Boosting Flow-based Generative Super-Resolution Models via Learned Prior

Li-Yuan Tsao[1] Yi-Chen Lo[2] Chia-Che Chang[2] Hao-Wei Chen[1] Roy Tseng[2] Chien Feng[1] Chun-Yi Lee[1]
[1]National Tsing Hua University [2]MediaTek Inc.
{lytsao, jaroslaw1007, kmes9961002}@gapp.nthu.edu.tw
{yichen.lo, chia-che.chang, roy.tseng}@mediatek.com cylee@cs.nthu.edu.tw

Abstract

Flow-based super-resolution (SR) models have demonstrated astonishing capabilities in generating high-quality images. However, these methods encounter several challenges during image generation, such as grid artifacts, exploding inverses, and suboptimal results due to a fixed sampling temperature. To overcome these issues, this work introduces a conditional learned prior to the inference phase of a flow-based SR model. This prior is a latent code predicted by our proposed latent module conditioned on the low-resolution image, which is then transformed by the flow model into an SR image. Our framework is designed to seamlessly integrate with any contemporary flow-based SR model without modifying its architecture or pre-trained weights. We evaluate the effectiveness of our proposed framework through extensive experiments and ablation analyses. The proposed framework successfully addresses all the inherent issues in flow-based SR models and enhances their performance in various SR scenarios. Our code is available at: https://github.com/liyuantsao/FlowSR-LP

1. Introduction

Image super-resolution (SR) aims to reconstruct a high-resolution (HR) image given its low-resolution (LR) counterpart. Typically, the effectiveness of SR methods is evaluated based on fidelity and perceptual quality of the generated SR images, with commonly used metrics such as Peak Signal-to-Noise Ratio (PSNR), Structural Similarity (SSIM) for the former, and Learned Perceptual Image Patch Similarity (LPIPS) [78] for the latter. However, optimizing both metrics simultaneously is fundamentally challenging due to the inherent perception-distortion trade-off [6] in SR tasks. As a result, existing SR methods are broadly classified into two categories: *fidelity-oriented SR*, which prioritizes pixel-wise reconstruction accuracy, and *generative SR*, which focuses on enhanced visual quality.

The recent emergence of Flow-based SR [34, 45, 50, 51, 67, 76] bridges this divide, as flow models possess the capability to control the diversity of image content during inference time by adjusting the sampling temperature (*i.e.*, the standard deviation of a Gaussian distribution). As a result, a single flow-based SR model (or simply "*flow model*" hereafter) can produce images that either prioritize high fidelity or exhibit improved perceptual quality. This unique feature provides flow-based models with the potential to excel in each category of SR methods, making them a promising framework for SR tasks.

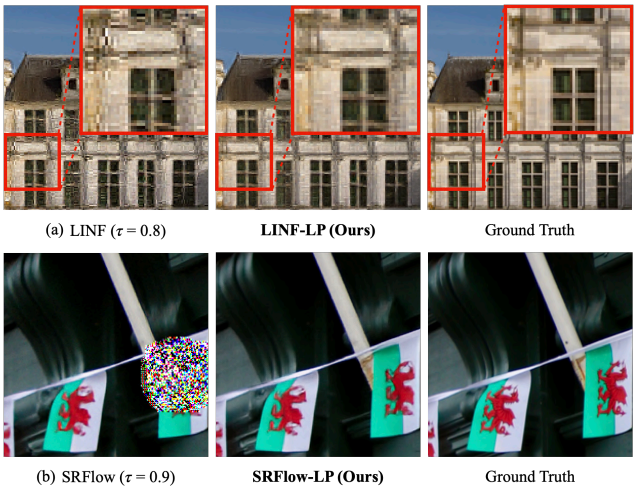


Figure 1. Our framework enhances the capability of contemporary flow-based SR models by: (1) mitigating grid artifacts, (2) preventing exploding inverses, and (3) eliminating the use of a fixed sampling temperature to unlock the full potential of flow models.

Despite their flexibility, flow-based SR methods encounter several challenges in the image generation process. These include (1) grid artifacts in the generated images, (2) the exploding inverse issue, and (3) suboptimal results stemming from the use of a fixed sampling temperature. "Grid artifacts" stands for the discontinuities in textures within an image [47]. As depicted in the left image of Fig. 1 (a), distinct borders between the generated image patches could occasionally lead to a checkerboard pattern [53]. Another critical aspect is the "exploding inverse" [4, 26] in invertible neural networks, which refers to the occurrence of infi-

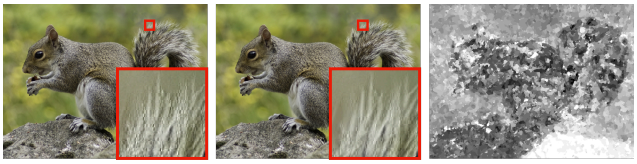


Figure 2. The values in the "best temperature map" represent the sampling temperature for each position that yields the optimal LPIPS score, where a lighter color represents a higher temperature. The best temperatures are determined through an exhaustive search, with the LPIPS scores calculated on the whole image.

nite values in the inverse process. This phenomenon leads to the appearance of noisy patches within images. As illustrated in Fig. 1 (b), images affected by exploding inverse display some noisy patches, which obscure the original content within the image. Moreover, despite using a fixed sampling temperature during evaluation is a common practice in flow-based SR methods, this approach might not always be suitable, as the ideal temperature settings could vary across different regions. As depicted in Fig. 2, the areas of low-frequency components such as backgrounds favor higher temperatures for diversity enhancement. In contrast, high-frequency areas (*e.g.*, the detailed fur of a squirrel) require lower temperatures to maintain consistent contents. While the optimal temperature for each area can be identified through an exhaustive search to boost performance, this approach may be highly cost-ineffective and impractical in real-world scenarios. As a result, the efficacy of flow models could be constrained by the suboptimal temperature setting. Based on these challenges, flow-based SR methods still hold the potential for further enhancement.

In light of the aforementioned issues, a key element to addressing these problems of flow models could be a learned prior. This learned prior should hold the following properties. First, it captures the correlation between image patches to alleviate the discontinuities. Second, this prior should be prevented from being an out-of-distribution input to the flow model, which leads to an exploding inverse [4, 26]. Third, to eliminate the need to fine-tune a sampling temperature, this prior should be directly generated by a model instead of sampling from a Gaussian prior. Fig. 3 illustrates the integration of a learned prior into SR tasks. In this approach, the flow model receives the learned prior as input, replacing the conventional method which employs a randomly sampled latent code.

To achieve this, this study proposes a framework that introduces a latent module as the conditional learned prior. The latent module is designed to directly estimate a learned latent code in a single-pass for flow model inference. This design philosophy aligns well with the principle that real-world SR applications prefer fast inference and consistent predictions [2, 9]. Specifically, the proposed framework consists of two main components: a latent module and a

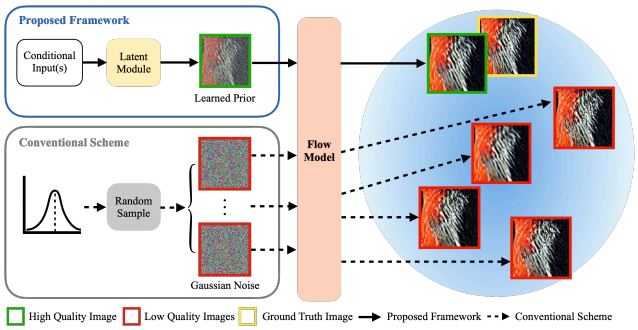


Figure 3. The proposed framework leverages a conditional learned prior to the inference phase of a flow-based SR model, aiming to approximate the ideal latent code corresponding to the HR image.

flow model, with the latter being any contemporary flow-based SR model. The latent module is responsible for predicting a latent code conditioned on the LR image, which is then transformed by the flow model into an SR image.

