Supplementary Material of
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

S1. Experimental details and more results

S1.1. OOD score

S1.1.1 Super-resolution space generation

We generated 1470 patches from the DIV2K [1] 4× validation dataset and ranked them based on their OOD score (s_{OOD}) using the conditioning network $g_{\boldsymbol{\theta}}$ of the fully-trained FS-NCSR [8]. For the super-resolution space generation, we concatenate the output of RRDB [10] blocks 1, 8, 15, 22, instead of directly using the output of $g_{\boldsymbol{\theta}}$ for better feature representation of the conditioning encoder, which is trained on the DF2K [9] training set 4×. Then, we collect patches of size 160×160 and compute the OOD score (i.e. s_{OOD}) for each patch. The method of concatenating blocks of RRDB stems from the work of SRFlow [6], where they concatenate equally spaced RRDB blocks 1, 8, 15, 22, and 23 to obtain the final output of the conditioning encoder. This corresponds to Section 3.3 and Figure 4 of the main paper.

Pixel error To verify the presented OOD score, we computed the pixel error probability for each patch by generating 10 samples from each image of the DIV2K validation set. For each sample, we calculated the number of erroneous pixels, with the minimum and maximum error threshold set as -0.5 and 1.5, respectively. This is because the output of the neural network should be within the range of $[0, 1]$ before clamping. However, it is important to note that this pixel error is only a necessary condition for the exploding inverse, and not a necessary and sufficient condition. This is because a value of 0 or 1 obtained after clamping may be intended. Table S1 shows the percentage of conditional inputs that generate at least one pixel whose value is outside the range of $[-0.5, 1.5]$. In the case of in-distribution, only 7% of the conditional inputs generated at least one pixel error. However, in the case of OOD, 90% of the conditional inputs generated pixel errors. It is worth noting that these values are significantly higher than the percentages of conditional inputs that generate an erroneous image in human eyes.

Train set	Test set	Distribution	% PixelErr↓
DF2K 4×	DIV2K 4×	in-distribution	7%
	EUrban100 4×	OOD	90%

Table S1: The percentage of conditional inputs that generate at least one error pixel (*i.e.*, pixel value is out of $[-0.5, 1.5]$) out of 10 randomly generated latent codes, each with $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \tau^2)$, where $\tau = 0.9$.

Enhanced Urban100 To investigate the case of severe OOD, we made modifications to the Urban100 dataset [2]. The original Urban100 dataset has an average OOD score of $\mathbb{E}[s_{\text{OOD}}] = 331.01$, which is only slightly larger than that of the training set ($\mathbb{E}[s_{\text{OOD}}] = 236.78$). To generate a severe OOD dataset, we enhanced each image of the Urban100 dataset by strengthening the high frequency components using a convolution kernel $\mathbf{H}$, where

$$\mathbf{H} = \frac{1}{3}\begin{bmatrix} -1 & -4 & -1 \\ -4 & 26 & -4 \\ -1 & -4 & -1 \end{bmatrix}. \tag{S1}$$

This operation enhanced the OOD score to $\mathbb{E}[s_{\text{OOD}}] = 511.35$, which is much larger than that of the training set.

S1.1.2 Low-light image enhancement

We also calculate the OOD score for the low-light image enhancement on the LOL [12] testset. The second row of Figure 1 in the main paper shows an erroneous sample generated from the patch with the highest OOD score among the 90 patches. Similar to the task of super-resolution space generation, we concatenate the output of RRDB blocks 1, 3, 5, 7 as the output of $g_{\boldsymbol{\theta}}$, the conditioning network fully trained on the LOL [12] training set. Then, we collect a total of 90 patches, each of size 100×100. We rank the OOD score based on the mahalanobis score of each patch.

[*] Corresponding author

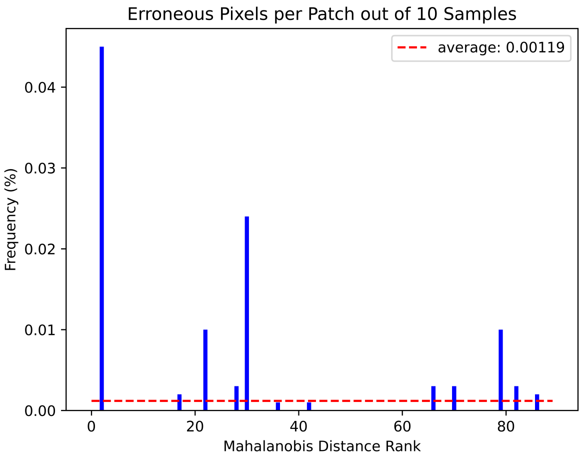

Figure S1: Pixel error probability for the patches ranked according to their OOD score (s_{OOD}). The average of 90 patches is marked as a dashed horizontal line.

