On the Robustness of Normalizing Flows for Inverse Problems in Imaging

Seongmin Hong[1], Inbum Park[1], Se Young Chun[1,2,*]
[1]Dept. of ECE, [2]INMC & IPAI, Seoul National University, Republic of Korea
{smhongok, inbum0215, sychun}@snu.ac.kr

Abstract

Conditional normalizing flows can generate diverse image samples for solving inverse problems. Most normalizing flows for inverse problems in imaging employ the conditional affine coupling layer that can generate diverse images quickly. However, unintended severe artifacts are occasionally observed in the output of them. In this work, we address this critical issue by investigating the origins of these artifacts and proposing the conditions to avoid them. First of all, we empirically and theoretically reveal that these problems are caused by "exploding inverse" in the conditional affine coupling layer for certain out-of-distribution (OOD) conditional inputs. Then, we further validated that the probability of causing erroneous artifacts in pixels is highly correlated with a Mahalanobis distance-based OOD score for inverse problems in imaging. Lastly, based on our investigations, we propose a remark to avoid exploding inverse and then based on it, we suggest a simple remedy that substitutes the affine coupling layers with the modified rational quadratic spline coupling layers in normalizing flows, to encourage the robustness of generated image samples. Our experimental results demonstrated that our suggested methods effectively suppressed critical artifacts occurring in normalizing flows for super-resolution space generation and low-light image enhancement.

1. Introduction

Deep learning techniques have demonstrated great potential for solving *ill-posed* inverse problems in imaging [25, 31]. Among them, conditional normalizing flow (NF)-based methods have a unique advantage over other deep learning methods, which is the capability of generating diverse solutions for a given input. Conditional NFs [6] have been explored for various inverse problems in imaging such as super-resolution space generation [26, 14, 38, 13, 22, 29, 27, 28], low-light image enhancement [42, 41], guided image generation [3, 35], image dehazing [45], denoising [1, 24] and inpainting [24]. Most of these prior works with conditional NFs for image processing and low-level computer vision have focused on excellent performance with diverse solutions.

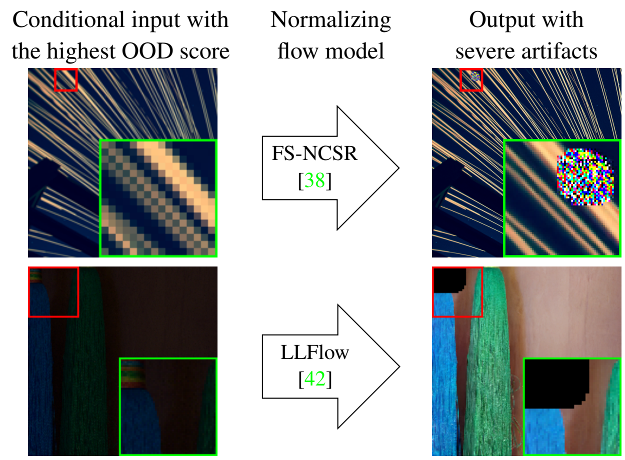


Figure 1: Demonstration of the occasional errors in normalizing flows solving inverse problems in imaging. The left images are the conditional inputs of normalizing flows (DIV2K 828 and LOL 179) with the highest OOD scores (11) and the right images are the outputs of them for super-resolution space generation and low-light image enhancement, displaying severe artifacts.

Existing conditional NFs for inverse problems in imaging occasionally generate unintended erroneous image samples. In super-resolution space generation, similar artifacts were observed in multiple independent works [27, 28]. For example, Song *et al.* reported that those artifacts occurred for more than 2% of all test images [38] and we confirmed that these artifacts occasionally appear as illustrated in the top row of Figure 1. Unintended artifacts were also observed in another computer vision task with conditional NFs. In low-light image enhancement [42], we also revealed that black regions with `Inf` values sometimes occur for certain conditional inputs as we sample diverse images as in the bottom row of Figure 1. In *unconditional* NFs,

[*]Corresponding author

such as Glow [15], similar artifacts called "exploding inverse" were observed [4], which were known to occasionally occur only when the training and test sets come from different distributions (*e.g.*, training with CIFAR-10 [20], testing with tinyImageNet [46]). However, in conditional NFs for inverse problems in imaging, artifacts may sometimes occur even when the training and test sets follow the same distribution, suggesting that the existing "exploding inverse" is insufficient to explain this phenomenon.

In this work, we address the robustness issue for solving inverse problems in imaging using conditional NFs by investigating the origins of these artifacts and proposing how to avoid them. Firstly, we empirically and theoretically reveal that artifacts arising from conditional NFs for inverse problems are caused by a mechanism very similar to that of unconditional NFs' exploding inverses [4]. This implies that although the conditional inputs that yielded exploding inverses are sampled from the same distribution as the training dataset, they may be out-of-distribution (OOD) data from the perspective of the conditioning network. We then validate this remark (Remark 1) by showing that the probability of causing erroneous pixels is highly correlated with a Mahalanobis distance-based OOD score [21] for inverse problems. Lastly, based on our investigations, we propose another remark (Remark 2) on how to avoid the exploding inverses in conditional NFs for inverse problems in imaging. As a simple remedy to meet the criteria of our remark, we suggest substituting the affine coupling layers with the modified rational-quadratic (RQ) spline coupling layers [8] in NFs, to encourage the robustness of generated image samples. Our experimental results demonstrated that our suggested methods effectively suppressed exploding inverses often occurring in conditional NFs for super-resolution space generation and low-light image enhancement. The contributions of this paper are summarized as follows:

- Revealing theoretically and experimentally that exploding inverses also occur in conditional affine coupling flows for inverse problems occasionally, even when the training and test dataset are sampled from the same distribution.
- Investigating that exploding inverses are more likely to occur when encoded conditional inputs are out-of-distribution.
- Proposing a remark on how to avoid the exploding inverses in conditional NFs and demonstrating how to use it by considering other factors such as performance.
- Demonstrating that the proposed method effectively suppressed erroneous samples in 2D toy experiment, super-resolution space generation and low-light image enhancement.

2. Preliminaries

2.1. Conditional normalizing flow

NFs [36, 32] learn a probability distribution from a dataset and can be used as both samplers and density estimators. Let $\mathcal{D}$ be a dataset from the true target probability distribution $p_{\mathbf{x}}$. One can utilize NF by fitting a flow-based model $q_{\mathbf{x}}$ to the true target distribution $p_{\mathbf{x}}$ using a simple base probability distribution $q_{\mathbf{z}}$ (*e.g.*, standard normal distribution) and a diffeomorphic (*i.e.*, invertible and differentiable) mapping $f_{\boldsymbol{\theta}} : \mathcal{X} \rightarrow \mathcal{Z}$ where $\mathcal{X}$ and $\mathcal{Z}$ are compact subsets of $\mathbb{R}^D$ with the following density transformation:

$$q_{\mathbf{x}}(\mathbf{x}) = q_{\mathbf{z}}(f_{\boldsymbol{\theta}}(\mathbf{x})) \left| \det \frac{\partial f_{\boldsymbol{\theta}}}{\partial \mathbf{x}}(\mathbf{x}) \right|. \quad (1)$$