In this work, the proposed latent module is integrated with two flow-based SR models, including the arbitrary-scale SR framework LINF [76] and the fixed-scale framework SRFlow [50], to demonstrate its effectiveness, generalizability, and flexibility with extensive experiments. The contribution of this study can be summarized as follows:

- This study reveals three inherent deficiencies in flow-based SR methods, which are key issues that require enhancements to fully unleash the potential of flow models.
- We introduce a conditional learned prior to the inference phase of a flow-based SR model, which effectively addresses the inherent issues of flow models without modifying their architecture or pre-trained weights.
- The proposed latent module is highly flexible in terms of network architecture design, which allows the adoption of any commonly used backbone. Even a lightweight module could introduce noticeable improvements.
- Our proposed framework generalizes to both fixed-scale and arbitrary-scale flow-based SR frameworks without requiring customized components, which also leads to advancements at out-of-training-distribution scales.

2. Related Work

2.1. Image Super-Resolution

Contemporary deep learning-based SR methods [17, 24, 33, 39, 62, 80] primarily fall into two categories: fidelity-oriented SR and generative SR. Fidelity-oriented SR methods [14, 42, 44, 48, 70, 81, 82] aim to promote pixel-wise reconstruction accuracy and are commonly trained with L1 or L2 loss. Despite achieving high fidelity, these methods often produce blurry results. To overcome this and synthesize realistic HR images, various generative SR techniques have been proposed, including GAN-based [40, 58, 59, 71,

基于深度学习的当代超分辨率方法：fidelityoriented SR and generative SR

72, 79], AR-based [23], flow-based [34, 45, 50, 51, 67], and diffusion-based [43, 60] methods. Although these works effectively generate diverse and visually plausible HR images, they are limited to super-resolving images at a predefined upsampling scale, restricting their real-world applicability. Recently, various arbitrary-scale SR methods [10, 12, 15, 27, 41, 74–76] have emerged. These approaches are capable of producing high-fidelity images even at significantly large upsampling scales (*e.g.*, 30×). Among these methods, LINF [76] pioneers the field of arbitrary-scale generative SR, which is able to generate realistic images across continuous scales.

介绍基于流的SR方法

2.2. Flow-based Super-Resolution

Flow-based SR methods utilize normalizing flows to establish a bijective mapping between the conditional distribution of HR images and a prior distribution. This approach addresses the ill-posed nature of SR tasks by modeling the entire HR image space. Besides excelling in general SR tasks [34, 45, 50, 51, 67, 76], flow-based SR methods find applications in a variety of specialized SR tasks. These include blind image SR [46], remote sensing image [73], Magnetic Resonance Imaging [37], Magnetic Resonance Spectroscopic Imaging [18], and scientific data SR [61].

2.3. Learned Prior

Modern deep generative models [7] typically learn mappings between data and latent variables (i.e., latent codes) based on Gaussian prior, either explicitly such as variational autoencoders (VAEs) [35], normalizing flows [38], diffusion models [25, 63–66] or implicitly such as generative adversarial networks (GANs) [21, 32]. Regardless of their distinct learning formulations, they generally adopt a fixed Gaussian prior for sampling during inference. Recently, improved VAEs [11, 13, 16, 19, 20, 22, 36, 54–56, 69] have highlighted more expressive priors for better lower bound and sample generation. These advancements suggest that learning priors using neural networks can be used for sampling since the decoder network was effectively trained on latent codes from the learned posterior. Likewise, we propose to learn a latent module as a form of learned prior over latent codes inverted from training data by conditional normalizing flow, to remedy the train-test gap as a general framework to boost flow-based super-resolution quality.

3. Preliminaries

3.1. Fixed-scale Flow-based SR

Given an LR image $x \in \mathbb{R}^{H\times W\times 3}$ and an HR image $y \in \mathbb{R}^{sH\times sW\times 3}$ with a predefined scaling factor s, fixed-scale flow-based SR models [34, 45, 50, 51] aim to capture the entire conditional distribution $p_{Y|X}(y|x)$. Specifically, they learn a bijective mapping between a conditional distribution $p_{Y|X}(y|x)$ and a prior distribution $p_Z(z)$, where $z \in \mathbb{R}^{sH\times sW\times 3}$ is typically a standard normal distribution $\mathcal{N}(0, I)$. By utilizing an invertible neural network with k invertible layers, such transformation is given by:

$$\begin{aligned} z &= f_\theta(y; x) = f_k \circ \cdots \circ f_1(y; x), \\ y &= f_\theta^{-1}(z; x) = f_1^{-1} \circ \cdots \circ f_k^{-1}(z; x). \end{aligned} \tag{1}$$

Additionally, according to the change of variable theorem, the probability of an HR-LR image pair (y, x) is defined as:

$$p_{Y|X}(y|x, \theta) = p_Z(f_\theta(y; x)) \cdot \left| \det \frac{\partial f_\theta(y; x)}{\partial y} \right|. \tag{2}$$

During training, fixed-scale flow-based SR models can be optimized through negative log likelihood (NLL) loss with a large set of HR-LR training pairs $\{(y_i, x_i)\}_{i=1}^N$.

3.2. Arbitrary-scale Flow-based SR

Despite successfully tackling the ill-posed nature of SR task by modeling the HR image space, fixed-scale flow-based SR models are only able to super-resolve images with a predefined scaling factor (*e.g.*, 4×), limiting their practicality. To address this issue, the arbitrary-scale flow-based SR framework "*Local Implicit Normalizing Flow*" (LINF) [76] shifts the learning target from the entire HR image to local patches. During inference, it generates local patches at corresponding coordinates independently, and then combines these patches to form the final image.

Specifically, given an LR image $x \in \mathbb{R}^{H\times W\times 3}$, the coordinate of a local patch $c_{i,j} \in \mathbb{R}^2$, a scaling factor s, and the corresponding HR patch $y_{i,j} \in \mathbb{R}^{n\times n\times 3}$, where n is typically set to 3, and i, j is the index of an image patch. The goal of LINF [76] is to learn a bijective mapping between a conditional distribution $p_{Y|X}(y_{i,j}|x, c_{i,j}, s)$ and a latent distribution $p_Z(z) = \mathcal{N}(0, I)$, where $z \in \mathbb{R}^{n\times n\times 3}$. Similar to fixed-scale framework, the probability is given by:

$$\begin{aligned} &p_{Y|X}(y_{i,j}|x, c_{i,j}, s, \theta) = \\ &\quad p_Z(f_\theta(y_{i,j}; x, c_{i,j}, s)) \cdot \left| \det \frac{\partial f_\theta(y_{i,j}; x, c_{i,j}, s)}{\partial y_{i,j}} \right|. \end{aligned} \tag{3}$$

In practice, the HR patch $y_{i,j}$ is replaced by the residual map $m_{i,j} = y_{i,j} - x_{i,j}^{\text{up}}$, where $x_{i,j}^{\text{up}}$ represents the bilinear-upsampled LR image. During training, LINF [76] utilizes the Negative Log-Likelihood (NLL) loss along with pixel-wise L1 loss and VGG perceptual loss [30] for additional fine-tuning across various metrics.

4. Methodology

In this section, we begin by defining the formulation of the proposed method, followed by a detailed description of the proposed framework and an elaboration on the design of the objective function.

