Towards General Low-Light Raw Noise Synthesis and Modeling (Supplementary Material)

Feng Zhang[1] Bin Xu[2] Zhiqiang Li[1,2] Xinran Liu[1] Qingbo Lu[2] Changxin Gao[1] Nong Sang[1†]
[1]National Key Laboratory of Multispectral Information Intelligent Processing Technology,
School of Artificial Intelligence and Automation, Huazhong University of Science and Technology
[2]DJI Technology Co., Ltd
{fengzhangaia, lxryx, cgao, nsang}@hust.edu.cn, {mila.xu, cristopher.li, qingbo.lu}@dji.com

Abstract

This supplementary material provides detailed network architecture of our proposed noise generator, the configuration of the discriminator architecture, the architectural design of the Fourier transformer block (FTB), an extra noise model analysis, and more visual comparison results.

1. Detailed Architecture of Noise Generator

To synthesize signal-independent noise, we employ a standard 2D residual U-Net-style structure [4]. It comprises an encoder block for feature compression and a decoder block for pixel-wise noise prediction. The architecture also employs skip connections that concatenate output features from encoder layers to corresponding decoder layers, as depicted in Fig. 1. Notably, the U-Net network takes in and outputs images with four channels. The network architecture comprises four encoder and four decoder blocks. Each encoder block consists of a 3×3 convolutional layer, followed by a SeLU activation layer and a stride-2 convolutional layer aimed at down-sampling the input images. On the other hand, each decoder block includes a stride-2 transpose-convolutional layer for up-sampling the feature maps, followed by a 3×3 convolutional layer and a SeLU activation layer to recover fine-grained details.

2. Fourier Transformer Discriminator

2.1. Detailed Architecture Configurations

In this study, we present a detailed account of the architecture configurations of the proposed Fourier Transformer Discriminator (FTD), as documented in Table. 1. The term ”Layer Flatten” is utilized herein to describe the process of patch splitting and subsequent linear transformation. The term ”FT-Blocks” represents the proposed Fourier transformer block. The term ”V-Block” represents the vanilla transformer block. At every stage, the resulting feature map is concatenated with a sequence of the input image at different scales. In the conclusive stage, a CLS token is introduced, and a vanilla transformer block is employed to establish correspondence between the extracted representation and the introduced CLS token. Ultimately, only the CLS token is selected by the classification Head to predict the authenticity- real/fake.

Table 1. Architecture configuration of our proposed Fourier transformer discriminator. $\star$ indicates that this layer is optional. See the main text for more explanation.

Fourier transformer discriminator		
Layer	Input Shape	Output Shape
Linear Flatten	$64 \times 64 \times 4$	$(16 \times 16) \times 64$
FT-Block1-1	$(16 \times 16) \times 64$	$(16 \times 16) \times 64$
FT-Block1-2$\star$	$(16 \times 16) \times 64$	$(16 \times 16) \times 64$
FT-Block1-3$\star$	$(16 \times 16) \times 64$	$(16 \times 16) \times 64$
AvgPooling	$(16 \times 16) \times 64$	$(8 \times 8) \times 64$
Concatenate	$(8 \times 8) \times 64$	$(8 \times 8) \times 128$
FT-Block2-1	$(8 \times 8) \times 128$	$(8 \times 8) \times 128$
FT-Block2-2$\star$	$(8 \times 8) \times 128$	$(8 \times 8) \times 128$
FT-Block2-3$\star$	$(8 \times 8) \times 128$	$(8 \times 8) \times 128$
AvgPooling	$(8 \times 8) \times 128$	$(4 \times 4) \times 128$
Concatenate	$(4 \times 4) \times 128$	$(4 \times 4) \times 256$
FT-Block3-1	$(4 \times 4) \times 256$	$(4 \times 4) \times 256$
FT-Block3-2$\star$	$(4 \times 4) \times 256$	$(4 \times 4) \times 256$
FT-Block3-3$\star$	$(4 \times 4) \times 256$	$(4 \times 4) \times 256$
CLS Token	$(4 \times 4) \times 256$	$(4 \times 4 + 1) \times 256$
V-Block	$(4 \times 4 + 1) \times 256$	$(4 \times 4 + 1) \times 256$
CLS Head	$(4 \times 4 + 1) \times 256$	1

2.2. Fourier Transformer Block

Inspired by fast Fourier Convolution (FFC) [3], we propose a transformer-based module, namely Fourier transformer block (FTB), which is based on a pixel-wise fast