For $\mathcal{D} = \{\mathbf{x}^{(n)}\}_{n=1}^{N}$ where $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(N)}$ are samples from $p_{\mathbf{x}}$, NFs are trained on $\mathcal{D}$ by minimizing the following negative log-likelihood (NLL):

$$\mathcal{L}_{\text{NLL}} = -\frac{1}{N} \sum_{n=1}^{N} \log q_{\mathbf{x}}(\mathbf{x}^{(n)}) \xrightarrow{N \rightarrow \infty} -\mathbb{E}_{\mathbf{x} \sim p_{\mathbf{x}}}[\log q_{\mathbf{x}}(\mathbf{x})]. \quad (2)$$

Conditional NFs can be defined by simply changing the network in (1) to be conditional to $\mathbf{y}$ so that

$$q_{\mathbf{x}|\mathbf{y}}(\mathbf{x}|\mathbf{y}) = q_{\mathbf{z}}(f_{\boldsymbol{\theta}}(\mathbf{x};\mathbf{y})) \left| \det \frac{\partial f_{\boldsymbol{\theta}}}{\partial \mathbf{x}}(\mathbf{x};\mathbf{y}) \right|. \quad (3)$$

For inverse problems in imaging, (3) is equivalent to modeling the posterior distribution $p_{\mathbf{x}|\mathbf{y}}(\mathbf{x}|\mathbf{y})$ where $\mathbf{y}$ is a corrupted measurement from a clean image $\mathbf{x}$.

2.2. Coupling transformations

For a given $\mathbf{y}$, conditional NFs can obtain multiple possible $\mathbf{x}$ when used as samplers with different $\mathbf{z} \sim q_{\mathbf{z}}$ thanks to the one-to-one relationship between $\mathbf{z}$ and $\mathbf{x}$. However, ensuring this one-to-one relationship has limited network structures. In order for NFs to be efficiently trainable with the NLL (2), $f_{\boldsymbol{\theta}}$ must not only be invertible, but also have a tractable Jacobian determinant. Although many successful deep learning networks in the image domain employ 3×3 convolution, max-pooling and ReLU (Rectified Linear Unit) layers, NFs cannot employ them since such layers are not invertible. A number of studies have found that only a few layers are appropriate for NFs in the image domain [5, 36, 6, 33, 16, 15].

The conditional coupling layers are what make the NF "conditional"; hence they are frequently used as the main layer. A conditional coupling transformation [5, 6] $\phi : \Omega \rightarrow$

$\Omega \subseteq \mathbb{R}^D$ is defined as

$$\phi(\mathbf{x})_i = \begin{cases} c(x_i; \mathbf{h}_i) & \text{for } i = d, \ldots, D, \\ x_i & \text{for } i = 1, \ldots, d-1, \end{cases} \tag{4}$$

where $\mathbf{h}_i = \text{NN}(x_{1:d-1}, g_{\boldsymbol{\theta}}(\mathbf{y}))$, NN is an arbitrary neural network, $g_{\boldsymbol{\theta}}$ is an encoder for the conditional input $\mathbf{y}$, and $c(\cdot; \mathbf{h}(\mathbf{y})) : \Omega' \to \Omega' \subseteq \mathbb{R}$ is an invertible function parameterized by a vector $\mathbf{h}$. The Jacobian determinant of this transformation is easily obtained from the derivative of c, expressed as $\det(\partial\phi/\partial\mathbf{x}) = \Pi_{i=d}^{D} \partial c(x_i; \mathbf{h}_i)/\partial x_i$. The inverse of ϕ is obtained as

$$\phi^{-1}(\mathbf{x})_i = \begin{cases} c^{-1}(x_i; \mathbf{h}_i) & \text{for } i = d, \ldots, D, \\ x_i & \text{for } i = 1, \ldots, d-1. \end{cases} \tag{5}$$

Many works employ affine transformations as c:

$$c(x_i; \mathbf{h}_i(\mathbf{y})) = s_i(\mathbf{y})x_i + t_i(\mathbf{y}) \tag{6}$$

where $\mathbf{h}_i = (s_i, t_i)$ due to computational efficiency for their Jacobian and inverse as well as sufficient expressive power [17, 6]. Thus, affine coupling transformations are suitable for generating images [15, 11, 26, 39] and speeches [34, 10] with large dimensions D.

There are also other coupling transformations for conditional NFs such as splines [30, 9, 7, 8, 37] or sigmoids [19], which are more complex than affine coupling transformations. Splines are diffeomorphic piecewise polynomials or rational functions. Even though employing conditional spline-based coupling layers is possible, splines or sigmoids are computationally challenging over affine transformations. Thus, they were not as popular as affine coupling transformations for imaging applications, but rather prominent in modeling probability distributions of smaller dimensions such as molecular structures [44, 19].

2.3. Conditional NFs for inverse problems

In most conditional NFs, the affine coupling transformations and affine injectors are the only components that depend on $\mathbf{y}$. NFs with this structure have successfully solved various inverse problems in imaging [26, 38, 42, 14, 41].

SRFlow [26] has achieved excellent performance on super-resolution space generation by adapting Glow [15] as its backbone. It has also been extended to many variants such as [13, 22, 14, 29, 38]. In low-light image enhancement, LLFlow [42] and TSFlow [41] have successfully utilized conditional NF to reconstruct normally exposed images from low-quality inputs. Some studies dealt with other inverse problems such as inpainting [24], dehazing [45], denoising [24] and colorization [3].

3. On the Robustness of Conditional NFs

We revisited the work of Behrmann *et al.* [4] for exploding inverses in unconditional NFs and described the clear differences between that and our work on exploding inverses in conditional NFs. With a simple toy example, we identified that exploding inverses can also occur in the conditional NF using affine coupling layer, if the conditional input is OOD. Then, we analyzed that the exploding inverse can be induced in the full conditional NF models, when the conditional inputs are OOD from the perspective of the conditional input encoder, even though they are in-distribution in human's perspective (*i.e.*, $g_{\boldsymbol{\theta}}(\mathbf{y})$ is OOD even though $\mathbf{y}$ is in-distribution). We further investigated this phenomenon for two concrete examples: super-resolution space generation (FS-NCSR [38]) and low-light image enhancement (LLFlow [42]). Lastly, we elaborate the conditions on how to avoid the exploding inverse for inverse problems in imaging and suggest a remedy to meet all the criteria.

3.1. Exploding inverses in unconditional NFs

Behrmann *et al.* [4] discovered and named "exploding inverse" in unconditional NFs. We revisit this work and reorganize the relevant parts of their work as follows.

Proposition 1. ***(Exploding inverses in unconditional NFs)*** *If $f_\theta : \mathcal{X} \to \mathcal{Z} \subseteq \mathbb{R}^D$ is an unconditional NF using the affine coupling transformation and trained with a dataset from the distribution $p_\mathbf{x}$, then there exist many $\mathbf{x} \not\sim p_\mathbf{x}$ s.t. $\|\mathbf{x}\|_\infty \ll \|f_\theta^{-1}(f_\theta(\mathbf{x}))\|_\infty$.*

Since $\mathbf{x} \not\sim p_\mathbf{x}$ suggests $f_\theta(\mathbf{x}) \not\sim q_\mathbf{z}$, it is reasonable to say that errors can occur. The problem we address in this work is clearly different. It can be summarized as follows.

