a}_l . Then, \mathbf{a}_l is modelled by an additional flow that is conditional on the concatenation of tensors $\mathbf{y}_L, \mathbf{y}_{L-1}, \dots, \mathbf{y}_l$ from multiple flow levels. By this design, the reconstruction of high-frequency component is hierarchically conditional on frequencies reconstructed from all previous levels. In forward propagation, similar to the depth-first traversal of a binary tree, we first compute $\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_L$ in order. Then, we model the factored out dimensions in a reverse order: $\mathbf{a}_L, \mathbf{a}_{L-1}, \dots, \mathbf{a}_1$. In inverse propagation, we compute \mathbf{y}_l and \mathbf{a}_l level by level, from level L to level 1. Note that the determinant of Jacobian of the whole flow can still be efficiently computed, since the conditional relations between $\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_L$ and $\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_L$ can be represented as an upper triangle block matrix.

The detailed architecture of HCFlow is shown in Fig. 3. For each level, the first layer is the squeeze layer, which transforms the $H \times W \times C$ input to a $\frac{H}{2} \times \frac{W}{2} \times 4C$ tensor by trading spatial size for number of channels. Then, K flow-steps are used for transforming the tensor and decomposing it into different components. More specifically, each flow-step consists of a sequence of three layers: Act-norm layer, invertible 1×1 convolution layer and affine coupling layer [8, 22]. After that, the split layer is used to

evenly split the tensor into two tensors $\mathbf{y}_l \in \mathcal{R}^{\frac{H}{2} \times \frac{W}{2} \times 2C}$ and $\mathbf{a}_l \in \mathcal{R}^{\frac{H}{2} \times \frac{W}{2} \times 2C}$ along the channel dimension. Note that, for the last level, we only keep 3 channels for \mathbf{y}_l to make it fit the RGB space of the LR image. Next, \mathbf{y}_l is fed to the next level, while \mathbf{a}_l is input into an additional flow.

In the l -th additional flow, \mathbf{a}_l is transformed to the latent variable \mathbf{z}_l by P flow-steps. Different from above flow-steps, we use conditional affine coupling layer [3, 45] rather than ordinary affine coupling layer to obtain a conditional flow. In particular, we first upscale the conditional feature \mathbf{c}_{l+1} from level $l+1$ by $\times 2$ nearest neighbor interpolation, and concatenate it with \mathbf{y}_l . Then, we use a feature extractor ϕ_l to extract image features, which act as the conditional feature \mathbf{c}_l for level l . Note that the feature extractor only provides scale and shift for an affine coupling during both forward and inverse propagation. Hence the constrains on being invertible and having a tractable Jacobian do not hold for this part. More formally, the hierarchical conditional mechanism of HCFlow is formulated as follows,

$$\mathbf{c}_l = \begin{cases} \phi_l(\mathbf{y}_l) & l = L \\ \phi_l([\mathbf{c}_L, \mathbf{c}_{L-1}, \dots, \mathbf{c}_{l+1}, \mathbf{y}_l]) & l = L-1, \dots, 1 \end{cases}, \quad (5)$$

where conditional features of different levels are computed in a reverse order, from \mathbf{c}_L to \mathbf{c}_1 .

Particularly, for the last level, we directly model \mathbf{y}_L by a Dirac delta function $\delta(\mathbf{y} - \mathbf{y}^*)$ instead of transforming it to another latent variable. This constrains part of the latent space to be the LR image space and implicitly makes the model be conditional on \mathbf{y}^* .

3.4. Training Objectives

Image SR. When HCFlow is used for image SR, it can be trained by minimizing the negative log-likelihood loss,

$$\mathcal{L}_{nll} = -\log p(\mathbf{x}), \quad (6)$$

which is unsupervised and converges stably. However, in practice, this loss converges slowly and does not provide strong supervision for image SR. To achieve better HR image PSNR, we can add \mathcal{L}_1 pixel loss on the generated SR image in inverse propagation, leading to a loss function as follows,

$$\mathcal{L} = \lambda_1 \mathcal{L}_{nll}(\mathbf{x}) + \lambda_2 \mathcal{L}_{pixel}(\mathbf{x}, \mathbf{x}_{\tau=0}), \quad (7)$$

where \mathbf{x} is the ground-truth HR image and $\mathbf{x}_{\tau=0}$ is the generated SR image by inputting the ground-truth LR image \mathbf{y}^* and sampling the latent variable \mathbf{z} with temperature $\tau = 0$. The added pixel loss can help the flow to learn the SR manifold centered around the PSNR-oriented SR image. Furthermore, we can add perceptual loss [17] and GAN loss [10] on the generated SR image to improve the visual quality. This is formulated as,

$$\begin{aligned} \mathcal{L} = & \lambda_1 \mathcal{L}_{nll}(\mathbf{x}) + \lambda_2 \mathcal{L}_{pixel}(\mathbf{x}, \mathbf{x}_{\tau=0}) \\ & + \lambda_3 \mathcal{L}_{percep}(\mathbf{x}, \mathbf{x}_{\tau=\tau_0}) + \lambda_4 \mathcal{L}_{gan}(\mathbf{x}, \mathbf{x}_{\tau=\tau_0}), \end{aligned} \quad (8)$$

where $\mathbf{x}_{\tau=\tau_0}$ is the generated SR image by inputting \mathbf{y}^* and sampling \mathbf{z} with $\tau = \tau_0$. Note that unlike the pixel loss that uses $\tau = 0$, τ_0 is set to 0.8 or 0.9 to preserve the diversity of HR images.