4.1. Problem Formulation

Given an LR image $x \in \mathbb{R}^{H\times W\times 3}$ and a flow model f_θ, the objective of this work is to derive a latent code

固定尺度和任意尺度基于流的SR模型的基本原理和数学公式

$z^* \in \mathbb{R}^{sH \times sW \times 3}$, given by: $y = f_\theta^{-1}(z^*; x)$, where $y \in \mathbb{R}^{sH \times sW \times 3}$ represents the ground truth HR image, which is not available during inference, and s represents a scaling factor. Successfully identifying z^* enables the precise reconstruction of the corresponding HR image y. This process is facilitated by the invertible nature of normalizing flow, which guarantees the existence of z^*. To realize this objective, we utilize a latent module G designed to generate a latent code $\hat{z}$ in a single-pass during inference, which aims to approximate z^* as closely as possible, expressed as:

$$z^* \approx \hat{z} = G(\cdot). \tag{4}$$

Similarly, for the arbitrary-scale SR framework, we aim to identify latent codes $\hat{z}_{i,j}$ such that $\hat{z}_{i,j} = z^*_{i,j}$ for all i, j within an image, formulated as:

$$z^*_{i,j} \approx \hat{z}_{i,j} = G(\cdot), \forall i, j. \tag{5}$$

After deriving $\hat{z}$, the flow model f_θ transforms $\hat{z}$ into an SR image $\hat{y}$, given by: $\hat{y} = f_\theta^{-1}(\hat{z}; \cdot)$. Note that only the latent module undergoes training, the pre-trained flow model f_θ remains frozen during both training and inference phases.

4.2. Framework Overview

Fig. 4 illustrates an overview of our proposed framework, which aims to leverage a conditional learned prior to address the inherent issues in flow-based SR models. Specifically, it consists of a proposed latent module and a flow model, with the latter being any contemporary flow-based SR model. The latent module processes signals from the LR image to produce a learned prior, which is then transformed by the flow model into the SR images. In this work, we integrate our framework with two existing flow models, including LINF [76], which represents arbitrary-scale SR framework, and SRFlow [50] for fixed-scale SR method.

4.3. Latent Module

Overview. The latent module is designed to predict the conditional learned prior by utilizing the LR conditional signals, where this prior is a latent code that holds several properties capable of addressing the inherent issues of flow models, as mentioned in Section 1. To achieve this, the latent module is designed to: (1) fuse information across patches to mitigate discontinuities in image content (*i.e.*, grid artifacts), (2) leverage LR conditional signals to avoid out-of-distribution predictions that lead to an exploding inverse, and (3) directly output a learned prior without random sampling. As a result, this work introduces a latent module that extracts features of an LR image from both image space and latent space, and processes these features with a deep neural network to derive the conditional learned prior. Specifically, the architecture of the latent module comprises two feature encoders and a latent generator. One encoder processes the LR image to extract image space features, while the other works on the "initial prior" to capture latent space signals, where the initial prior is the latent code corresponding to the upsampled LR image. Then, the latent generator utilizes these features to produce the learned latent code. The introduction of an initial prior provides a promising initialization in the latent space for seeking z^*, thus easing the prior generation process. A detailed analysis of this effect is presented in Section 5.3.

The Design of the Latent Module. Regarding the design of the latent module, the feature encoders adopt a five-layer dense block [28] architecture but maintain independent weights to process inputs from different spaces. On the other hand, the architecture of the latent generator is highly flexible, which allows the incorporation of any commonly used module. In this work, UNet [57] is primarily selected as the latent generator due to its efficiency and capabilities in capturing multi-level features, which aligns well with our objective. In addition, Section 5.3 provides a detailed analysis of alternative architectures of the latent generator, including EDSR-baseline [48] and Swin Transformer [49].

4.4. Objective Function

To generate a latent code $\hat{z}$ that approximates the optimal latent code z^*, our framework employs an objective function that aims at boosting both the accuracy in latent space and the perceptual quality of generated images. For the learning in latent space, we define a loss function $\mathcal{L}_{latent}$ to minimize the L1 distance between $\hat{z}$ and z^*, formulated as:

$$\mathcal{L}_{latent} = \frac{1}{N} \sum_{i}^{N} ||z^*_i - \hat{z}_i||_1, \tag{6}$$

where z^* is obtained by transforming the HR image into the corresponding latent code with a flow model, which is available during training. In addition, we adopt the VGG perceptual loss [30] $\mathcal{L}_{percep}$ in the image space to produce enhanced SR results. This objective guides our latent module to generate a latent code, which corresponds to an image that visually resembles the HR image. Specifically, it calculates the L1 distance between features extracted by the SR image and the HR image, with a pre-trained VGG19 [31] network Ψ_{per}, which can be expressed as:

$$\mathcal{L}_{percep} = \frac{1}{N} \sum_{i}^{N} ||\Psi_{per}(y_i) - \Psi_{per}(\hat{y}_i)||_1. \tag{7}$$

As a result, the overall objective can be expressed as:

$$\mathcal{L}_{total} = \mathcal{L}_{percep} + \lambda \mathcal{L}_{latent}. \tag{8}$$

In practice, λ is set to 0 when integrating our framework with LINF, and 0.1 when combined with SRFlow based on the following observations. First, we find the sole application of Eq. (6) results in an average prediction across all potential latent codes. This outcome resembles the "regression-to-the-mean" [8] effect observed with L1 regression loss in image space. Furthermore, while adopting

Latent Module: 在推理阶段为流模型生成一个条件学习先验，这个先验是一个在代码，代替传统高斯随机采样

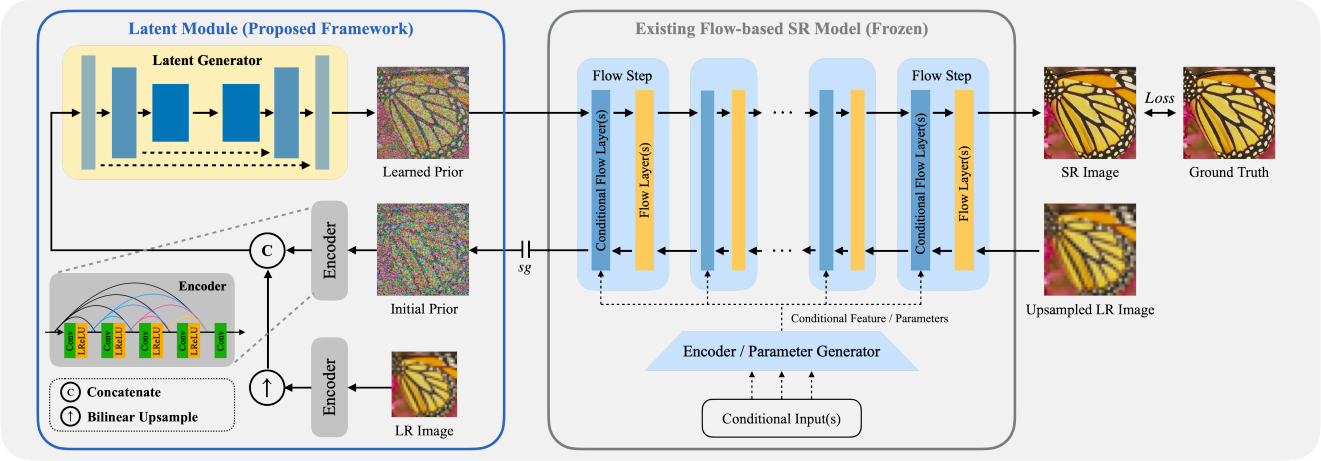


Figure 4. An overview of the proposed inference scheme for flow-based SR models. This framework consists of the proposed latent module and a flow model, which can be any existing flow-based SR model. The latent module generates a learned latent by leveraging the information extracted from an LR image. Then, the flow model transforms this latent into the corresponding SR image.

the perceptual loss is sufficient when integrating our framework with LINF model, SRFlow experiences training instability under this scheme, where infinite values emerge occasionally. To address this phenomenon, we employ the latent space loss $\mathcal{L}_{latent}$ as a regularization term, which effectively prevents the generation of out-of-distribution latent codes [26] that lead to an exploding inverse. Section 5.3 delivers an analysis of the impact of objective functions.

5. Experiments

In this section, we present the experimental results and ablation studies. The results demonstrate how the proposed framework addressing the inherent issues of flow-based SR methods, and the effects after integrating our proposed framework with LINF [76] and SRFlow [50].