Proposition 2. ***(Artifacts in conditional NFs)*** *If $f_\theta : \mathcal{X} \times \mathcal{Y} \to \mathcal{Z} \subseteq \mathbb{R}^D$ is a conditional NF using the conditional affine coupling transformation and trained with a dataset from the distribution $p_{\mathbf{x}|\mathbf{y}}$, then there exist many $\mathbf{y} \sim p_\mathbf{y}$, $\mathbf{z} \sim q_\mathbf{z}$ such that $f_\theta^{-1}(\mathbf{z}; \mathbf{y})$ is erroneous.*

The most significant difference between these two propositions is that the sample (*i.e.*, $f_\theta^{-1}(\mathbf{z}; \mathbf{y})$) can be erroneous even though the inputs (*i.e.*, $\mathbf{y}$ and $\mathbf{z}$) are in-distribution in human's perspective. To help understanding where these errors come from, we build and verify this proposition, and then investigate which $\mathbf{y}$ generates artifacts in the next subsections.

3.2. Exploding inverse in conditional NFs

3.2.1 A 2D toy experiment

We constructed a simple 2D toy experiment, demonstrating that the conditional affine coupling flows can suffer from exploding inverse for OOD conditional inputs. A forward model of the inverse problem is selected as follows:

$$\mathbf{y}_{\text{in}} = \mathbf{A}\mathbf{x} + \boldsymbol{\epsilon}, \mathbf{A} = \begin{bmatrix} 0.7 & 0.3 \\ 0.3 & 0.7 \end{bmatrix}, \boldsymbol{\epsilon} \sim \mathcal{N}(0, \sigma_n^2 \mathbf{I}) \tag{7}$$

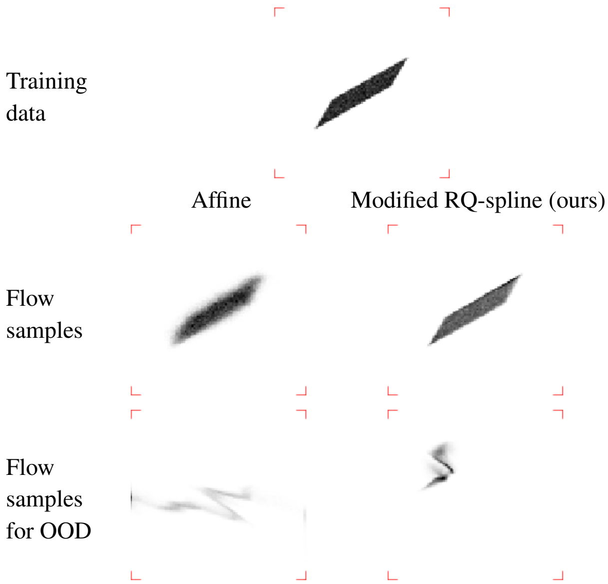
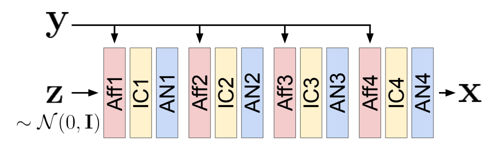


Figure 2: 2D toy experiment results. The first row shows the training data (uniformly distributed). The second and third row show the flow samples for in-distribution/OOD conditional input (*i.e.*, $\mathbf{y}_{\text{in}}$ and $\mathbf{y}_{\text{OOD}}$), respectively. The left and right columns show the results of employing the conditional affine/RQ-spline coupling layers, respectively. The displayed area is $[-1, 1]^2$, marked with red angle brackets.

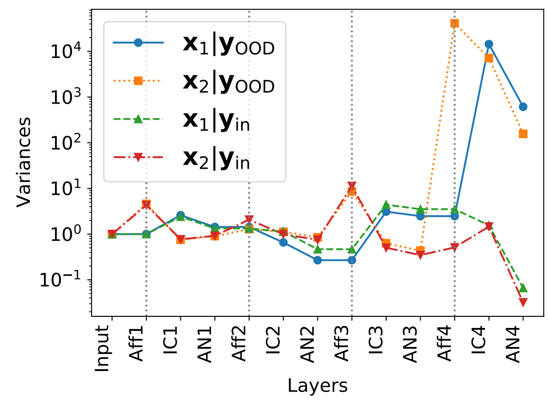
where $\sigma_n = 0.01$, $\mathbf{x}, \mathbf{y}_{\text{in}} \in \mathbb{R}^2$. Training data was generated as illustrated in the first row of Figure 2. We also generated OOD conditional input $\mathbf{y}_{\text{OOD}}$ by shifting $\mathbf{y}_{\text{in}}$ as $\mathbf{y}_{\text{OOD}} = \mathbf{y}_{\text{in}} + \begin{bmatrix} 0.8 & -0.8 \end{bmatrix}^{\text{T}}$. See the supplementary material for further information on the toy experiment.

In the left column of Figure 2, flow samples for in-distribution (*i.e.*, $\mathbf{y}_{\text{in}}$) show that the flow model in Figure 3a learned the distribution well, but flow samples for OOD (*i.e.*, $\mathbf{y}_{\text{OOD}}$) show that the flow model failed to learn the distribution correctly. Although the support of the distribution of $\mathbf{x}$ was a subset of a small region (*i.e.*, $\text{supp}(p_{\mathbf{x}|\mathbf{y}}(\mathbf{x}|\mathbf{y}_{\text{in}})) \subset [-1, 1]^2$), the flow model generated samples that were located outside the region (*i.e.*, $\text{supp}(q_{\mathbf{x}|\mathbf{y}}(\mathbf{x}|\mathbf{y}_{\text{OOD}})) \not\subset [-1, 1]^2$). Note that we put all samples outside the region $[-1, 1]^2$ on the edges. This corresponds to clipping the pixel value of image samples to $[0, 1]$ (for unsigned 8-bit integer, $[0, 255]$), which can explain the saturated color compositions of the artifacts that have been observed in conditional NF-based methods for super-resolution space generation and low-light image enhancement as illustrated in the right column of Figure 1.

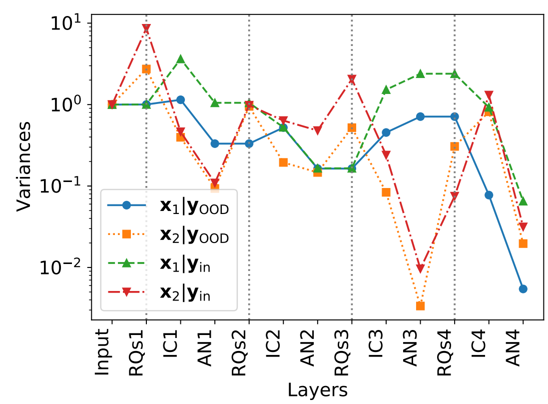
We further investigated this instability problem by looking into the variances of features (Figure 3b) in each layer of the flow model (Figure 3a) where the conditional inputs are either in-distribution or OOD. Before the 4th affine coupling layer (Aff4), features have very similar variances at each layer. However, at Aff4, the variance explodes more than 10,000 times for the OOD case, whereas the variance is maintained for the in-distribution case. Even though our setting was different from unconditional NFs, the cause and the results are very similar to "exploding inverse" [4]. Thus, these results support the following remark:



(a) Network architecture



(b) Variances of features



(c) Variances of features

Figure 3: (a) Network architecture for toy experiment. (b) Variances of features for in-distribution and OOD conditional inputs. Aff, IC and AN denote the conditional affine coupling, invertible 1×1 convolution and activation normalization layers, respectively. (c) Variances of features for in-distribution and OOD conditional inputs, replacing the affine coupling layers to the RQ-spline coupling layers, denoted as RQs.