In Figure S1, we show the pixel error probability of the LOL dataset ranked according to the OOD score of each patch. In all cases, ours showed the best results.

S1.2. 2D toy experiment

Training data The training data is obtained by the following equation:

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} \frac{\sqrt{3}}{4} & -\frac{1}{10} \\ \frac{1}{4} & -\frac{\sqrt{3}}{10} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}, \tag{S2}$$

where $u_1, u_2 \sim \mathcal{U}(-1, 1)$ (*i.e.*, uniform distribution on $(-1, 1)$). We generated 100,000 samples using (S2).

Network architecture NN in the coupling layers was a fully connected network composed of four hidden layers with a width of 64. For the modified RQ-spline coupling layer, the output of NN is four-dimensional (*i.e.*, $\mathbf{h}_2 \in \mathbb{R}^4$). The four components of the output are bias (*i.e.* t), input coordinate of the learnable knot, output coordinate of the learnable knot, and slope of the learnable knot (*i.e.*, derivative of the RQ-spline transformation at the learnable knot). The input coordinate of the learnable knot is normalized (via sigmoid) to be in $(B_1 + \epsilon, B_2 - \epsilon)$, and the output coordinate of the learnable knot is normalized (via sigmoid) to be in $(B_1 + t + \epsilon, B_2 + t - \epsilon)$. We set the slope of the learnable knot in (ϵ, ∞), via exponential function. We used $(B_1, B_2) = (-0.5, 0.5)$ and $\epsilon = 0.001$. $\mathbf{z}$ was assumed to be the standard Gaussian.

Training We trained the network using Adam optimizer [3], with $(\beta_1, \beta_2) = (0.9, 0.999)$, learning rate 5×10^{-4}, batch size 1,000, for 8,000 iterations.

S1.3. Super-resolution space generation

Training data For the DIV2K [1] validation set, the training set is a combination of DIV2K 1-800 and Flickr2K [9] 1-2,650 (total 3,450, and the union of DIV2K and Flickr2K is referred as DF2K), and the test set is DIV2K 801-900 and EUrban100. We used 160×160 RGB patches as HR images. We randomly cropped the original images to generate 160×160 RGB patches. We used bicubic kernel to generate the conditional inputs. We applied 90° rotations and horizontal flips randomly for data augmentation.

Network architecture In the case of substituting the affine coupling layer of FS-NCSR [8] with the modified RQ-spline coupling layer, the output of NN was set in the same manner as in Section S1.2. NN was a CNN, which is the same as FS-NCSR. The other structures were also exactly same as FS-NCSR. We set $\tau = 0.9$.

Training We trained the network using Adam optimizer [3], with $(\beta_1, \beta_2) = (0.9, 0.999)$, initial learning rate 2×10^{-4}. The learning rate is halved when 50%, 75%, 90%, 95% of the total number of iterations are trained. For DIV2K 4× dataset, batch size was 16, and the number of iterations was 180,000. For DIV2K 8× dataset, batch size was 12, and the number of iterations was 200,000. For fast training on 8× datasets, we replaced the invertible 1×1 convolutions with fixed random unitary matrices. This technique is proposed by Lugmayr *et al.* [7], and has the effect of reducing training time while maintaining performance. We train the networks on a NVIDIA GeForce GTX 3090 GPU.

Additional results We provide additional examples of artifacts in Figure S2.