5.1. Experimental Setup

模型：

Arbitrary-scale Flow-based SR. When integrating our framework with the arbitrary-scale framework LINF [76], we utilize two variants of their 3×3 patch-based models: RRDB-LINF and EDSR-baseline-LINF. The former employs an RRDB [72] as the image encoder, whereas the latter uses an EDSR-baseline [48] backbone. For RRDB-LINF, we adopt their released model. In the case of EDSR-baseline-LINF, we reproduced the baseline model using their codebase due to the absence of a pre-trained model. Upon incorporating our proposed "*Learned Prior*" (LP) into LINF, we refer to the enhanced versions as "*LINF-LP*", with two configurations: "*EDSR-baseline-LINF-LP*" and "*RRDB-LINF-LP*". To train LINF-LP, we set the LR image size to 96×96, and crop the corresponding $96s \times 96s$ HR image patch as ground truth, where s denotes a continuous upsampling scale $s \in \mathcal{U}(1, 4)$. LINF-LP is trained over 1,000 epochs with a batch size of 16. The initial learning rate is $1e^{-4}$, which is halved at [200, 400, 600, 800] epochs when using UNet and EDSR-baseline as the latent generator, and at [500, 850, 900, 950] epochs for Swin-T.

Fixed-scale Flow-based SR. For the fixed-scale SR method SRFlow [50], we also implement the enhanced version "*SRFlow-LP*". The experiments of SRFLow-LP are conducted on the DIV2K 4x SR task, and we integrate our proposed framework with their released pre-trained model. We train SRFlow-LP over five epochs, using a batch size of 12. The HR image size is set to 160×160, and the initial learning rate is $1e^{-4}$, which is halved after each epoch.

Datasets. We use the DIV2K [1] dataset for training EDSR-baseline-LINF-LP, and the joint of DIV2K and Flickr2K [68] to train RRDB-LINF-LP and SRFlow-LP. These two datasets consist of 800 and 2,650 images, respectively. The evaluation is conducted on the DIV2K validation set. For the arbitrary-scale framework LINF-LP, we further test on several SR benchmark datasets, including Set5 [5], Set14 [77], B100 [52], and Urban100 [29].

Network Details. For our proposed latent module, we utilize a three-layer deep UNet [57] as the latent generator. It starts with an initial feature dimension of 64, comprising both initial prior and LR image features, each having a dimension of 32. The feature dimension is doubled with each downsample operation and halved with each upsample operation. For the ablation study in Section 5.3, the feature dimensions of EDSR-baseline [48] and Swin Transformer (Swin-T) [49] are 64 and 192, respectively.

Evaluation Metrics. For evaluation, we select several commonly used metrics in SR tasks. These include PSNR and SSIM for fidelity measurements, and LPIPS [78] for assessing perceptual quality. In addition, we employ the LR-PSNR metric [3], which calculates the PSNR between the bicubic downsampled SR image and the original LR image.

Table 1. The 4× SR results on the DIV2K [1] validation set. The best results are highlighted in red.

Generative SR Method	PSNR(↑)	SSIM(↑)	LPIPS(↓)	LR-PSNR(↑)
ESRGAN [72]	26.22	0.75	0.124	39.03
RankSRGAN [79]	26.55	0.75	0.128	42.33
SRDiff [43]	27.41	**0.79**	0.136	**55.21**
SROOE [59]	**27.69**	**0.79**	**0.096**	50.80
Flow-based Generative SR Method	**PSNR(↑)**	**SSIM(↑)**	**LPIPS(↓)**	**LR-PSNR(↑)**
HCFlow++ ($\tau = 0.9$) [45]	26.61	0.74	0.110	50.07
SRFlow ($\tau = 0.9$) [50]	27.09	0.76	0.121	49.96
SRFlow-LP (Ours)	**27.51**	**0.78**	**0.109**	**51.51**
EDSR-baseline-LINF ($\tau = 0.8$) [76]	27.02	0.76	0.130	43.19
EDSR-baseline-LINF-LP (Ours)	**27.64**	**0.78**	**0.119**	**46.96**
RRDB-LINF ($\tau = 0.8$) [76]	27.33	0.77	0.112	43.64
RRDB-LINF-LP (Ours)	**28.00**	**0.78**	**0.105**	**47.30**

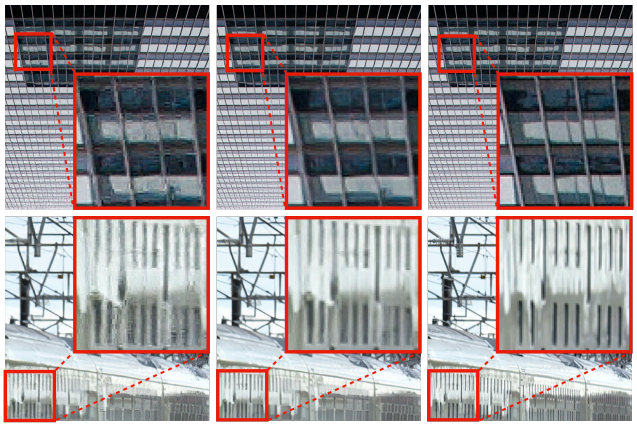
5.2. Experimental Results

5.2.1 Generative Super-Resolution

We compare the performance of LINF-LP and SRFlow-LP with Flow-based [45, 50, 76] and other generative SR methods [43, 59, 72, 79]. The results in Table 1 reveal the benefits of our framework in two aspects: effectiveness and generalizability. Firstly, both SRFlow-LP and LINF-LP demonstrate significant improvements in fidelity- and perception-oriented metrics. For instance, SRFlow-LP gains an improvement of 0.42 dB in PSNR and 0.012 in LPIPS, and LINF-LP achieves an improvement of 0.62 dB and 0.67 dB in PSNR, along with gains of 0.01 and 0.007 in LPIPS when using EDSR-baseline and RRDB backbones, respectively. Table 1 also demonstrates that our proposed framework enables flow-based methods to achieve comparability with state-of-the-art methods [43, 45, 59, 72, 79]. In addition, the simultaneous enhancements in both metrics present the capability of our framework to further push the boundary of the perception-distortion trade-off [6]. Secondly, the improvements observed in both SRFlow-LP and LINF-LP exhibit the generalizability of our approach, which is capable of extending to both fixed-scale and arbitrary-scale frameworks without the need for customized components. This finding validates the capability of the proposed framework across various SR scenarios.

5.2.2 Arbitrary-scale Flow-based SR

Quantitative Results. For the arbitrary-scale SR framework LINF [76], we compare the performance of EDSR-baseline-LINF-LP and RRDB-LINF-LP with their corresponding baselines. We evaluate the performance at both in-distribution (2×, 3×, 4×) and out-of-training-distribution (6×, 8×) upsampling scales on widely used SR benchmark datasets [5, 29, 52, 77]. Since our focus is flow-based generative models, we adopt LPIPS [78] for assessing the perceptual quality of generated images. In addition, LINF adopts a specific sampling temperature τ for different scaling factors: $\tau = 0.5$ for 2×, 3×, 4× SR, $\tau = 0.4$ for 6× SR, and $\tau = 0.2$ for 8× SR. In contrast, our framework directly predicts a latent code from the LR image without the need to fine-tune this hyperparameter.



LINF ($\tau = 0.8$) **LINF-LP (Ours)** Ground Truth

Figure 5. A qualitative comparison between the 4× SR results of LINF [76] and LINF-LP. (zoom in for better clarity)

Table 2 demonstrates the adaptability and the potential of our framework. (1) Both EDSR-baseline-LINF-LP and RRDB-LINF-LP achieve considerable improvements across in-distribution and out-of-training-distribution scales. This demonstrates the ability of our framework to adaptively predict a conditional learned prior for arbitrary upsampling scales, even at OOD scales. (2) RRDB-LINF-LP typically shows greater enhancement than EDSR-baseline-LINF-LP. For instance, RRDB-LINF-LP gains an improvement of 0.022 in LPIPS on Urban100 8× SR, compared to a 0.009 enhancement by EDSR-baseline-LINF-LP. We attribute this to the stronger capability of the RRDB [72] backbone, which provides superior learning signals for the latent module. In light of this, we suggest the enhancements achieved by our framework are proportional to the capacity of the flow models, with more powerful models potentially yielding more pronounced improvements.