Remark 1. *Conditional NFs with affine coupling layers can generate erroneous samples due to exploding inverse for certain OOD conditional input.*

Section 3.2.2 presents a theoretical analysis with a simplified model for exploding inverse to support Remark 1. Section 3.3 verifies that Remark 1 is valid in full-size networks for real inverse problems in imaging.

3.2.2 Theoretical analysis on exploding inverse

From the convex optimization perspective, we explain why the exploding inverse occurs for certain conditional inputs.

As in (2), $f_{\boldsymbol{\theta}}$ is trained by minimizing $\mathcal{L}_{\text{NLL}} = -\mathbb{E}_{\mathbf{x},\mathbf{y}\sim\hat{p}_{\mathbf{x},\mathbf{y}}}[\log q_{\mathbf{x}|\mathbf{y}}(\mathbf{x}|\mathbf{y})]$. For simplicity, we assume that a model $f_{\boldsymbol{\theta}}$ consists of one conditional affine coupling layer where $\mathbf{x},\mathbf{y} \in \mathbb{R}^2$. Using (3) with (4) and (6) when $d = D = 2$, we obtain the following NLL loss $\mathcal{L}_{\text{NLL}}$

$$\begin{aligned} &= -\mathbb{E}_{\mathbf{x},\mathbf{y}}\left[\log q_{\mathbf{z}}(f_{\boldsymbol{\theta}}(\mathbf{x};\mathbf{y})) + \log\left|\det\frac{\partial f_{\boldsymbol{\theta}}}{\partial \mathbf{x}}(\mathbf{x};\mathbf{y})\right|\right] \\ &= \mathbb{E}_{\mathbf{x},\mathbf{y}}\left[\frac{\|f_{\boldsymbol{\theta}}(\mathbf{x};\mathbf{y})\|_2^2}{2\sigma_z^2} - \log\left|\det\begin{bmatrix}1 & 0\\ * & s_1\end{bmatrix}\right|\right] \\ &= \mathbb{E}_{\mathbf{x},\mathbf{y}}\left[\frac{x_1^2 + (s_1x_1+t_1)^2}{2\sigma_z^2} - \log(s_1)\right], \end{aligned} \tag{8}$$

where $\mathbf{z}$ is assumed to be Gaussian and (s_1, t_1) are the functions of $\mathbf{y}$. Thus, (8) is a convex function of (s_1, t_1). This NLL loss is unbounded below, so that there is a degenerative case for (s_1, t_1), *i.e.*, $s_1 \to \infty$ with $t_1 \to -s_1x_1$. Similarly, Kirichenko *et al.* [18] reported that s often diverges to infinity in the affine coupling layer, which led to performance degradation in density estimation.

This undesirable unboundedness of the NLL loss can be avoided by setting an upper bound on s_1 so that the optimization problem becomes proper (*i.e.*, bounded below):

$$\min_{0<s_1\leq 1, t_1} \frac{x_1^2 + (s_1x_1+t_1)^2}{2\sigma_z^2} - \log(s_1), \tag{9}$$

which has the analytic solution $(s_1, t_1) = (1, -x_1)$. Interestingly, recent flow models such as SRFlow [26] and LLFlow [42] set an upper bound of s_i to 1 without any theoretical discussion like our analysis with (9).

The exploding inverse that was observed in Figure 3 can be explained theoretically with the convex optimization (9). For the in-distribution conditional input $\mathbf{y}$, $(s_1(\mathbf{y}), t_1(\mathbf{y}))$ will usually yield values close to the optimal point $(1, -x_1)$. Since s_1 is close to 1, the conditional affine coupling layer will not increase the variance of features much. However, for the OOD conditional input, there may be some $(s_1(\mathbf{y}), t_1(\mathbf{y}))$ that are far from the optimal point $(1, -x_1)$, which would result in $s_1 \ll 1$. Considering the sampling process, which is the reverse process of density estimation as in (5), this can significantly increase the variance of features since $1/s_1 \gg 1$, which causes exploding inverse and thus generates erroneous images. One may think that setting a proper lower bound on s_1 could resolve this issue (*e.g.*, $0.1 < s_1 \leq 1.1$ in (9)). Section 4 provides the experimental results that this naïve approach cannot solve it.

3.3. OOD conditional inputs for conditional NFs

Here we verify that certain conditional inputs that cause errors are OOD, even though they are in-distribution in Human's eye. Specifically, we check the difference between $\mathbf{y}_{\text{in}}$ and $\mathbf{y}_{\text{OOD}}$ by investigating the encoder output $g_{\boldsymbol{\theta}}$ for the conditional input $\mathbf{y}$. The Mahalanobis distance [21] was selected to measure these differences. The Mahalanobis distance of a point $\mathbf{v}$ from a probability measure p is defined as

$$d_M(\mathbf{v}, p) = \sqrt{(\mathbf{v}-\mu_p)^{\mathrm{T}}\Sigma_p^{-1}(\mathbf{v}-\mu_p)}, \tag{10}$$

where $\mu_p = \mathbb{E}_{\mathbf{u}\sim p}[\mathbf{u}]$, $\Sigma_p = \mathbb{E}_{\mathbf{u}\sim p}[\mathbf{u}\mathbf{u}^{\mathrm{T}}]$. To check the difference from the perspective of $g_{\boldsymbol{\theta}}$ rather than the data themselves, we compare $d_M(g_{\boldsymbol{\theta}}(\mathbf{y}_{\text{in}}), g_{\boldsymbol{\theta}\#}\hat{p}_{\mathbf{y}})$ and $d_M(g_{\boldsymbol{\theta}}(\mathbf{y}_{\text{OOD}}), g_{\boldsymbol{\theta}\#}\hat{p}_{\mathbf{y}})$, where $g_{\boldsymbol{\theta}\#}\hat{p}_{\mathbf{y}}$ is a pushforward of $\hat{p}_{\mathbf{y}} = (1/N)\sum_{j=1}^{N}\delta_{\mathbf{y}^{(j)}}$ with respect to $g_{\boldsymbol{\theta}}$. For simplicity, we denote our OOD score as follows:

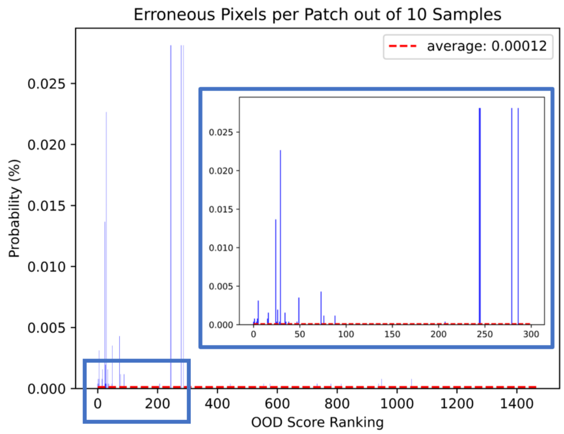
$$s_{\text{OOD}}(\mathbf{y}') = d_M(g_{\boldsymbol{\theta}}(\mathbf{y}'), g_{\boldsymbol{\theta}\#}\hat{p}_{\mathbf{y}}). \tag{11}$$

We calculate s_{OOD} where $\mathbf{y}'$ is a cropped patch in the test set and $\hat{p}_{\mathbf{y}}$ is the distribution of the training set.