S1.3.1 Additional experiment on another dataset

For the CelebA [5] validation set, CelebA 1-182,340 served as the training set, while CelebA 182,341-202,600 (total 20,260) was the validation set. In the case of CelebA 8× dataset, batch size was 12, and the number of iterations was 100,000. We provide examples of artifacts in Figure S3. Table S2 shows the quantitative results of 8x super-resolution space generation on CelebA datasets. %Inf demonstrates that our method effectively suppressed exploding inverses. However, compared to Table 1 in the text, FS-NCSR has a relatively small %Inf, as the CelebA dataset has very few OOD conditional inputs overall. Since the occurrence of exploding inverses was infrequent in this dataset, there was no significant difference in $\overline{\min}$ and $\overline{\sigma}$. Nevertheless, our

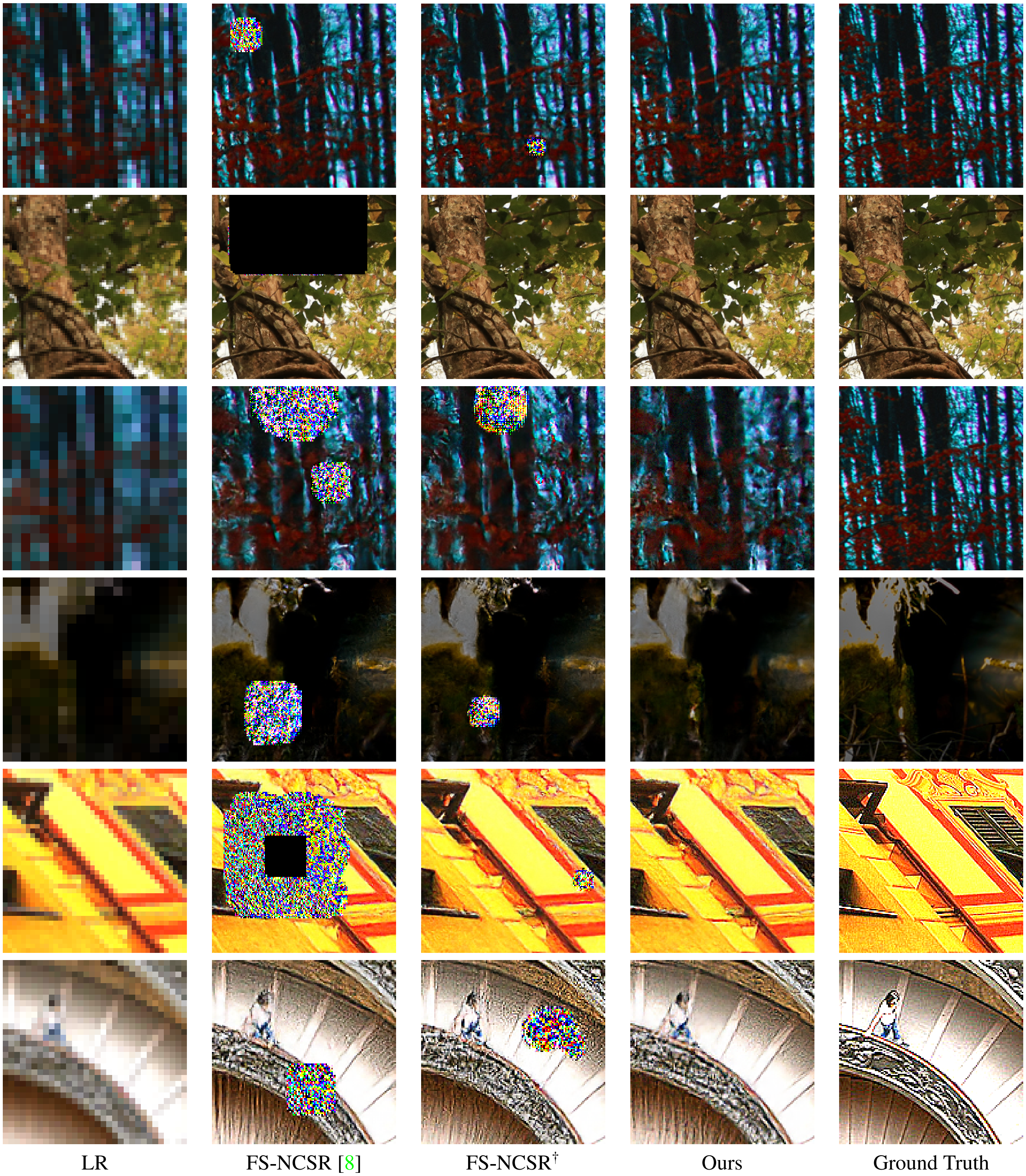

Figure S2: Qualitative comparison of coupling transformation in super-resolution space generation. The 1st-2nd, 3rd-4th, and 5th-6th rows show the samples from DIV2K [1] 4×, DIV2K 8×, and EUrban100 4×. The † sign denotes that the lower bound of the scale parameter is 0.1.

method, with the exception of $\overline{\min}$, exhibited the most favorable results.