Image rescaling. Different from image SR, image rescaling aims to recover exactly the same HR image. Following [47], we regard the invertible HCFlow as an encoder-decoder framework, in which the forward and inverse processes correspond to the encoding and decoding stages, respectively. The loss is as follows,

$$\begin{aligned} \mathcal{L} = & \lambda_1 \mathcal{L}_{pixel.hr}(\mathbf{x}, \mathbf{x}_{\tau=1}) + \lambda_2 \mathcal{L}_{pixel.lr}(\mathbf{y}^*, \mathbf{y}) \\ & + \lambda_3 \mathcal{L}_{latent}(\mathbf{z}), \end{aligned} \quad (9)$$

where $\mathcal{L}_{pixel.hr}$ is the \mathcal{L}_1 pixel loss to ensure that, after downscaling and upscaling, the reconstructed image $\mathbf{x}_{\tau=1}$ is close to the input \mathbf{x} . Note that this loss would dramatically decrease the diversity of generated images. Besides, $\mathcal{L}_{pixel.lr}$ is the \mathcal{L}_2 pixel loss on the LR image, which guides \mathbf{y} to be close to the bicubic LR image \mathbf{y}^* , so as to generate visually-pleasing LR images in downscaling. The last term $\mathcal{L}_{latent}(\mathbf{z})$ is the \mathcal{L}_2 regularization on the latent variable \mathbf{z} .

4. Experiments

4.1. Experimental Setup

We conduct experiments on general image SR, face image SR and image rescaling to show the effectiveness of HCFlow. For image SR experiments, we train the model by three loss combinations: \mathcal{L}_{nll} , $\mathcal{L}_{nll} + \mathcal{L}_{pixel}$ and $\mathcal{L}_{nll} + \mathcal{L}_{pixel} + \mathcal{L}_{percep} + \mathcal{L}_{gan}$. The corresponding learned models are denoted as **HCFlow**, **HCFlow+** and **HCFlow++**, respectively.

Image SR. For general image SR ($\times 4$), we set L, K, P to 2, 13 and 13, respectively. Two 13-block RRDB networks [43] are used as feature extractors. More details on the architecture are provided in the supplementary. The model is trained on the training set of DIV2K [1] and Flickr2K [39] with random flips. The crop patch size and mini-batch size are set to 160×160 and 16, respectively. Adam optimizer [21] with $\beta_1 = 0.9$ and $\beta_2 = 0.99$ is used for optimization. For HCFlow (with only \mathcal{L}_{nll}), the learning rate is 2.5×10^{-4} and reduced by half at 50%, 75%, 90% and 95% of 300k iterations. We fine-tune HCFlow+ (with $\mathcal{L}_{nll} + \mathcal{L}_{pixel}$) for 50k iterations from the pretrained HCFlow. The weight of \mathcal{L}_{nll} and \mathcal{L}_{pixel} are $\lambda_1 = 2 \times 10^{-3}$ and $\lambda_2 = 1$, respectively. It is worth pointing out that we can achieve even higher PSNR (about 0.2dB) if we train HCFlow+ from scratch. Similarly, we can fine-tune HCFlow++ by further adding \mathcal{L}_{percep} and \mathcal{L}_{gan} . The loss weighting parameters are $\lambda_1 = 2 \times 10^{-3}$, $\lambda_2 = 1$, $\lambda_3 = 5 \times 10^{-2}$ and $\lambda_4 = 5 \times 10^{-1}$.

For face image SR ($\times 8$), L, K, P are set to 3, 13 and 13, respectively. Three 8-block RRDB networks are used as feature extractors. We train the model on the CelebA training set [31] and test it using first 5,000 images from the testing set. Following [22, 32], we crop and resize the HR images to the resolution of 160×160 , and flip them randomly for data augmentation. Other training details are the same as general image SR.

Image rescaling. For image rescaling ($\times 4$), we set L, K, P to 2, 8 and 6, respectively. Two 3-block RRDB networks are used as feature extractors. In particular, we use Haar transformation to replace the squeeze layer and remove invertible 1×1 convolution layers. Details on data preparation and optimizer are the same as general image SR. The learning rate is initialized as 2.5×10^{-4} and halved at [100k, 200k, 300k, 400k] (500k iterations in total). The loss weighting parameters are $\lambda_1 = 1$, $\lambda_2 = 5 \times 10^{-2}$ and $\lambda_3 = 1 \times 10^{-5}$, respectively.