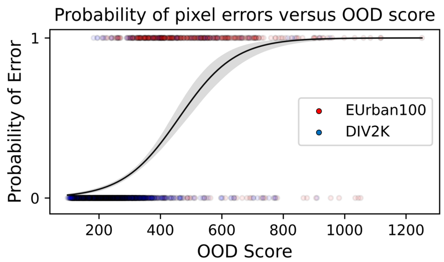
Figure 1 shows erroneous samples generated from the conditional inputs with the highest OOD score in the test set. Both conditional inputs generated erroneous samples. To further validate that conditional inputs with high OOD score are prone to generate exploding inverses, we plotted the probability of pixel errors versus the OOD score in Figure 4. See the supplementary material for more details. Figure 4a shows that the top 300 ranked patches among the DIV2K validation set (*i.e.*, test set from the same distribution as the training set) generally have higher probability of generating erroneous pixels compared to the dashed horizontal line, which denotes the average error probability for all patches. To investigate severe OOD cases, we utilized the Enhanced Urban100 (EUrban100) dataset [12], which can be perceived as a prominent example of severe OOD to the human eye. The logistic regression result shows that conditional inputs with high OOD score are prone to generate pixel errors. To summarize, conditional inputs which frequently generates erroneous images are OOD in the perspective of the conditioning network $g_{\boldsymbol{\theta}}$.

3.4. On how to avoid exploding inverse

From the experimental and theoretical investigations in the previous subsections 3.2 and 3.3, we propose a remark on how to avoid exploding inverse. For a diffeo-



(a) Error pixel rate sorted by s_{OOD} ranking



(b) Logistic regression: $\exists$ Error (Pixel) vs. s_{OOD}

Figure 4: (a) Erroneous pixels for the patches (*i.e.*, conditional inputs) ranked with their OOD scores (s_{OOD}) and their average (dashed horizontal line). (b) Logistic regression: the probability of existence of pixel error versus s_{OOD}.

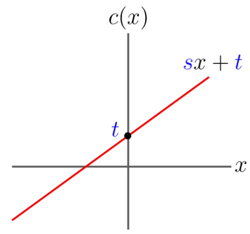
morphic function of a conditional coupling transformation $c : \mathbb{R} \to \mathbb{R}$, let $c'(x) \in [s_l, s_u]$ (*i.e.*, c is bi-Lipschitz continuous). By generalizing the optimization problem for the affine transformation to this function c, the same phenomenon as that in Section 3.2.2 can be observed:

$$c'(x) \begin{cases} \simeq s_u & \text{for in-distribution } \mathbf{y}, \\ \ll s_u & \text{for OOD } \mathbf{y}. \end{cases} \tag{12}$$

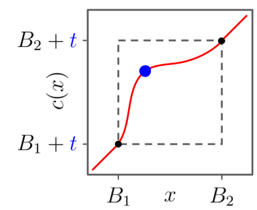
To avoid exploding inverse, coupling transformations must satisfy the following remark:

Remark 2. *To avoid exploding inverse, the derivatives of the element-wise transformation c of the conditional coupling layer must yield similar lower and upper bounds when the input has a sufficiently large absolute value. In other words, $c'(x) \simeq s_u$ for $\mathbf{y}$ is OOD and $|x| \gg 1$.*

While Remark 2 can guide one to design or to select a proper coupling transformation for conditional NFs to avoid exploding inverse, there are also other conditions to consider for performance such as sufficient expressive power and computational efficiency. In this work, we demonstrate how to select a coupling transformation considering both Remark 2 and other conditions like performance among existing ones. The same rules can be utilized for designing a new one.



(a) Affine



(b) Modified RQ-spline (ours)

Figure 5: (a) Affine and (b) the modified RQ-spline transformations for coupling layers. The learnable parameters are labeled in blue.

A solution to satisfy Remark 2: We can set $c(x) = x + t$ for all $x \notin (B_1, B_2)$, which satisfies Remark 2 ($c'(x) = 1$ for all $x \in (-\infty, B_1) \cup (B_2, \infty)$) and computational efficiency (by using only a few parameters). However, this would not have sufficient expressive power. The additive coupling transformation [5] also lacks expressive power. In the meanwhile, spline-based transformations have better expressive power than affine transformations [30, 9, 7, 8, 37], but with inefficient, relatively long computation.

RQ-spline coupling layer [8] satisfies Remark 2 and has sufficient expressive power. Figure 5 compares the affine and RQ-spline transformations. As in Figure 5b, we set only one out of three knots as learnable parameter to be computationally efficient. To maintain the expressive power with a small number of parameters, we propose to add a bias term t (called the modified RQ-spline), which does not violate Remark 2. The modified RQ-spline is defined as

$$c(x) = \begin{cases} c_1(x) + t & \text{for } x \in (B_1, x^*), \\ c_2(x) + t & \text{for } x \in [x^*, B_2), \\ x + t & \text{otherwise,} \end{cases} \tag{13}$$

where c_1 and c_2 are RQ functions satisfying $c_1(x^*) = c_2(x^*) = y^*$, $c_1'(x^*) = c_2'(x^*) = d$, $c_1'(B_1) = c_2'(B_2) = 1$, and (x^*, y^*, d, t) is the output of NN. Note that this method is an example which avoids exploding inverse while having reasonable computational efficiency and expressive power. Other choices or designs could be done for better performance, but our Remark 2 will be an important guideline to avoid potential errors due to exploding inverse.