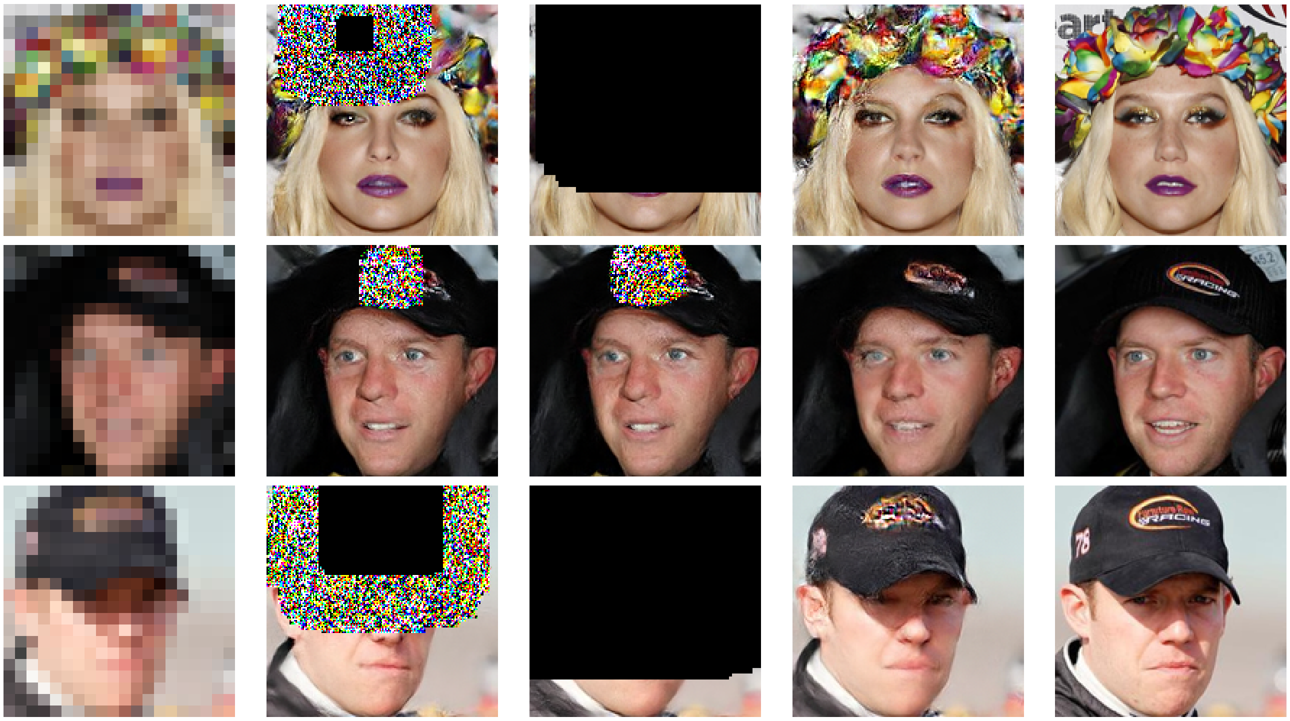

Figure S3: Qualitative comparison of coupling transformation in super-resolution space generation, on CelebA [5] 8×. The † sign denotes that the lower bound of the scale parameter is 0.1.

Train → Test	CelebA 8× → CelebA 8×			
Model	%`Inf` ↓	$\overline{\min}$ ↑	$\overline{\sigma}$ ↓	% PixelErr ↓
FS-NCSR [8]	0.074	50.73	0.223	2.78
FS-NCSR†	0.020	**51.09**	0.214	1.62
Ours	**0**	50.63	**0.199**	**0.48**

Table S2: Quantitative comparison on CelebA 8× dataset. The † sign denotes that the lower bound of the scale parameter is 0.1. '%`Inf`' and '% PixelErr' refer to the percentage of conditional inputs that generate at least one `Inf` pixel / pixel whose value is out of $[-0.5, 1.5]$ out of 10 randomly generated latent codes, each with $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \tau^2)$, respectively. $\overline{\min}$ and $\overline{\sigma}$ refer the average of the minimum and standard deviation of LR-PSNR, respectively.