Performance evaluation. Following SRFlow [32] and IRN [47], we evaluate PSNR and SSIM on the RGB color space for image SR, and on the Y channel of the YCbCr color space for image rescaling. We also use perceptual metric LPIPS [52] and two no-reference metrics, NIQE [35] and BRISQUE [34], for better visual quality comparison. Pixel standard deviation of 5 samples is used to compare the diversity of results. In addition, Consistency (PSNR between the downscaled SR image and the ground-truth LR image) and LR-PSNR (PSNR between the generated LR image in forward propagation and the ground-truth LR image) are also reported.

4.2. Ablation Study

Fitting to the LR image space. To learn a fully invertible flow between HR and LR image pairs, HCFlow constrains

Table 1: Ablation study on latent space and conditional priors for general image SR ($\times 4$). Results are tested on DIV2K [1] validation set.

Case	Latent Space	Conditional Prior ($l = 2$)	Conditional Prior ($l = 1$)	PSNR \uparrow ($\tau = 0$)	SSIM \uparrow ($\tau = 0$)	LPIPS \uparrow ($\tau = 0.9$)	Consistency \uparrow ($\tau = 0.9$)	LR-PSNR \uparrow
1	$\mathbf{z}_2, \mathbf{z}_1$	-	-	4.76	0.34	0.863	10.56	-
2	$\mathbf{z}_2, \mathbf{z}_1$	\mathbf{y}^*	$\mathbf{y}^*, \mathbf{y}_1$	28.73	0.81	0.123	41.97	-
3	$\mathbf{y}_2, \mathbf{z}_2, \mathbf{z}_1$	\mathbf{y}^*	$\mathbf{y}^*, \mathbf{y}_1$	28.71	0.81	0.124	41.79	52.77
4	$\mathbf{y}_2, \mathbf{z}_2, \mathbf{z}_1$	-	-	18.95	0.47	0.361	40.79	53.88
5	$\mathbf{y}_2, \mathbf{z}_2, \mathbf{z}_1$	\mathbf{y}_2	\mathbf{y}_1	28.60	0.80	0.126	41.94	52.19
HCFLOW	$\mathbf{y}_2, \mathbf{z}_2, \mathbf{z}_1$	\mathbf{y}_2	$\mathbf{y}_2, \mathbf{y}_1$	28.71	0.81	0.124	42.01	53.37

Table 2: General image SR ($\times 4$) results on DIV2K [1] validation set. For SRFlow and our method, the mean results of 5 draws are reported.

Method	#Param	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow	NIQE \downarrow	BRISQUE \downarrow	Diversity \uparrow	Consistency \uparrow	LR-PSNR \uparrow
Bicubic	-	26.70	0.77	0.409	5.20	53.8	0	38.70	-
EDSR [29]	43.1M	28.98	0.83	0.270	4.46	43.3	0	54.89	-
RRDB [43]	16.7M	29.44	0.84	0.253	5.08	52.4	0	49.20	-
ESRGAN [43]	16.7M	26.22	0.75	0.124	2.61	22.7	0	39.03	-
RankSRGAN [53]	13.7M	26.55	0.75	0.128	2.45	17.2	0	42.33	-
SRFlow, $\tau = 0$ [32]	39.5M	29.07	0.81	0.254	5.20	39.4	0	55.13	-
SRFlow, $\tau = 0.9$ [32]	39.5M	27.09	0.76	0.121	3.57	17.8	5.6	49.96	-
HCFLOW , $\tau = 0$	23.2M	28.71	0.81	0.285	4.61	44.1	0	42.03	53.37
HCFLOW , $\tau = 0.9$	23.2M	27.02	0.76	0.124	2.79	21.7	4.8	42.01	53.37
HCFLOW+ , $\tau = 0$	23.2M	29.25	0.83	0.212	4.45	43.2	0	51.11	53.95
HCFLOW++ , $\tau = 0.9$	23.2M	26.61	0.74	0.110	2.85	22.0	5.2	50.07	52.59

Table 3: Face image SR ($\times 8$) results on CelebA [31] testing set. For SRFlow and our method, the mean results of 5 draws are reported.