Mitigating Grid Artifacts. As illustrated in Fig. 5, grid artifacts are prominent in images produced by LINF [76]. This issue arises since LINF constructs an image by combining independently sampled patches. When utilizing a higher temperature τ to generate images with diverse content, the magnitude of values sampled by adjacent patches could vary greatly, which leads to discontinuities in image content. However, LINF-LP effectively reduces the presence of grid artifacts. The key to this improvement lies in the learned prior predicted by our latent module, which efficiently captures global information from an LR image and can guide LINF to generate coherent content.

5.2.3 Fixed-scale Flow-based SR

The Occurrence of Exploding Inverses. This analysis examines the likelihood of SRFlow [50] and SRFlow-LP encountering exploding inverses and measures the quality of generated images with LPIPS and LR-PSNR. Table 3

Table 2. The arbitrary-scale SR results on SR benchmark datasets. "*In-scales*" and "*OOD-scales*" refer to in- and out-of-training-distribution scales. LPIPS [78] scores are reported (lower is better), with the best and second-best highlighted in **red** and blue, respectively.

Method	Set5 [5]					Set14 [77]					B100 [52]					Urban100 [29]				
	In-scales			OOD-scales		In-scales			OOD-scales		In-scales			OOD-scales		In-scales			OOD-scales	
	×2	×3	×4	×6	×8	×2	×3	×4	×6	×8	×2	×3	×4	×6	×8	×2	×3	×4	×6	×8
EDSR-baseline-MetaSR [27]	0.057	0.125	0.175	0.253	0.326	0.094	0.207	0.286	0.395	0.460	0.147	0.285	0.376	0.492	0.565	0.065	0.157	0.233	0.352	0.446
EDSR-baseline-LIIF [15]	0.056	0.124	0.173	0.248	0.307	0.093	0.205	0.284	0.390	0.449	0.147	0.282	0.372	0.486	0.556	0.064	0.155	0.228	0.338	0.422
EDSR-baseline-LTE [41]	0.056	0.123	0.174	0.257	0.326	0.092	0.203	0.283	0.396	0.463	0.146	0.280	0.371	0.495	0.570	0.063	0.152	0.224	0.345	0.436
EDSR-baseline-LINF [76] ($\tau = \tau_0$)	0.035	0.067	0.088	0.158	0.249	0.064	0.115	0.163	0.275	0.375	0.108	0.172	0.207	0.319	0.451	0.050	0.110	0.158	0.273	0.386
EDSR-baseline-LINF-LP (Ours)	**0.026**	**0.047**	**0.074**	**0.145**	**0.243**	**0.054**	**0.094**	**0.144**	**0.253**	**0.364**	**0.084**	**0.127**	**0.177**	**0.289**	**0.425**	**0.044**	**0.098**	**0.146**	**0.253**	**0.377**
RRDB-LINF [76] ($\tau = \tau_0$)	0.034	0.064	0.084	0.147	0.247	0.059	0.110	0.146	0.252	0.359	0.097	0.152	0.194	0.306	0.444	0.040	0.093	0.137	0.239	0.354
RRDB-LINF-LP (Ours)	**0.023**	**0.042**	**0.066**	**0.131**	**0.234**	**0.043**	**0.087**	**0.124**	**0.221**	**0.322**	**0.061**	**0.113**	**0.163**	**0.264**	**0.378**	**0.033**	**0.081**	**0.126**	**0.219**	**0.331**

Table 3. The 4× SR results on the DIV2K [1] validation set, which are the average of 10 validation runs. "*%Inf*" [4, 26] represents the probability of generating an exploding inverse. The best and second best results are highlighted in **red** and blue, respectively.

Method	%Inf (↓)	LPIPS(↓)	LR-PSNR(↑)
SRFlow ($\tau = 0.8$) [50]	**0**	0.124	50.35
SRFlow ($\tau = 0.9$) [50]	0.8	0.120	49.93
SRFlow ($\tau = 1.0$) [50]	6.7	0.130	48.62
SRFlow-LP (Ours)	**0**	**0.109**	**51.51**

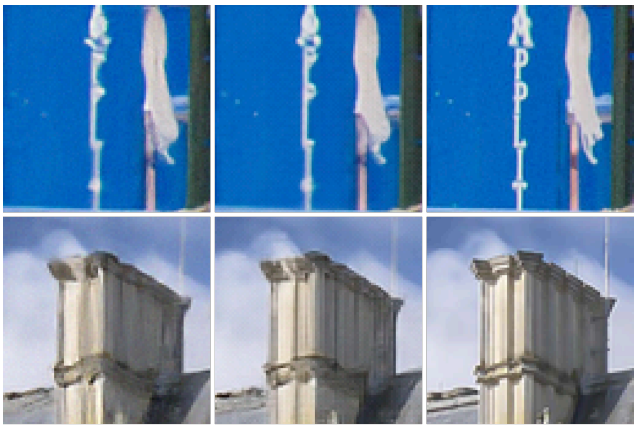
demonstrates that SRFlow-LP effectively prevents the occurrence of exploding inverses. This enhancement could be attributed to the conditional learned prior predicted by our framework, which avoids out-of-distribution predictions that lead to subsequent exploding inverses [26]. A detailed analysis of this effect is presented in Section 5.3.

For SRFlow, the frequency of producing exploding inverses rises with increasing temperature. To elaborate, SRFlow achieves optimal LPIPS scores at $\tau = 0.9$, with a slight risk of encountering exploding inverses, which strikes a balance between perceptual quality and consistency. In addition, SRFlow shows distinct effects at $\tau = 0.8$ and $\tau = 1.0$. At a temperature $\tau = 0.8$, SRFlow delivers a higher LR-PSNR score and shows no exploding inverses, while the perceptual quality is compromised. At $\tau = 1.0$, the probability of encountering exploding inverses rises sharply. Also, excessively high temperatures could introduce image artifacts [76], thus affecting both LPIPS and LR-PSNR scores.

Qualitative Results. Fig. 6 presents a qualitative comparison between SRFlow-LP and SRFlow. The former can generate finer details such as lines and circles, while the latter struggles to render these even at a high-temperature setting of $\tau = 0.9$ for more diverse contents. This observation indicates the capabilities of our framework, which not only mitigates grid artifacts by integrating global information, as described in Section 5.2.2 but also excels in capturing intricate details, resulting in visually appealing effects.

5.3. Ablation Studies

This section presents ablation studies and in-depth discussions on the design of our latent generator, along with the



SRFlow ($\tau = 0.9$) **SRFlow-LP (Ours)** Ground Truth

Figure 6. A qualitative comparison between the 4× SR results of SRFlow [50] and SRFlow-LP.

effects of input selection and the objective function. In the second and third analyses, we adopt EDSR-baseline-LINF and EDSR-baseline-LINF-LP for comparison. For clarity, we denote them as "LINF" and "LINF-LP", respectively.

Design Flexibility of the Latent Generator. As described in Section 4.3, the architecture of our latent generator allows the use of commonly used backbones. In this analysis, we explore this flexibility by adopting EDSR-baseline [48] and Swin-T [49] as alternative latent generators. The results presented in Table 4 illustrate the effectiveness and efficiency of our proposed framework. Firstly, LINF-LP yields considerable improvements with all these backbones in both fidelity- and perceptual-oriented metrics compared to LINF. Moreover, even the lightweight backbone EDSR-baseline, with a model size of only 1.4M, significantly boosts the performance. Among these results, LINF-LP (UNet) delivers the best PSNR and LPIPS scores. This superior performance could be attributed to the multi-level architecture of UNet [57], which effectively incorporates both local and global features into the learned prior.