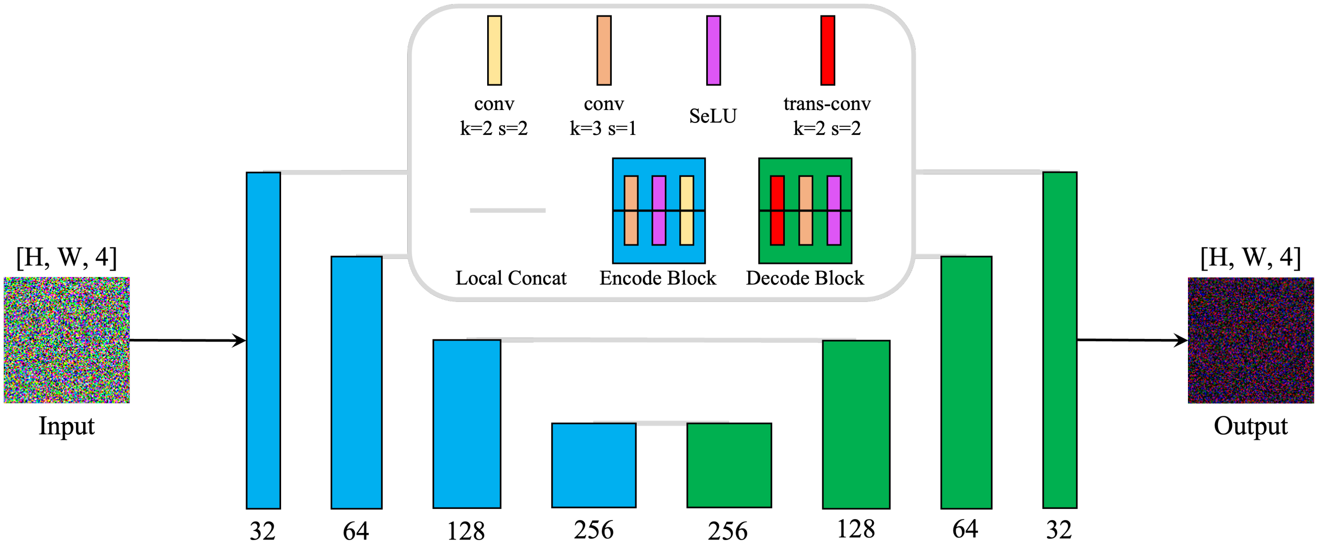


Figure 1. Network architecture of our proposed noise generator.

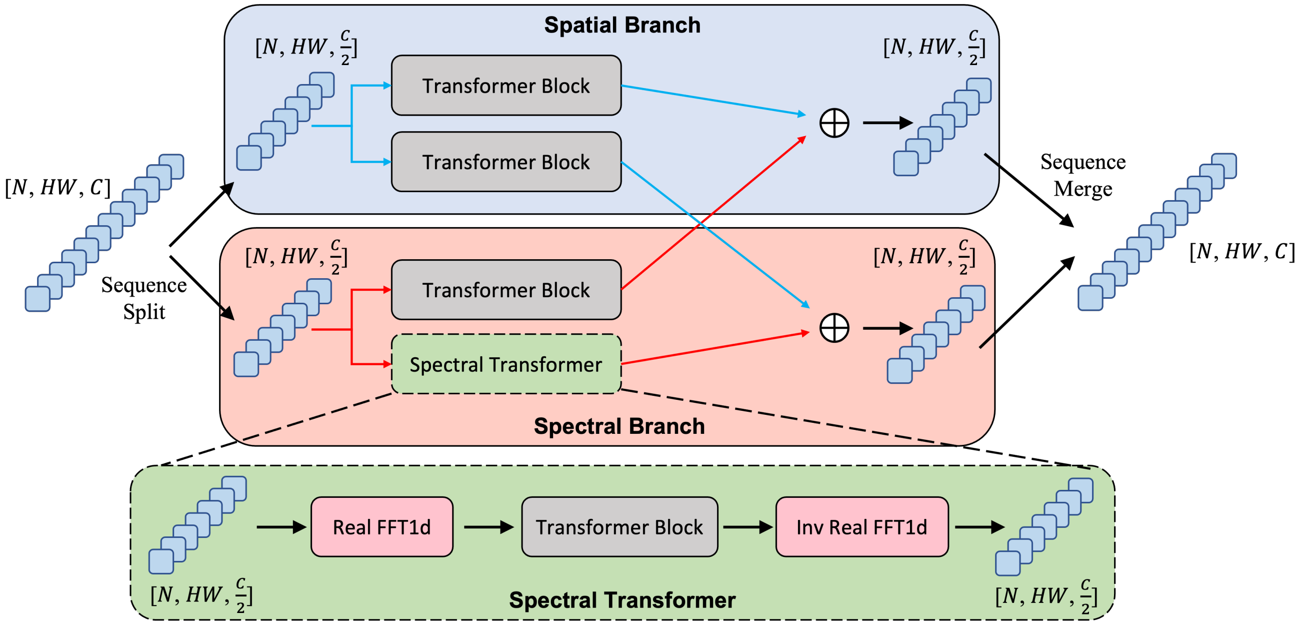


Figure 2. Architecture design of Fourier transformer block (FTB). " $\oplus$ " denotes element-wise sum. The proposed FTB splits the input sequence into two parallel branches: i) spatial branch uses vanilla transformer blocks, and ii) spectral branch uses a vanilla transformer block and a spectral transformer block. See the main text for more explanation.