Method	#Param	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow	NIQE \downarrow	BRISQUE \downarrow	Diversity \uparrow	Consistency \uparrow	LR-PSNR \uparrow
Bicubic	-	23.15	0.63	0.517	7.82	58.6	0	35.19	-
RRDB [43]	16.7M	26.59	0.77	0.230	6.02	49.7	0	48.22	-
ESRGAN [43]	16.7M	22.88	0.63	0.120	3.46	23.7	0	34.04	-
SRFlow, $\tau = 0$ [32]	40.0M	26.74	0.76	0.216	5.74	40.4	0	56.57	-
SRFlow, $\tau = 0.8$ [32]	40.0M	25.24	0.71	0.110	4.20	23.2	5.2	50.85	-
HCFLOW , $\tau = 0$	27.0M	26.66	0.77	0.210	6.42	48.0	0	51.83	54.50
HCFLOW , $\tau = 0.8$	27.0M	24.99	0.71	0.104	4.34	31.6	5.9	51.81	54.50
HCFLOW+ , $\tau = 0$	27.0M	27.02	0.78	0.212	6.04	49.5	0	51.11	53.95
HCFLOW++ , $\tau = 0.8$	27.0M	24.83	0.69	0.090	3.87	23.8	4.0	51.57	51.82

part of the latent space to be the LR image space, instead of using the LR image as an external prior. To show the impact, we remove the LR image \mathbf{y}_2 from the latent space as shown in case 1 and 2 of Table 1. When there is no conditional prior (case 1), the model fails to converge as it does not have enough information for SR. When we replace \mathbf{y}_2 with ground-truth LR image \mathbf{y}^* as a conditional prior (case 2, similar to SRFlow [32]), it achieves slightly better performance than HCFLOW although they have almost the same conditional information. The underlying reason might be that it has a larger latent space than HCFLOW.

Ground-truth LR image as a conditional prior. HCFLOW is conditional on \mathbf{y}_1 and \mathbf{y}_2 , which are generated during propagation. When we use the ground-truth LR image \mathbf{y}^* as a conditional prior to replace \mathbf{y}_2 (case 3, Table 1), the model achieves similar performance as HCFLOW. In fact, since we model the distribution of \mathbf{y}_2 as a Dirac delta function $\delta(\mathbf{y}_2 - \mathbf{y}^*)$, \mathbf{y}_2 would be nearly equal to \mathbf{y}^* after model convergence, which is confirmed by the high LR-PSNR. Therefore, conditional on the generated \mathbf{y}_2 and the external \mathbf{y}^* have similar effects.

Hierarchical conditional mechanism. As shown in case 4 of Table 1, similar to IRN [47], we assume the LR image and the rest high-frequency component is independent by removing all conditional priors. It yields significantly worse performance because the reconstruction of HR image (high-frequency component) is highly conditional on the LR image (low-frequency component) for image SR. Despite this, it has better results than case 1, as fitting to the LR image space could partly play the role of conditional prior. In case 5, we change from hierarchical conditional mechanism to single-scale conditional mechanism, by removing \mathbf{y}_2 from level 1. In this case, \mathbf{z}_l ($l = 1, 2$) is only conditional on \mathbf{y}_l from the same level. The performance drops in terms of all kinds of metrics, which shows that the hierarchical conditional mechanism can better model the conditional relations between high-frequency and low-frequency components.

4.3. Experiments on Image SR

General image SR. For general image SR ($\times 4$), we compare HCFLOW with state-of-the-art CNN-based and flow-based SR models, including the PSNR-oriented EDSR [29]

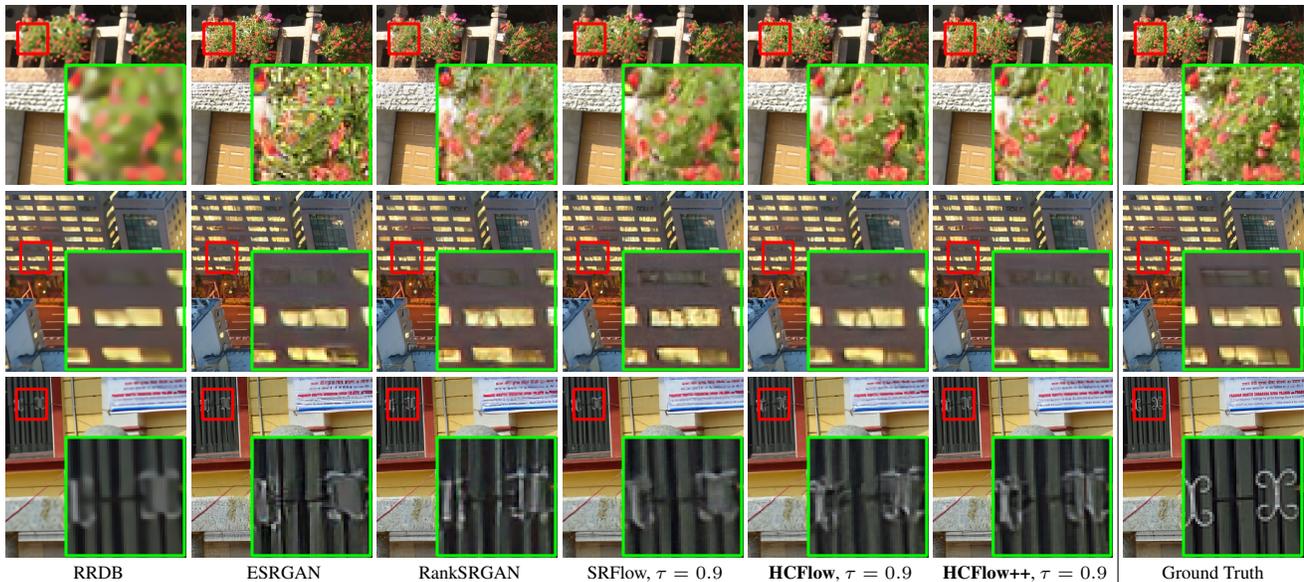


Figure 4: Visual results of general image SR ($\times 4$) on the DIV2K [1] validation set.