The Influence of Different Inputs. We conduct analyses on the input to the proposed latent module by experimenting with various combinations, including (1) using the LR image only, (2) adopting the initial prior only, and (3) com-

Table 4. The 4× SR results on the DIV2K [1] validation set. The names in the parentheses (*e.g.*, UNet) refer to the architecture of our latent generator. The best and second-best results are highlighted in **red** and blue, respectively.

Method	PSNR (↑)	SSIM (↑)	LPIPS (↓)	Parameters (M)
EDSR-baseline-LINF (t=0.8) [76]	27.02	0.76	0.130	2.1
EDSR-baseline-LINF-LP (EDSR-baseline)	27.53	0.77	0.121	2.1 + 1.4
EDSR-baseline-LINF-LP (UNet)	**27.64**	**0.78**	**0.119**	2.1 + 4.6
EDSR-baseline-LINF-LP (Swin-T)	27.58	**0.78**	0.120	2.1 + 7.3
RRDB-LINF (t=0.8) [76]	27.33	0.77	0.112	17.5
RRDB-LINF-LP (EDSR-baseline)	27.86	0.78	0.110	17.5 + 1.4
RRDB-LINF-LP (UNet)	**28.00**	0.78	**0.105**	17.5 + 4.6
RRDB-LINF-LP (Swin-T)	27.99	**0.79**	**0.105**	17.5 + 7.3

Table 5. An analysis of the usage of different types of input. "LPIPS$_{in}$" denotes LPIPS [78] scores on the DIV2K 4× SR task, while "LPIPS$_{OOD}$" refers to LPIPS scores on the Urban100 8×SR task. The best and second-best results are highlighted in **red** and blue, respectively.

Method	LR Image	Initial Prior	LPIPS$_{in}$ (↓)	LPIPS$_{OOD}$ (↓)
LINF [76]	✗	✗	0.130	0.386
LINF-LP (Ours)	✓	✗	0.201	0.413
LINF-LP (Ours)	✗	✓	0.125	0.384
LINF-LP (Ours)	✓	✓	**0.119**	**0.377**

bining both, which is our final setting. During the analyses, we keep the total feature dimension at 64 in each experiment to ensure a fair comparison. The results in Table 5 reveal the importance of both image space and latent space information. Firstly, an initial prior is crucial to our proposed framework. By solely adopting an initial prior as input, LINF-LP boosts the performance at both in- and out-of-distribution scales. This suggests an initial prior provides a promising initialization in latent space and eases the prior generation process. Moreover, combining features from both image and latent spaces facilitates the best results, which infers that it is necessary to leverage both image and latent space features to enhance the quality of the learned prior. Lastly, relying solely on image space features is insufficient under our framework, as it is challenging to transform an LR image into a precise latent space signal with a single backbone. This setting leads to a decrease of 0.071 in LPIPS from the baseline on DIV2K 4× SR task.

Objective Function. This analysis assesses the impact of different objective functions. The experiment employs three configurations: (1) using $\mathcal{L}_{latent}$ only, (2) adopting a combination of $\mathcal{L}_{latent}$ and $\mathcal{L}_{percep}$, and (3) exclusively adopting $\mathcal{L}_{percep}$, which is our chosen approach for LINF-LP. Note that in this analysis, the weight of $\mathcal{L}_{latent}$ is set to 0.1 when combined with $\mathcal{L}_{percep}$. The results in Table 6 demonstrate that solely using $\mathcal{L}_{percep}$ yields the best LPIPS results. In contrast, employing $\mathcal{L}_{latent}$ only loss leads to an average prediction [8] of all possible latent codes, resulting in inferior perceptual quality. In addition, the dual application of $\mathcal{L}_{latent}$ and $\mathcal{L}_{percep}$ deteriorates LPIPS score by

Table 6. An analysis of the impact of the objective function. LPIPS and LPIPS$_{OOD}$ represent the LPIPS scores on the DIV2K 4×SR and Urban100 8×SR tasks, respectively. The best results are highlighted in **red**, with the second-best in blue.

Method	$\mathcal{L}_{latent}$	$\mathcal{L}_{percep}$	LPIPS (↓)	LPIPS$_{OOD}$ (↓)
LINF [76]	✗	✗	0.1297	0.3863
LINF-LP (Ours)	✓	✗	0.2658	0.4180
LINF-LP (Ours)	✓	✓	**0.1185**	0.3914
LINF-LP (Ours)	✗	✓	0.1191	**0.3774**

0.014 at out-of-distribution (OOD) scales, yet slightly improves the in-distribution performance. This phenomenon suggests the $\mathcal{L}_{latent}$ enables LINF-LP to better fit the training distribution, as it receives guidance directly from the latent space (*i.e.*, HR ground truth latent codes) during training, making its predictions toward in-distribution outcomes. Given that the performance at OOD scales is crucial to arbitrary-scale SR tasks, we only use $\mathcal{L}_{percep}$ for LINF-LP.

Based on the observation that the model trained with $\mathcal{L}_{latent}$ tends to yield latent codes within the training distribution, we adopt $\mathcal{L}_{latent}$ as a regularization term for SRFlow-LP. This approach aims to prevent SRFlow-LP from generating OOD predictions, which could lead to an exploding inverse as noted in [26]. Under this setting, as illustrated in Table 3, we successfully prevent the occurrence of exploding inverses without modifying the architecture [26] and the pre-trained weights of SRFlow.

6. Conclusion

In this work, we identify several challenges in flow-based SR methods, including grid artifacts, exploding inverses, and suboptimal results due to a fixed sampling temperature. To tackle these issues, we introduce a learned prior, which is predicted by the proposed latent module, to the inference phase of flow-based SR models. This framework not only addresses the inherent issues in flow-based SR models but also enhances the performance of these models without modifying their original design or pre-trained weights. Our proposed framework is effective, flexible in design, and able to generalize to both fixed-scale and arbitrary-scale SR frameworks without requiring customized components.

Acknowledgments

The authors gratefully acknowledge the support from the National Science and Technology Council (NSTC) in Taiwan under grant numbers MOST 111-2223-E-007-004-MY3, as well as the financial support from MediaTek Inc., Taiwan. The authors would also like to express their appreciation for the donation of the GPUs from NVIDIA Corporation and NVIDIA AI Technology Center (NVAITC) used in this work. Furthermore, the authors extend their gratitude to the National Center for High-Performance Computing (NCHC) for providing the necessary computational and storage resources.