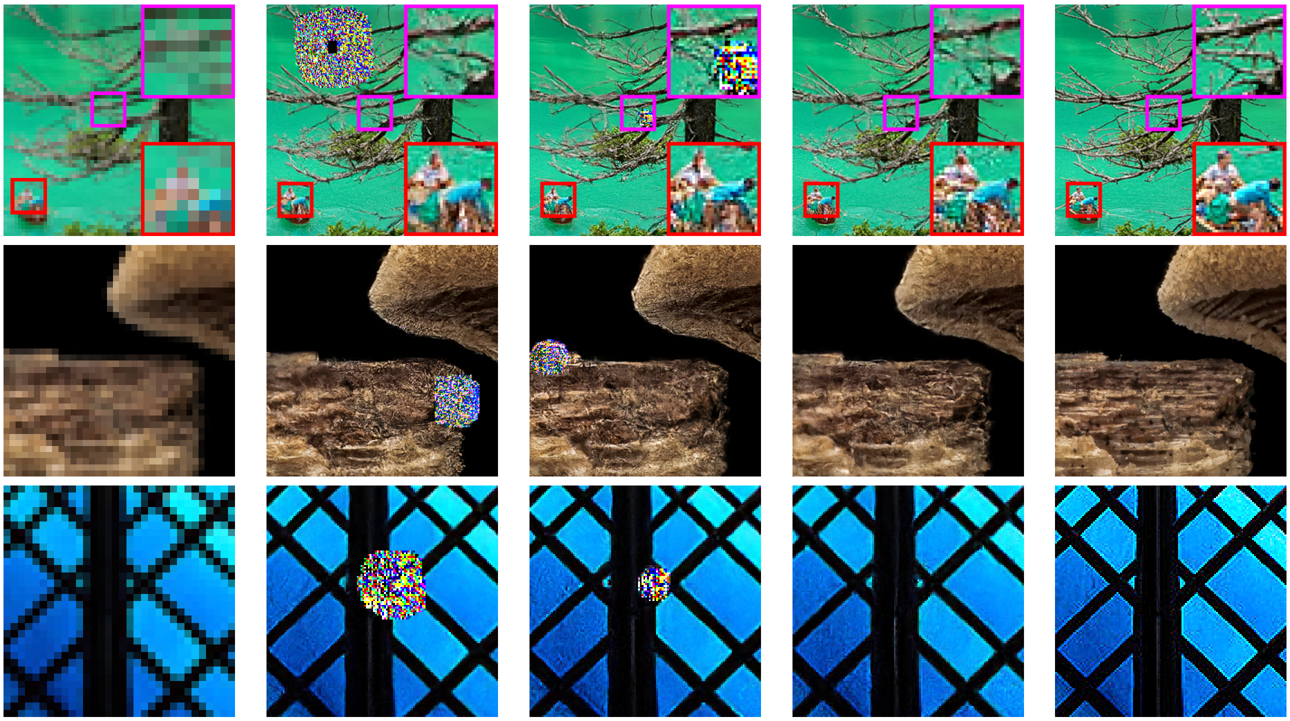


Figure 6: Qualitative comparison of coupling transformation in super-resolution space generation. The first, second, and third row shows the samples from DIV2K [2] 4×, DIV2K 8×, and EUrban100 4×. The † sign denotes that the lower bound of the scale parameter is 0.1.

Train → Test	DF2K 4× → DIV2K 4×			DF2K 8× → DIV2K 8×			DF2K 4× → EUrban100 4× (OOD)		
Model	%Inf ↓	$\overline{\min}$ ↑	$\overline{\sigma}$ ↓	%Inf ↓	$\overline{\min}$ ↑	$\overline{\sigma}$ ↓	%Inf ↓	$\overline{\min}$ ↑	$\overline{\sigma}$ ↓
FS-NCSR [38]	2	50.86	0.202	2	48.47	0.461	30	36.12	3.544
FS-NCSR†	**0**	50.83	0.077	**0**	49.50	0.183	**0**	42.95	1.046
Ours	**0**	**51.10**	**0.012**	**0**	**50.20**	**0.041**	**0**	**44.70**	**0.136**

Table 1: Quantitative comparison. The † sign denotes that the lower bound of the scale parameter is 0.1. '%Inf' refers to the percentage of conditional inputs that generate at least one Inf pixel out of 10 randomly generated latent codes, each with $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \tau^2)$. $\overline{\min}$ and $\overline{\sigma}$ refer the average of the minimum and standard deviation of LR-PSNR, respectively. DF2K means the union set of DIV2K [2] and Flickr2K [40].

4. Experimental Results

4.1. 2D toy experiment

The right column in Figure 2 shows the results of performing the 2D toy experiment in Section 3.2.1 using the proposed modified RQ-spline coupling transformation instead of the affine coupling transformation. We also plotted variances of features in Figure 3c. Unlike the case with affine coupling transformation, the variance does not explode for OOD conditional inputs ($\mathbf{y}_{\text{OOD}}$) when RQ-spline coupling transformation is used. Therefore, all samples are included within the range shown in the figure (*i.e.*, $\text{supp}(q_{\mathbf{x}|\mathbf{y}}(\mathbf{x}|\mathbf{y}_{\text{OOD}})) \subset [-1, 1]^2$).

4.2. Super-resolution space generation

We qualitatively and quantitatively compare generated samples from diverse datasets. For some conditional inputs, FS-NCSR [38] generated SR images with artifacts, as in the second column of Figure 6. Our model with the modified RQ-spline layers does not generate any erroneous image samples as illustrated in the fourth column of Figure 6, even though the conditional inputs were severe OOD (EUrban100 4×, whose average OOD score was about 2.15 times larger than DIV2K 4× validation set).

For fair evaluation in detecting occasional errors, we use the average of the minimum and standard deviation of LR-

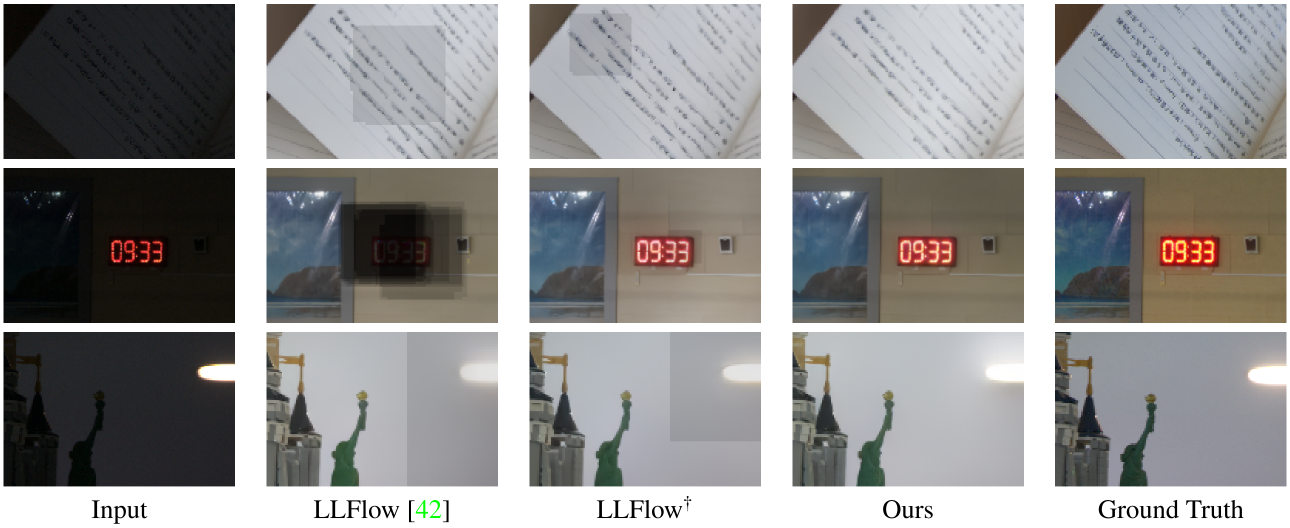


Figure 7: Qualitative comparison of coupling transformation in low-light image enhancement. The top two rows and the bottom row are samples from the LOL [43] and VE-LOL [23] datasets, respectively. The † sign denotes that the lower bound of the scale parameter is 0.1.