Figure 5: Visual results of face image SR ($\times 8$) on the CelebA [31] testing set.

and RRDB [43], perception-oriented ESRGAN [43] and RankSRGAN [53], as well as SRFlow [32]. All methods are trained on the same training dataset. From Table 2 and Fig. 4, we have several observations as follows. First, when sampling HR images with temperature $\tau = 0$, HCFlow acts like a PSNR-oriented model, achieving similar performance as EDSR and RRDB. Adding the HR pixel loss (*i.e.*, HCFlow+) can further improve the PSNR and SSIM by large margins. Second, when $\tau = 0.9$, the perceptual metrics of HCFlow are boosted dramatically. With perceptual loss and GAN loss (*i.e.*, HCFlow++), the perceptual metrics are further improved by significant margins in

terms of LPIPS and BRISQUE, which is confirmed by the visual results. Note that, unlike ESRGAN and RankSRGAN, the generated HR images of HCFlow++ are still diversified. Third, HCFlow achieves state-of-the-art performance in terms of both quantitative metrics and visual quality. It generates sharp images with few artifacts. In contrast, RRDB and SRFlow tend to produce blurry images, while ESRGAN and RankSRGAN suffer from over-sharpen artifacts and distortions. In addition, HCFlow only has about half of the number of parameters compared with SRFlow.

Face image SR. We also test HCFlow on face image SR ($\times 8$) to show its effectiveness. The compared methods

Table 4: Image rescaling ($\times 4$) results (Y-channel PSNR / SSIM) on different datasets. For IRN [47] and our method, the mean results of 5 draws are reported. Differences of PSNR / SSIM of different samples are less than 0.02.

Downscaling & Upscaling	Param	Set5 [4]	Set14 [48]	BSD100 [33]	Urban100 [14]	DIV2K [1]
Bicubic & Bicubic	-	28.42 / 0.8104	26.00 / 0.7027	25.96 / 0.6675	23.14 / 0.6577	26.66 / 0.8521
Bicubic & SRCNN [9]	57.3K	30.48 / 0.8628	27.50 / 0.7513	26.90 / 0.7101	24.52 / 0.7221	-
Bicubic & RDN [55]	22.3M	32.47 / 0.8990	28.81 / 0.7871	27.72 / 0.7419	26.61 / 0.8028	-
Bicubic & EDSR [29]	43.1M	32.62 / 0.8984	28.94 / 0.7901	27.79 / 0.7437	26.86 / 0.8080	29.38 / 0.9032
Bicubic & RCAN [54]	15.6M	32.63 / 0.9002	28.87 / 0.7889	27.77 / 0.7436	26.82 / 0.8087	30.77 / 0.8460
Bicubic & RFANet [30]	11.2M	32.67 / 0.9004	28.88 / 0.7894	27.79 / 0.7442	26.92 / 0.8112	-
Bicubic & RRDB [43]	16.3M	32.74 / 0.9012	29.00 / 0.7915	27.84 / 0.7455	27.03 / 0.8152	30.92 / 0.8486
TAD & TAU [18]	-	31.81 / -	28.63 / -	28.51 / -	26.63 / -	31.16 / -
CAR & EDSR [38]	52.8M	33.88 / 0.9174	30.31 / 0.8382	29.15 / 0.8001	29.28 / 0.8711	32.82 / 0.8837
IRN [47]	4.4M	36.19 / 0.9451	32.67 / 0.9015	31.64 / 0.8826	31.41 / 0.9157	35.07 / 0.9318
HCFlow	4.4M	36.29 / 0.9468	33.02 / 0.9065	31.74 / 0.8864	31.62 / 0.9206	35.23 / 0.9346

include PSNR-oriented RRDB, perception-oriented ESRGAN and the flow-based SRFlow. As shown in Table 3 and Fig. 5, similar observations as in general image SR can be concluded for face image SR. HCFlow achieves best quantitative and visual performance compared with competing methods. In particular, HCFlow generates sharp faces with natural details, especially on eyes, teeth and hairs. By comparison, other methods suffer from either over-smoothed results or obvious artifacts.

4.4. Experiments on Image Rescaling

As a unified framework for image SR and image rescaling, HCFlow also achieves state-of-the-art performance in image rescaling. We compare it with three kinds of rescaling methods: (1) bicubic interpolation & state-of-the-art SR models [9, 55, 29, 54, 43, 30]; (2) encoder-decoder models [18, 38]; (3) invertible neural networks [47].