References

[1] E. Agustsson and R. Timofte. NTIRE 2017 challenge on single image super-resolution: Dataset and study. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition Workshop (CVPRW)*, pages 1122–1131, 2017. 5, 6, 7, 8

[2] N. Ahn, B. Kang, and K.-A. Sohn. Fast, accurate, and lightweight super-resolution with cascading residual network. In *Proc. European Conf. on Computer Vision (ECCV)*, pages 252–268, 2018. 2

[3] L. Andreas, D. Martin, and T. Radu. Ntire 2021 learning the super-resolution space challenge. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 596–612, 2021. 5

[4] J. Behrmann, P. Vicol, K.-C. Wang, R. Grosse, and J.-H. Jacobsen. Understanding and mitigating exploding inverses in invertible neural networks. In *International Conference on Artificial Intelligence and Statistics*, pages 1792–1800. PMLR, 2021. 1, 2, 7

[5] M. Bevilacqua, A. Roumy, C. Guillemot, and M.-L. Alberi Morel. Low-complexity single-image super-resolution based on nonnegative neighbor embedding. In *Proc. British Machine Vision Conf. (BMVC)*, pages 1–10, 2012. 5, 6, 7

[6] Y. Blau and T. Michaeli. The perception-distortion tradeoff. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018. 1, 6

[7] S. Bond-Taylor, A. Leach, Y. Long, and C. G. Willcocks. Deep generative modelling: A comparative review of vaes, gans, normalizing flows, energy-based and autoregressive models. *IEEE Trans. Pattern Analysis and Machine Intelligence (TPAMI)*, 2021. 3

[8] J. Bruna, P. Sprechmann, and Y. LeCun. Super-resolution with deep convolutional sufficient statistics. *arXiv preprint arXiv:1511.05666*, 2015. 4, 8

[9] Dong C, C. C. Loy, and X. Tang. Accelerating the super-resolution convolutional neural network. In *Proc. European Conf. on Computer Vision (ECCV)*, pages 391–407, 2016. 2

[10] J. Cao, Q. Wang, Y. Xian, Y. Li, B. Ni, Z. Pi, K. Zhang, Y. Zhang, R. Timofte, and L. Van Gool. Ciaosr: Continuous implicit attention-in-attention network for arbitrary-scale image super-resolution. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 1796–1807, 2023. 3

[11] C. Chadebec, L. Vincent, and S. Allassonnière. Pythae: Unifying generative autoencoders in python-a benchmarking use case. *Proc. Conf. on Neural Information Processing Systems (NeurIPS)*, 35:21575–21589, 2022. 3

[12] H.-W. Chen, Y.-S. Xu, M.-F. Hong, Y.-M. Tsai, H.-K. Kuo, and C.-Y. Lee. Cascaded local implicit transformer for arbitrary-scale super-resolution. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 18257–18267, 2023. 3

[13] X. Chen, D. P. Kingma, T. Salimans, Y. Duan, P. Dhariwal, J. Schulman, I. Sutskever, and P. Abbeel. Variational lossy autoencoder. *Proc. Int. Conf. on Learning Representations (ICLR)*, 2017. 3

[14] X. Chen, X. Wang, J. Zhou, Y. Qiao, and C. Dong. Activating more pixels in image super-resolution transformer. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 22367–22377, 2023. 2

[15] Y. Chen, S. Liu, and X. Wang. Learning continuous image representation with local implicit image function. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 8628–8638, 2021. 3, 7

[16] N. Dilokthanakul, P. A. M. Mediano, M. Garnelo, M. C. H. Lee, H. Salimbeni, K. Arulkumaran, and M. Shanahan. Deep unsupervised clustering with gaussian mixture variational autoencoders. *Proc. Int. Conf. on Learning Representations (ICLR)*, 2017. 3

[17] C. Dong, C. C. Loy, K. He, and X. Tang. Image super-resolution using deep convolutional networks. *IEEE Trans. Pattern Analysis and Machine Intelligence (TPAMI)*, 38(2): 295–307, 2016. 2

[18] S. Dong, G. Hangel, E. Z. Chen, S. Sun, W. Bogner, G. Widhalm, C. You, J. A. Onofrey, R. de Graaf, and J. S. Duncan. Flow-based visual quality enhancer for super-resolution magnetic resonance spectroscopic imaging. In *MICCAI Workshop on Deep Generative Models*, pages 3–13, 2022. 3

[19] P. Esser, R. Rombach, and B. Ommer. Taming transformers for high-resolution image synthesis. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 12873–12883, 2021. 3

[20] P. Ghosh, M. S. M. Sajjadi, A. Vergari, M. Black, and B. Scholkopf. From variational to deterministic autoencoders. In *Proc. Int. Conf. on Learning Representations (ICLR)*, 2020. 3

[21] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. *Proc. Conf. on Neural Information Processing Systems (NeurIPS)*, 27, 2014. 3

[22] I. Gulrajani, K. Kumar, F. Ahmed, A. A. Taiga, F. Visin, D. Vazquez, and A. Courville. Pixelvae: A latent variable model for natural images. *Proc. Int. Conf. on Learning Representations (ICLR)*, 2017. 3

[23] B. Guo, X. Zhang, H. Wu, Y. Wang, Y. Zhang, and Y.-F. Wang. Lar-sr: A local autoregressive model for image super-resolution. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 1899–1908, 2022. 3

[24] M. Haris, G. Shakhnarovich, and N. Ukita. Deep back-projection networks for super-resolution. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 1664–1673, 2018. 2

[25] J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models. *Proc. Conf. on Neural Information Processing Systems (NeurIPS)*, 33:6840–6851, 2020. 3

[26] S. Hong, I. Park, and S. Y. Chun. On the robustness of normalizing flows for inverse problems in imaging. In *Proc. IEEE Int. Conf. on Computer Vision (ICCV)*, pages 10745–10755, 2023. 1, 2, 5, 7, 8

[27] X. Hu, H. Mu, X. Zhang, Z. Wang, T. Tan, and J. Sun. Meta-sr: A magnification-arbitrary network for super-resolution. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 1575–1584, 2019. 3, 7

[28] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger. Densely connected convolutional networks. In *Proc. IEEE*

Conf. on Computer Vision and Pattern Recognition (CVPR), 2017. 4

[29] J.-B. Huang, A. Singh, and N. Ahuja. Single image super-resolution from transformed self-exemplars. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 5197–5206, 2015. 5, 6, 7

[30] J. Johnson, A. Alahi, and F.-F. Li. Perceptual losses for real-time style transfer and super-resolution. In *Proc. European Conf. on Computer Vision (ECCV)*, pages 694–711, 2016. 3, 4

[31] S. Karen and Z. Andrew. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 4

[32] T. Karras, S. Laine, and T. Aila. A style-based generator architecture for generative adversarial networks. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 4401–4410, 2019. 3

[33] J. Kim, J. K. Lee, and K. M. Lee. Accurate image super-resolution using very deep convolutional networks. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 1646–1654, 2016. 2

[34] Y. Kim and D. Son. Noise conditional flow model for learning the super-resolution space. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 424–432, 2021. 1, 3

[35] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 3

[36] D. P. Kingma, T. Salimans, R. Jozefowicz, X. Chen, I. Sutskever, and M. Welling. Improved variational inference with inverse autoregressive flow. *Proc. Conf. on Neural Information Processing Systems (NeurIPS)*, 29, 2016. 3

[37] K. Ko, B. Lee, J. Hong, D. Kim, and H. Ko. Mriflow: Magnetic resonance image super-resolution based on normalizing flow and frequency prior. *Journal of Magnetic Resonance*, 352:107477, 2023. 3

[38] I. Kobyzev, S. J. D. Prince, and M. A. Brubaker. Normalizing flows: An introduction and review of current methods. *IEEE Trans. Pattern Analysis and Machine Intelligence (TPAMI)*, pages 3964–3979, 2020. 3

[39] W.-S. Lai, J.-B. Huang, N. Ahuja, and M.-H. Yang. Deep laplacian pyramid networks for fast and accurate super-resolution. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 624–632, 2017. 2

[40] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi. Photo-realistic single image super-resolution using a generative adversarial network. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 105–114, 2017. 2

[41] J. Lee and K. H. Jin. Local texture estimator for implicit representation function. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 1929–1938, 2022. 3, 7

[42] A. Li, L. Zhang, Y. Liu, and C. Zhu. Feature modulation transformer: Cross-refinement of global representation via high-frequency prior for image super-resolution. In *Proc. IEEE Int. Conf. on Computer Vision (ICCV)*, pages 12514–12524, 2023. 2

[43] H. Li, Y. Yang, M. Chang, H. Feng, Z. Xu, Q. Li, and Y. Chen. Srdiff: Single image super-resolution with diffusion probabilistic models. *Neurocomputing*, 479:47–59, 2022. 3, 6