Train → Test	LOL → LOL				LOL → VE-LOL			
Model	%`Inf` ↓	PSNR↑	SSIM↑	LPIPS↓	%`Inf` ↓	PSNR↑	SSIM↑	LPIPS↓
LLFlow [42]	20	20.51	0.897	0.110	22	26.60	**0.919**	0.067
LLFlow†	20	19.66	0.894	0.121	13	23.70	0.904	0.080
Ours	**0**	**21.00**	**0.904**	**0.106**	**0**	**26.61**	**0.919**	**0.066**

Table 2: Quantitative comparison. The † sign denotes that the lower bound of the scale parameter is 0.1. '%`Inf`' refers to the percentage of conditional inputs that generate at least one `Inf` pixel out of 10 randomly generated latent codes, each with $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \tau^2)$. The temperature of the latent code (*i.e.*, τ) is 1 for both LOL [43] and VE-LOL [23] datasets.

PSNR among 10 samples, as LR-PSNR was one of the official evaluation metrics in the 2021, 2022 NTIRE challenges [27, 28]. Another metric %`Inf`, which was used in [4], refers to the percentage of conditional inputs that generate at least one `Inf` pixel. As shown in Table 1, FS-NCSR† (*i.e.*, FS-NCSR with $0.1 < s_1 \leq 1.1$ as discussed in Section 3.2.2) also suppressed `Inf` pixels. Still, erroneous images were sampled for the same conditional inputs, as shown in the third column of Figure 6. In contrast, our method was completely free of errors and showed the best results.

4.3. Low-light image enhancement

We qualitatively compare the mean of 10 generated outputs of LLFlow [42] in Figure 7. It is shown in the second column of Figure 7 that erroneous images (*e.g.* black regions with `Inf` pixels around the clock) are generated through affine coupling layers while images generated through our method do not present artifacts, as in the fourth column of Figure 7.

We also quantitatively compare the results in Table 2. The first and second rows of Table 2 show that the affine coupling transformation is prone to generating erroneous images whereas the modified RQ-spline coupling transformation is robust to OOD samples. Even with the scale parameter of the affine transformation adjusted to exceed 0.1, the black regions are still shown as in the third column of Figure 7. See the supplementary material for details and more various erroneous images sampled from LLFlow.

5. Discussion

Artifact type We could observe two types of artifacts. One shows random primary colors, while the other shows only black. To find out why those two types of artifacts appear, we extracted feature maps from the middle of the network (FS-NCSR [38]) when they co-occurred. Figure 8 shows the absolute values (log scale) of feature maps for a sample with both types of artifacts. In the first feature map, which is the closest to the latent variable $\mathbf{z}$ among the five, it can be seen that the absolute value is large only in a very small area (zoomed). The bright pixels, which have large absolute values, gradually spread out for the rest of

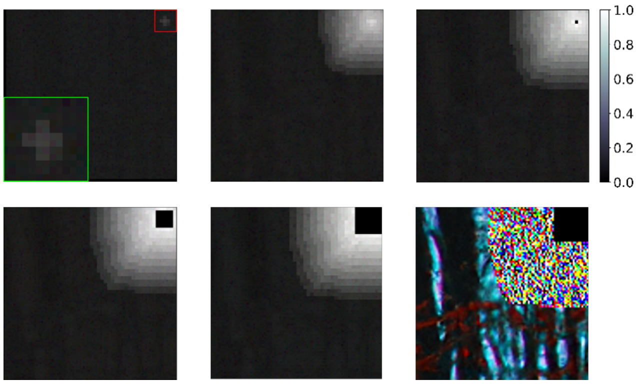


Figure 8: Visualization of feature map with exploding inverse (log scale). The white pixels from the top right of the first image gradually spread out and eventually form a black region with `Inf` pixels.

the feature maps and eventually form a black region with `Inf` values. Finally, both types of artifacts appear in the output, as in the last of Figure 8. This also explains why artifacts occur even without `Inf` pixels (see FS-NCSR† in Figure 6 and Table 1). One may wonder why the exploding inverse is gradually spreading, even though NFs do not use inter-pixel operations such as 3×3 convolution or max-pooling. One reason is that NFs have the equivalent effect of using inter-pixel operations, employing both inter-channel and pixel shuffle operations. The other reason is that the NN used in (4), (5) also employs inter-pixel operations.

Unconditional NFs The work of [4] performed the operation $f_\theta^{-1}(f_\theta(\mathbf{x}))$ for $\mathbf{x} \not\sim p_{\mathbf{x}}$ (Proposition 1 in [4]) to intentionally induce exploding inverses for the experiments on unconditional NFs. We believe that our proposed solution will work effectively in this case since the direct cause for unstable output is the same: the affine coupling layer.

Limitation Although our modified RQ-spline coupling transformation has an analytic inverse, it still imposes numerical overhead compared to the affine coupling transformation (about 2× training time). As we mentioned in Section 3.4, there may exist a computationally efficient method to ensure robustness. Measuring OOD scores is challenging but there is room for improvement for the accuracy of the Mahalanobis distance-based OOD score.

6. Conclusion

We addressed the issue of erroneous image samples in conditional NFs for inverse problems by revealing exploding inverse in affine coupling transformations and investigating OOD conditional inputs using the Mahalanobis distance. Then, we proposed the remarks to avoid exploding inverse in coupling transformations and suggested the modified RQ-spline coupling layer following the remarks for 2D toy, super-resolution space generation and low-light image enhancement, suppressing severe artifacts.

Acknowledgements

This work was supported in part by the National Research Foundation of Korea(NRF) grants funded by the Korea government(MSIT) (NRF-2022R1A4A1030579), Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education(NRF-2017R1D1A1B05035810) and Creative-Pioneering Researchers Program through Seoul National University. Also, the authors acknowledged the financial supports from BK21 FOUR program of the Education and Research Program for Future ICT Pioneers, Seoul National University. We also thank the authors of FS-NCSR [38], especially Kang-wook Kim, for insightful discussion on super-resolution space generation problems.

References

[1] Abdelrahman Abdelhamed, Marcus A. Brubaker, and Michael S. Brown. Noise flow: Noise modeling with conditional normalizing flows. In *ICCV*, 2019. 1

[2] Eirikur Agustsson and Radu Timofte. Ntire 2017 challenge on single image super-resolution: Dataset and study. In *CVPRW*, pages 126–135, 2017. 7

[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. 1, 3

[4] Jens Behrmann, Paul Vicol, Kuan-Chieh Wang, Roger Grosse, and Joern-Henrik Jacobsen. Understanding and mitigating exploding inverses in invertible neural networks. In *AISTATS*, 2021. 2, 3, 4, 8, 9

[5] Laurent Dinh, David Krueger, and Yoshua Bengio. NICE: non-linear independent components estimation. In *ICLR (Workshop)*, 2015. 2, 6

[6] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real NVP. In *ICLR (Poster)*, 2017. 1, 2, 3

[7] Hadi Mohaghegh Dolatabadi, Sarah Erfani, and Christopher Leckie. Invertible generative modeling using linear rational splines. In *AISTATS*, pages 4236–4246, 2020. 3, 6

[8] Conor Durkan, Artur Bekasov, Iain Murray, and George Papamakarios. Neural spline flows. *NeurIPS*, 32, 2019. 2, 3, 6

[9] Conor Durkan, Artur Bekasovs, Iain Murray, and Georgios Papamakarios. Cubic-spline flows. In *ICML Workshop on Invertible Neural Nets and Normalizing Flows*, 2019. 3, 6

[10] Jinzheng He, Zhou Zhao, Yi Ren, Jinglin Liu, Baoxing Huai, and Nicholas Jing Yuan. Flow-based unconstrained lip to speech generation. In *AAAI*, pages 843–851, 2022. 3