As can be seen from Table 4, when the downscaling process is fixed (*i.e.*, bicubic interpolation), performances of different state-of-the-art SR models are similar and limited. When the downscaling models are optimized for the upscaling models, the results are largely improved. IRN further boosts the performance by joint optimization based on the invertible architecture. Compared with IRN, the proposed HCFlow achieves better performance on all testing datasets with an increased PSNR of 0.10 ~ 0.35dB. Besides, as shown in Fig. 6, HCFlow can better preserve image details and generates sharper edges than IRN. Since these two models have same number of parameters, HCFlow is more efficient than IRN for image rescaling, which can be attributed to the conditional modelling between high-frequency and low-frequency components.

5. Conclusion

In this paper, we proposed a unified framework, *i.e.*, hierarchical conditional flow (HCFlow), for both image super-resolution and image rescaling. It learns a fully invertible mapping between HR image and LR image as well as the latent variable. Particularly, we learn the LR image space and

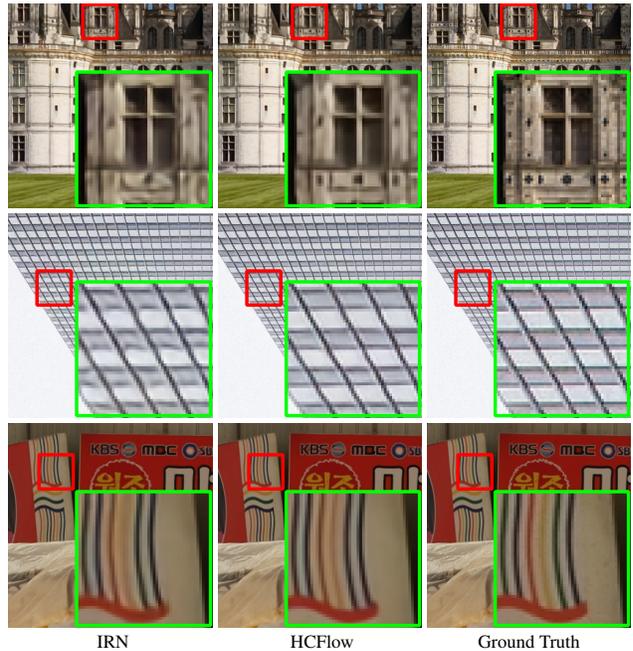


Figure 6: Visual results of image rescaling ($\times 4$) on the DIV2K [1] validation set. More results are shown in the supplementary.

design a hierarchical conditional mechanism between the latent variable (high-frequency component) and the LR image (low-frequency component). For image SR, HCFlow is trained by the negative log-likelihood loss, and is further enhanced by pixel loss, perceptual loss and GAN losses for better performance. For image rescaling, it is trained as an encoder-decoder framework, where the forward and inverse progresses are jointly optimized. Experiments demonstrate that HCFlow achieves state-of-the-art performance on general image SR, face image SR and image rescaling, in terms of both quantitative metrics and visual quality.

Acknowledgements We thank Dr. Suryansh Kumar for helpful discussion. This work was partially supported by the ETH Zurich Fund (OK), a Huawei Technologies Oy (Finland) project, the China Scholarship Council and a Microsoft Azure grant. Special thanks goes to Yijue Chen.