[44] J. Liang, J. Cao, G. Sun, K. Zhang, L. Van Gool, and R. Timofte. Swinir: Image restoration using swin transformer. In *Proc. IEEE Int. Conf. on Computer Vision Workshop (ICCVW)*, pages 1833–1844, 2021. 2

[45] J. Liang, A. Lugmayr, K. Zhang, M. Danelljan, L. Van Gool, and R. Timofte. Hierarchical conditional flow: A unified framework for image super-resolution and image rescaling. In *Proc. IEEE Int. Conf. on Computer Vision (ICCV)*, pages 4056–4065, 2021. 1, 3, 6

[46] J. Liang, K. Zhang, S. Gu, L. Van Gool, and R. Timofte. Flow-based kernel prior with application to blind super-resolution. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 10601–10610, 2021. 3

[47] Jie Liang, Hui Zeng, and Lei Zhang. Details or artifacts: A locally discriminative learning approach to realistic image super-resolution. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 5657–5666, 2022. 1

[48] B. Lim, S. Son, H. Kim, S. Nah, and K. M. Lee. Enhanced deep residual networks for single image super-resolution. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition Workshop (CVPRW)*, pages 1132–1140, 2017. 2, 4, 5, 7

[49] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proc. IEEE Int. Conf. on Computer Vision (ICCV)*, pages 10012–10022, 2021. 4, 5, 7

[50] A. Lugmayr, M. Danelljan, L. Van Gool, and R. Timofte. Srflow: Learning the super-resolution space with normalizing flow. In *Proc. European Conf. on Computer Vision (ECCV)*, 2020. 1, 2, 3, 4, 5, 6, 7

[51] A. Lugmayr, M. Danelljan, F. Yu, L. Van Gool, and R. Timofte. Normalizing flow as a flexible fidelity objective for photo-realistic super-resolution. In *Proc. IEEE Winter Conf. on Applications of Computer Vision (WACV)*, pages 874–883, 2022. 1, 3

[52] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proc. IEEE Int. Conf. on Computer Vision (ICCV)*, pages 416–425, 2001. 5, 6, 7

[53] A. Odena, V. Dumoulin, and C. Olah. Deconvolution and checkerboard artifacts. *Distill*, 2016. 1

[54] A. Van Den Oord, O. Vinyals, and K. Kavukcuoglu. Neural discrete representation learning. *Proc. Conf. on Neural Information Processing Systems (NeurIPS)*, 30, 2017. 3

[55] A. Razavi, A. Van den Oord, and O. Vinyals. Generating diverse high-fidelity images with vq-vae-2. *Proc. Conf. on Neural Information Processing Systems (NeurIPS)*, 32, 2019.

[56] D. Rezende and S. Mohamed. Variational inference with normalizing flows. In *Proc. Int. Conf. on Machine Learning (ICML)*, pages 1530–1538, 2015. 3

[57] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention*, pages 234–241, 2015. 4, 5, 7

[58] Park S.-H., Moon Y.-S., and Cho N.-I. Flexible style image super-resolution using conditional objective. *IEEE Access*, 10:9774–9792, 2022. 2

[59] Park S.-H., Moon Y.-S., and Cho N.-I. Perception-oriented single image super-resolution using optimal objective estimation. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 1725–1735, 2023. 2, 6

[60] C. Saharia, J. Ho, W. Chan, T. Salimans, D. J. Fleet, and M. Norouzi. Image super-resolution via iterative refinement. *IEEE Trans. Pattern Analysis and Machine Intelligence (TPAMI)*, PP, 2022. 3

[61] J. Shen and H.-W. Shen. Psrflow: Probabilistic super resolution with flow-based models for scientific data. *IEEE Transactions on Visualization and Computer Graphics*, 2023. 3

[62] W. Shi, J. Caballero, F. Huszar, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 1874–1883, 2016. 2

[63] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *Proc. Int. Conf. on Machine Learning (ICML)*, pages 2256–2265, 2015. 3

[64] Y. Song and S. Ermon. Generative modeling by estimating gradients of the data distribution. *Proc. Conf. on Neural Information Processing Systems (NeurIPS)*, 32, 2019.

[65] Y. Song and S. Ermon. Improved techniques for training score-based generative models. *Proc. Conf. on Neural Information Processing Systems (NeurIPS)*, 33:12438–12448, 2020.

[66] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole. Score-based generative modeling through stochastic differential equations. *Proc. Int. Conf. on Learning Representations (ICLR)*, 2020. 3

[67] K.-U. Sung, D. Shim, K.-W. Kim, J.-Y. Lee, and Y. Kim. Fs-ncsr: Increasing diversity of the super-resolution space via frequency separation and noise-conditioned normalizing flow. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition Workshop (CVPRW)*, pages 967–976. IEEE, 2022. 1, 3

[68] R. Timofte, E. Agustsson, L. Van Gool, M.-H. Yang, and L. Zhang. NTIRE 2017 challenge on single image super-resolution: Methods and results. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition Workshop (CVPRW)*, pages 1110–1121, 2017. 5

[69] J. Tomczak and M. Welling. Vae with a vampprior. In *International Conference on Artificial Intelligence and Statistics*, pages 1214–1223, 2018. 3

[70] H. Wang, X. Chen, B. Ni, Y. Liu, and J. Liu. Omni aggregation networks for lightweight image super-resolution. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 22378–22387, 2023. 2

[71] X. Wang, K. Yu, C. Dong, and C. C. Loy. Recovering realistic texture in image super-resolution by deep spatial feature transform. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 606–615, 2018. 2

[72] X. Wang, K. Yu, S. Wu, J. Gu, Y. Liu, C. Dong, Y. Qiao, and C. C. Loy. Esrgan: Enhanced super-resolution generative adversarial networks. In *Proc. European Conf. on Computer Vision Workshop (ECCVW)*, pages 63–79, 2018. 3, 5, 6

[73] H. Wu, N. Ni, S. Wang, and L. Zhang. Blind super-resolution for remote sensing images via conditional stochastic normalizing flows. *arXiv preprint arXiv:2210.07751*, 2022. 3

[74] X. Xu, Z. Wang, and H. Shi. Ultrasr: Spatial encoding is a missing key for implicit image function-based arbitrary-scale super-resolution. *CoRR*, abs/2103.12716, 2021. 3

[75] J. Yang, S. Shen, H. Yue, and K. Li. Implicit transformer network for screen content image continuous super-resolution. In *Proc. Conf. on Neural Information Processing Systems (NeurIPS)*, pages 13304–13315, 2021.

[76] J.-E. Yao, L.-Y. Tsao, Y.-C. Lo, R. Tseng, C.-C. Chang, and C.-Y. Lee. Local implicit normalizing flow for arbitrary-scale image super-resolution. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 1776–1785, 2023. 1, 2, 3, 4, 5, 6, 7, 8

[77] R. Zeyde, M. Elad, and M. Protter. On single image scale-up using sparse-representations. In *Curves and Surfaces*, pages 711–730, 2010. 5, 6, 7

[78] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 586–595, 2018. 1, 5, 6, 7, 8

[79] W. Zhang, Y. Liu, C. Dong, and Y. Qiao. Ranksrgan: Generative adversarial networks with ranker for image super-resolution. In *Proc. IEEE Int. Conf. on Computer Vision (ICCV)*, pages 3096–3105, 2019. 3, 6

[80] Y. Zhang, K. Li, K. Li, L. Wang, B. Zhong, and Y. Fu. Image super-resolution using very deep residual channel attention networks. In *Proc. European Conf. on Computer Vision (ECCV)*, pages 286–301, 2018. 2

[81] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, and Y. Fu. Residual dense network for image super-resolution. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 2472–2481, 2018. 2

[82] Y. Zhou, Z. Li, C.-L. Guo, S. Bai, M.-M. Cheng, and Q. Hou. Srformer: Permuted self-attention for single image super-resolution. *Proc. IEEE Int. Conf. on Computer Vision (ICCV)*, 2023. 2