S1.4. Low-light enhancement

Sampling method LLFlow [11] suggested two sampling schemes to solve the low-light image enhancement problem. One is to fix the latent code $\mathbf{z}$ to $\mathbf{0}$ (*i.e.*, $\hat{\mathbf{x}} = f_{\boldsymbol{\theta}}^{-1}(\mathbf{0}; \mathbf{y})$). The other is to select a batch of $\mathbf{z}$ from the Gaussian distribution, and then calculate the mean (*i.e.*, $\hat{\mathbf{x}} = \mathbb{E}_{\mathbf{z}\sim\mathcal{N}(0,\tau^2)}[f_{\boldsymbol{\theta}}^{-1}(\mathbf{z}; \mathbf{y})]$). Although LLFlow proposed both schemes, the authors only experimented with the first scheme. Here, we show experimental results that the second scheme generates erroneous images, while our solution does not.

Training data We follow the training method of LLFlow [11] where we perform two evaluations: one on the LOL [12] validation set (trained on the LOL training set) and one on the VE-LOL [4] captured validation set (trained on the LOL training set).

Training We trained the network using the same hyperparameters as the authors of LLFlow [11]. For both experiments, the batch size was 16 for the baseline model and 8 for our model. The number of iterations was 40,000 for the baseline model and 80,000 for our model. We train the networks on a NVIDIA Titan RTX GPU.

Additional results We provide additional examples of artifacts in Figures S4 and S5.

S2. Additional Resources

We used the source code of Zhang *et al.* [13] to zoom images.

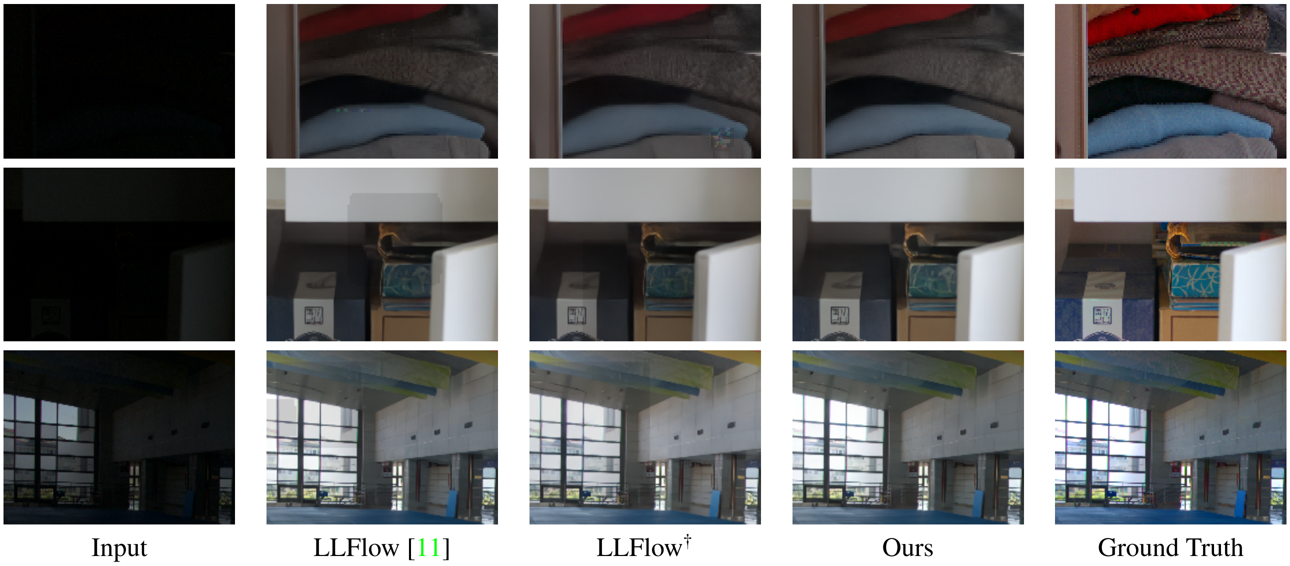

Figure S4: Qualitative comparison of coupling transformation in low-light image enhancement on the LOL [12] dataset. The † sign denotes that the lower bound of the scale parameter is 0.1.