References

- [1] Eirikur Agustsson and Radu Timofte. Ntire 2017 challenge on single image super-resolution: Dataset and study. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 126–135, 2017. 5, 6, 7, 8
- [2] Lynton Ardizzone, Jakob Kruse, Carsten Rother, and Ullrich Köthe. Analyzing inverse problems with invertible neural networks. In *International Conference on Learning Representations*, 2018. 3
- [3] Lynton Ardizzone, Carsten Lüth, Jakob Kruse, Carsten Rother, and Ullrich Köthe. Guided image generation with conditional invertible neural networks. *arXiv preprint arXiv:1907.02392*, 2019. 4
- [4] Marco Bevilacqua, Aline Roumy, Christine Guillemot, and Marie line Alberi Morel. Low-complexity single-image super-resolution based on nonnegative neighbor embedding. In *British Machine Vision Conference*, pages 135.1–135.10, 2012. 8
- [5] Jiezhong Cao, Yawei Li, Kai Zhang, and Luc Van Gool. Video super-resolution transformer. *arXiv preprint arXiv:2106.06847*, 2021. 2
- [6] Wenlong Cheng, Mingbo Zhao, Zhiling Ye, and Shuhang Gu. Mfagan: A compression framework for memory-efficient on-device super-resolution gan. *arXiv preprint arXiv:2107.12679*, 2021. 2
- [7] Laurent Dinh, David Krueger, and Yoshua Bengio. Nice: Non-linear independent components estimation. *arXiv preprint arXiv:1410.8516*, 2014. 1, 2, 3
- [8] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real nvp. *arXiv preprint arXiv:1605.08803*, 2016. 1, 2, 3, 4
- [9] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Learning a deep convolutional network for image super-resolution. In *European Conference on Computer Vision*, pages 184–199, 2014. 2, 8
- [10] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2672–2680, 2014. 2, 5
- [11] Yong Guo, Jian Chen, Jingdong Wang, Qi Chen, Jiezhong Cao, Zeshuai Deng, Yanwu Xu, and Mingkui Tan. Closed-loop matters: Dual regression networks for single image super-resolution. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 5407–5416, 2020. 2
- [12] Jonathan Ho, Xi Chen, Aravind Srinivas, Yan Duan, and Pieter Abbeel. Flow++: Improving flow-based generative models with variational dequantization and architecture design. In *International Conference on Machine Learning*, pages 2722–2730, 2019. 1, 2, 3, 4
- [13] Chin-Wei Huang, David Krueger, Alexandre Lacoste, and Aaron Courville. Neural autoregressive flows. *arXiv preprint arXiv:1804.00779*, 2018. 3
- [14] Jia-Bin Huang, Abhishek Singh, and Narendra Ahuja. Single image super-resolution from transformed self-exemplars. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 5197–5206, 2015. 8
- [15] Takashi Isobe, Xu Jia, Shuhang Gu, Songjiang Li, Shengjin Wang, and Qi Tian. Video super-resolution with recurrent structure-detail network. In *European Conference on Computer Vision*, pages 645–660. Springer, 2020. 2
- [16] Priyank Jaini, Kira A Selby, and Yaoliang Yu. Sum-of-squares polynomial flow. *arXiv preprint arXiv:1905.02325*, 2019. 1, 2, 3
- [17] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European Conference on Computer Vision*, pages 694–711, 2016. 2, 5
- [18] Heewon Kim, Myungsub Choi, Bee Lim, and Kyoung Mu Lee. Task-aware image downscaling. In *European Conference on Computer Vision*, pages 399–414, 2018. 2, 8
- [19] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. Accurate image super-resolution using very deep convolutional networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1646–1654, 2016. 2
- [20] Sungwon Kim, Sang-gil Lee, Jongyoon Song, Jaehyeon Kim, and Sungroh Yoon. Flowavenet: A generative flow for raw audio. *arXiv preprint arXiv:1811.02155*, 2018. 1
- [21] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5
- [22] Durk P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. In *Advances in Neural Information Processing Systems*, pages 10215–10224, 2018. 1, 2, 3, 4, 5
- [23] Durk P Kingma, Tim Salimans, Rafal Jozefowicz, Xi Chen, Ilya Sutskever, and Max Welling. Improved variational inference with inverse autoregressive flow. In *Advances in Neural Information Processing Systems*, pages 4743–4751, 2016. 3
- [24] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4681–4690, 2017. 2
- [25] Yue Li, Dong Liu, Houqiang Li, Li Li, Zhu Li, and Feng Wu. Learning a convolutional neural network for image compact-resolution. *IEEE Transactions on Image Processing*, 28(3):1092–1107, 2018. 2
- [26] Jingyun Liang, Jiezhong Cao, Guolei Sun, Kai Zhang, Luc Van Gool, and Radu Timofte. SwinIR: Image restoration using swin transformer. In *IEEE Conference on International Conference on Computer Vision Workshops*, 2021. 2
- [27] Jingyun Liang, Guolei Sun, Kai Zhang, Luc Van Gool, and Radu Timofte. Mutual affine network for spatially variant kernel estimation in blind image super-resolution. In *IEEE Conference on International Conference on Computer Vision*, 2021. 2
- [28] Jingyun Liang, Kai Zhang, Shuhang Gu, Luc Van Gool, and Radu Timofte. Flow-based kernel prior with application to blind super-resolution. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 10601–10610, 2021. 1, 3
- [29] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Enhanced deep residual networks for single