[11] Jonathan Ho, Xi Chen, Aravind Srinivas, Yan Duan, and Pieter Abbeel. Flow++: Improving flow-based generative

models with variational dequantization and architecture design. In *ICML*, pages 2722–2730, 2019. 3

[12] Jia-Bin Huang, Abhishek Singh, and Narendra Ahuja. Single image super-resolution from transformed self-exemplars. In *CVPR*, 2015. 5

[13] Younghyun Jo, Sejong Yang, and Seon Joo Kim. Srflow-da: Super-resolution using normalizing flow with deep convolutional block. In *CVPR*, pages 364–372, 2021. 1, 3

[14] Younggeun Kim and Donghee Son. Noise conditional flow model for learning the super-resolution space. In *CVPRW*, pages 424–432, 2021. 1, 3

[15] Durk P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. *NeurIPS*, 31, 2018. 2, 3

[16] Durk P Kingma, Tim Salimans, Rafal Jozefowicz, Xi Chen, Ilya Sutskever, and Max Welling. Improved variational inference with inverse autoregressive flow. *NIPS*, 29, 2016. 2

[17] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *ICLR*, 2014. 3

[18] Polina Kirichenko, Pavel Izmailov, and Andrew G Wilson. Why normalizing flows fail to detect out-of-distribution data. *NeurIPS*, 33, 2020. 5

[19] Jonas Köhler, Andreas Krämer, and Frank Noé. Smooth normalizing flows. *NeurIPS*, 34:2796–2809, 2021. 3

[20] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 2

[21] Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. *NeurIPS*, 31, 2018. 2, 5

[22] Jingyun Liang, Andreas Lugmayr, Kai Zhang, Martin Danelljan, Luc Van Gool, and Radu Timofte. Hierarchical conditional flow: A unified framework for image super-resolution and image rescaling. In *CVPR*, pages 4076–4085, 2021. 1, 3

[23] Jiaying Liu, Xu Dejia, Wenhan Yang, Minhao Fan, and Haofeng Huang. Benchmarking low-light image enhancement and beyond. *IJCV*, 129:1153–1184, 2021. 8

[24] You Lu and Bert Huang. Structured output learning with conditional generative flows. In *AAAI*, pages 5005–5012, 2020. 1, 3

[25] Alice Lucas, Michael Iliadis, Rafael Molina, and Aggelos K. Katsaggelos. Using deep neural networks for inverse problems in imaging: Beyond analytical methods. *IEEE Signal Processing Magazine*, 35(1):20–36, 2018. 1

[26] Andreas Lugmayr, Martin Danelljan, Luc Van Gool, and Radu Timofte. Srflow: Learning the super-resolution space with normalizing flow. In *ECCV*, pages 715–732, 2020. 1, 3, 5

[27] Andreas Lugmayr, Martin Danelljan, and Radu Timofte. Ntire 2021 learning the super-resolution space challenge. In *CVPRW*, pages 596–612, 2021. 1, 8

[28] Andreas Lugmayr, Martin Danelljan, Radu Timofte, Kangwook Kim, Younggeun Kim, Jae-young Lee, Zechao Li, Jinshan Pan, Dongseok Shim, Ki-Ung Song, et al. Ntire 2022 challenge on learning the super-resolution space. In *CVPRW*, pages 786–797, 2022. 1, 8

[29] Andreas Lugmayr, Martin Danelljan, Fisher Yu, Luc Van Gool, and Radu Timofte. Normalizing flow as a flexible fidelity objective for photo-realistic super-resolution. In *WACV*, pages 1756–1765, 2022. 1, 3

[30] Thomas Müller, Brian McWilliams, Fabrice Rousselle, Markus Gross, and Jan Novák. Neural importance sampling. *ACM Transactions on Graphics (TOG)*, 38(5):1–19, 2019. 3, 6

[31] Gregory Ongie, Ajil Jalal, Christopher A. Metzler, Richard G. Baraniuk, Alexandros G. Dimakis, and Rebecca Willett. Deep learning techniques for inverse problems in imaging. *IEEE Journal on Selected Areas in Information Theory*, 1(1):39–56, 2020. 1

[32] George Papamakarios, Eric T Nalisnick, Danilo Jimenez Rezende, Shakir Mohamed, and Balaji Lakshminarayanan. Normalizing flows for probabilistic modeling and inference. *Journal of Machine Learning Research*, 22(57):1–64, 2021. 2

[33] George Papamakarios, Theo Pavlakou, and Iain Murray. Masked autoregressive flow for density estimation. *NIPS*, 30, 2017. 2

[34] Ryan Prenger, Rafael Valle, and Bryan Catanzaro. Waveglow: A flow-based generative network for speech synthesis. In *IEEE ICASSP*, pages 3617–3621, 2019. 3

[35] Albert Pumarola, Stefan Popov, Francesc Moreno-Noguer, and Vittorio Ferrari. C-flow: Conditional generative flow models for images and 3d point clouds. In *CVPR*, 2020. 1

[36] Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *ICML*, pages 1530–1538, 2015. 2

[37] Danilo Jimenez Rezende, George Papamakarios, Sébastien Racaniere, Michael Albergo, Gurtej Kanwar, Phiala Shanahan, and Kyle Cranmer. Normalizing flows on tori and spheres. In *ICML*, pages 8083–8092, 2020. 3, 6

[38] Ki-Ung Song, Dongseok Shim, Kang-wook Kim, Jae-young Lee, and Younggeun Kim. Fs-ncsr: Increasing diversity of the super-resolution space via frequency separation and noise-conditioned normalizing flow. In *CVPRW*, June 2022. 1, 3, 7, 8, 9

[39] Rhea Sanjay Sukthanker, Zhiwu Huang, Suryansh Kumar, Radu Timofte, and Luc Van Gool. Generative flows with invertible attentions. In *CVPR*, pages 11234–11243, 2022. 3

[40] 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 *CVPRW*, pages 114–125, 2017. 7

[41] Haolin Wang, Jiawei Zhang, Ming Liu, Xiaohe Wu, and Wangmeng Zuo. Learning diverse tone styles for image retouching. *arXiv preprint arXiv:2207.05430*, 2022. 1, 3

[42] Yufei Wang, Renjie Wan, Wenhan Yang, Haoliang Li, Lap-Pui Chau, and Alex C Kot. Low-light image enhancement with normalizing flow. In *AAAI*, pages 2604–2612, 2022. 1, 3, 5, 8

[43] Chen Wei, Wenjing Wang, Wenhan Yang, and Jiaying Liu. Deep retinex decomposition for low-light enhancement. In *BMVC*, 2018. 8

[44] Hao Wu, Jonas Köhler, and Frank Noé. Stochastic normalizing flows. *NeurIPS*, 33:5933–5944, 2020. 3

[45] Yiqiang Wu, Dapeng Tao, Yibing Zhan, and Chenyang Zhang. Bin-flow: Bidirectional normalizing flow for robust image dehazing. *IEEE Transactions on Image Processing*, 2022. 1, 3

[46] Tong Xiao, Tian Xia, Yi Yang, Chang Huang, and Xiaogang Wang. Learning from massive noisy labeled data for image classification. In *CVPR*, 2019. 2