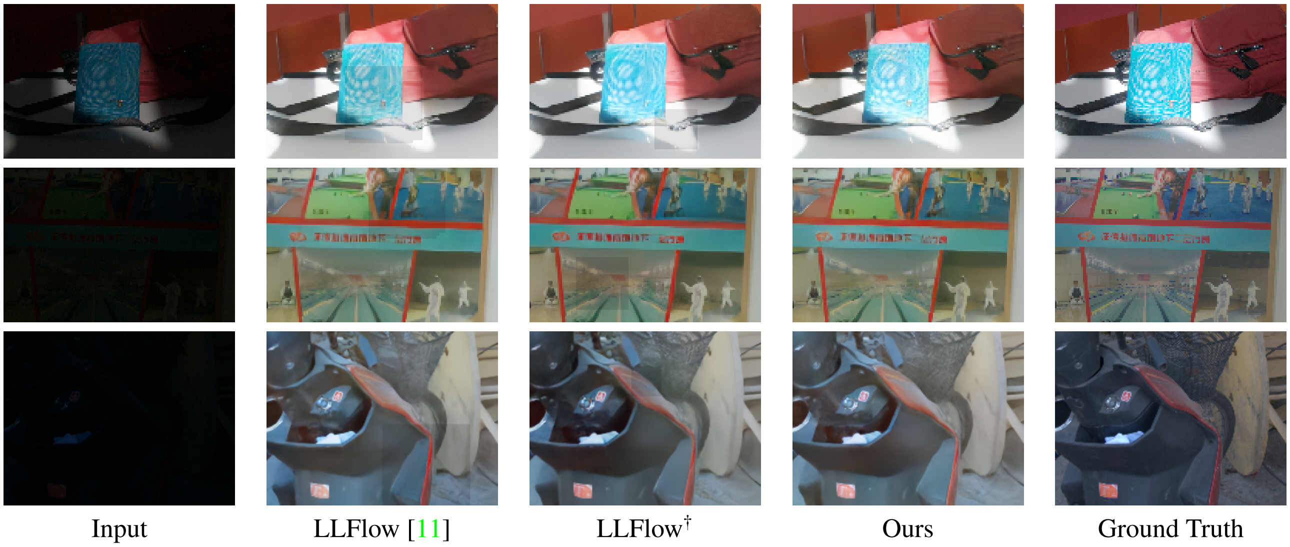

Figure S5: Qualitative comparison of coupling transformation in low-light image enhancement on the VE-LOL [4] dataset. The † sign denotes that the lower bound of the scale parameter is 0.1.

References

[1] Eirikur Agustsson and Radu Timofte. Ntire 2017 challenge on single image super-resolution: Dataset and study. In *CVPRW*, pages 126–135, 2017. 1, 2, 3

[2] Jia-Bin Huang, Abhishek Singh, and Narendra Ahuja. Single image super-resolution from transformed self-exemplars. In *CVPR*, 2015. 1

[3] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 2

[4] Jiaying Liu, Xu Dejia, Wenhan Yang, Minhao Fan, and Haofeng Huang. Benchmarking low-light image enhancement and beyond. *IJCV*, 129:1153–1184, 2021. 4, 5

[5] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *ICCV*, pages 3730–3738, 2015. 2, 4

[6] 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

[7] 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. 2

[8] 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, 2, 3, 4

[9] 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. 1, 2

[10] Xintao Wang, Ke Yu, Shixiang Wu, Jinjin Gu, Yihao Liu, Chao Dong, Yu Qiao, and Chen Change Loy. Esrgan: Enhanced super-resolution generative adversarial networks. In *ECCVW*, September 2018. 1

[11] 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. 4, 5

[12] Chen Wei, Wenjing Wang, Wenhan Yang, and Jiaying Liu. Deep retinex decomposition for low-light enhancement. In *BMVC*, 2018. 1, 4, 5

[13] Kai Zhang, Xiaoyu Zhou, Hongzhi Zhang, and Wangmeng Zuo. Revisiting single image super-resolution under internet environment: blur kernels and reconstruction algorithms. In *Pacific Rim Conference on Multimedia*, pages 677–687. Springer, 2015. 4