- image super-resolution. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 136–144, 2017. 6, 8
- [30] Jie Liu, Wenjie Zhang, Yuting Tang, Jie Tang, and Gangshan Wu. Residual feature aggregation network for image super-resolution. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2359–2368, 2020. 2, 8
- [31] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *IEEE Conference on International Conference on Computer Vision*, pages 3730–3738, 2015. 5, 6, 7
- [32] Andreas Lugmayr, Martin Danelljan, Luc Van Gool, and Radu Timofte. Srfflow: Learning the super-resolution space with normalizing flow. In *European Conference on Computer Vision*, pages 715–732, 2020. 1, 2, 4, 5, 6, 7
- [33] David Martin, Charless Fowlkes, Doron Tal, and Jitendra Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *IEEE Conference on International Conference on Computer Vision*, pages 416–423, 2001. 8
- [34] Anish Mittal, Anush K Moorthy, and Alan C Bovik. Blind/referenceless image spatial quality evaluator. In *Asilomar Conference on Signals, Systems and Computers*, pages 723–727, 2011. 5
- [35] Anish Mittal, Rajiv Soundararajan, and Alan C Bovik. Making a “completely blind” image quality analyzer. *IEEE Signal processing letters*, 20(3):209–212, 2012. 5
- [36] Didrik Nielsen, Priyank Jaini, Emiel Hoogeboom, Ole Winther, and Max Welling. Survae flows: Surjections to bridge the gap between vaes and flows. *arXiv preprint arXiv:2007.02731*, 2020. 1, 2, 3
- [37] George Papamakarios, Theo Pavlakou, and Iain Murray. Masked autoregressive flow for density estimation. In *Advances in Neural Information Processing Systems*, pages 2338–2347, 2017. 3
- [38] Wanjie Sun and Zhenzhong Chen. Learned image downscaling for upscaling using content adaptive resampler. *IEEE Transactions on Image Processing*, 29:4027–4040, 2020. 2, 8
- [39] Radu Timofte, Eirikur Agustsson, Luc Van Gool, Ming-Hsuan Yang, and Lei Zhang. Ntire 2017 challenge on single image super-resolution: Methods and results. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 114–125, 2017. 5
- [40] Longguang Wang, Yingqian Wang, Xiaoyu Dong, Qingyu Xu, Jungang Yang, Wei An, and Yulan Guo. Unsupervised degradation representation learning for blind super-resolution. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 10581–10590, 2021. 2
- [41] Longguang Wang, Yingqian Wang, Zhengfa Liang, Zaiping Lin, Jungang Yang, Wei An, and Yulan Guo. Learning parallax attention for stereo image super-resolution. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 12250–12259, 2019. 2
- [42] Longguang Wang, Yingqian Wang, Zaiping Lin, Jungang Yang, Wei An, and Yulan Guo. Learning a single network for scale-arbitrary super-resolution. pages 10581–10590, 2021. 2
- [43] Xintao Wang, Ke Yu, Shixiang Wu, Jinjin Gu, Yihao Liu, Chao Dong, Yu Qiao, and Chen Change Loy. Esrgan: Enhanced super-resolution generative adversarial networks. In *European Conference on Computer Vision Workshops*, pages 701–710, 2018. 2, 5, 6, 7, 8
- [44] Yunxuan Wei, Shuhang Gu, Yawei Li, Radu Timofte, Longcun Jin, and Hengjie Song. Unsupervised real-world image super resolution via domain-distance aware training. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 13385–13394, 2021. 2
- [45] Christina Winkler, Daniel Worrall, Emiel Hoogeboom, and Max Welling. Learning likelihoods with conditional normalizing flows. *arXiv preprint arXiv:1912.00042*, 2019. 4
- [46] Valentin Wolf, Andreas Lugmayr, Martin Danelljan, Luc Van Gool, and Radu Timofte. Deflow: Learning complex image degradations from unpaired data with conditional flows. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 94–103, 2021. 2
- [47] Mingqing Xiao, Shuxin Zheng, Chang Liu, Yaolong Wang, Di He, Guolin Ke, Jiang Bian, Zhouchen Lin, and Tie-Yan Liu. Invertible image rescaling. In *European Conference on Computer Vision*, pages 126–144, 2020. 1, 2, 5, 6, 8
- [48] Roman Zeyde, Michael Elad, and Matan Protter. On single image scale-up using sparse-representations. In *International Conference on Curves and Surfaces*, pages 711–730, 2010. 8
- [49] Kai Zhang, Luc Van Gool, and Radu Timofte. Deep unfolding network for image super-resolution. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3217–3226, 2020. 2
- [50] Kai Zhang, Yawei Li, Wangmeng Zuo, Lei Zhang, Luc Van Gool, and Radu Timofte. Plug-and-play image restoration with deep denoiser prior. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021. 2
- [51] Kai Zhang, Jingyun Liang, Luc Van Gool, and Radu Timofte. Designing a practical degradation model for deep blind image super-resolution. In *IEEE Conference on International Conference on Computer Vision*, 2021. 2
- [52] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 586–595, 2018. 5
- [53] Wenlong Zhang, Yihao Liu, Chao Dong, and Yu Qiao. Ranksrgan: Generative adversarial networks with ranker for image super-resolution. In *IEEE Conference on International Conference on Computer Vision*, pages 3096–3105, 2019. 2, 6, 7
- [54] Yulun Zhang, Kunpeng Li, Kai Li, Lichen Wang, Bineng Zhong, and Yun Fu. Image super-resolution using very deep residual channel attention networks. In *European Conference on Computer Vision*, pages 286–301, 2018. 2, 8
- [55] Yulun Zhang, Yapeng Tian, Yu Kong, Bineng Zhong, and Yun Fu. Residual dense network for image super-resolution. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2472–2481, 2018. 2, 8