Fourier transform (FFT) [1]. The architecture of our proposed FTB is shown in Fig 2. Fundamentally, the FTB comprises two interconnected branches: a spatial branch that performs standard operations on half of input feature sequences and a spectral branch that operates on the other half of the sequences in the spectral domain. Each branch is capable of capturing complementary information with varying receptive fields, and information exchange between these branches is internal to the FTB.

We adopt two vanilla transformer blocks to extract the feature sequences of the image in the spatial domain, and one of the extracted feature sequences is exchanged to the spectral domain. In the spectral domain, one vanilla transformer block is employed to retain the spatial feature information of the input sequence and exchange it into the spatial domain, while the other spectral transformer block is adopted to extract the noise features in the spectral domain. As the spectral transformer block proposed in FFC is designed to enlarge the receptive field, modifications are made to it in two ways to enhance noise feature extraction. First, we remove the local Fourier unit to reduce the computational complexity. Next, we modify the Fourier unit block. The original implementation of the spectral transformer block uses a *Real FFT2d* and a convolution layer. We replace this architecture with *Real FFT1d* and a vanilla transformer block. *Real FFT1d* can be applied only to real valued signals, and *inverse real FFT1d* ensures that the output is real valued. *Real FFT1d* uses only half of the spectrum compared to the FFT. Specifically, our proposed spectral transformer block makes the following steps:

a) applies Real FFT1d to an input sequence

$$\textit{Real FFT1d} = \mathbb{R}^{HW \times C} \rightarrow \mathbb{C}^{\frac{HW}{2} \times C}; \tag{1}$$

and concatenates real and imaginary parts

$$\textit{Complex To Real} = \mathbb{C}^{\frac{HW}{2} \times C} \rightarrow \mathbb{R}^{\frac{HW}{2} \times 2C}; \tag{2}$$

b) applies a vanilla transformer block in the frequency domain

$$\textit{Transformer Block} = \mathbb{R}^{\frac{HW}{2} \times 2C} \rightarrow \mathbb{R}^{\frac{HW}{2} \times 2C}; \tag{3}$$

c) applies inverse transform to recover a spatial structure

$$\textit{Real To Complex} = \mathbb{R}^{\frac{HW}{2} \times 2C} \rightarrow \mathbb{C}^{\frac{HW}{2} \times C}; \tag{4}$$

$$\textit{Inverse Real FFT1d} = \mathbb{C}^{\frac{HW}{2} \times C} \rightarrow \mathbb{R}^{HW \times C}. \tag{5}$$

Ultimately, the outputs of the spatial and spectral branches are merged, forming a comprehensive representation. The proposed Fourier transformer blocks (FTBs) are fully differentiable and can be effortlessly incorporated instead of conventional transformer blocks. By operating in both the spatial and spectral domains, the FTBs enable the discriminator to consider both global and local noise distribution information, which is a critical aspect for effectively discriminating high-noise-level raw images.

3. Extra Noise Model Analysis

In order to accurately reflect the stochastic properties of the noise, it is desired that the noise model provide distinct noise samples during each forward pass. Fig. 3 exhibits the noise images produced by five separate forward passes of our noise model, illustrating the generation of different noise samples in each pass.

To further demonstrate the effectiveness and generalizability of our noise model, we utilize a single learned model to generate noise samples at varying ISO levels. The visualization results can be found in Fig. 4. The results demonstrate the effectiveness and power generation capability of our noise model.

4. Extra Visualization of Denoising Results

In this section, we present more visual comparison results on different datasets (SID, ELD, LRD) as shown in Fig. 5, 6, 7. In comparison to state-of-the-art methods, our method provides competitive denoising results with visual pleasurable.

References

[1] E Oran Brigham and RE Morrow. The fast fourier transform. *IEEE spectrum*, 4(12):63–70, 1967. 2

[2] Chen Chen, Qifeng Chen, Jia Xu, and Vladlen Koltun. Learning to see in the dark. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3291–3300, 2018. 5

[3] Lu Chi, Borui Jiang, and Yadong Mu. Fast fourier convolution. *Advances in Neural Information Processing Systems*, 33:4479–4488, 2020. 1

[4] Kristina Monakhova, Stephan R Richter, Laura Waller, and Vladlen Koltun. Dancing under the stars: video denoising in starlight. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16241–16251, 2022. 1

[5] Kaixuan Wei, Ying Fu, Jiaolong Yang, and Hua Huang. A physics-based noise formation model for extreme low-light raw denoising. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2758–2767, 2020. 5

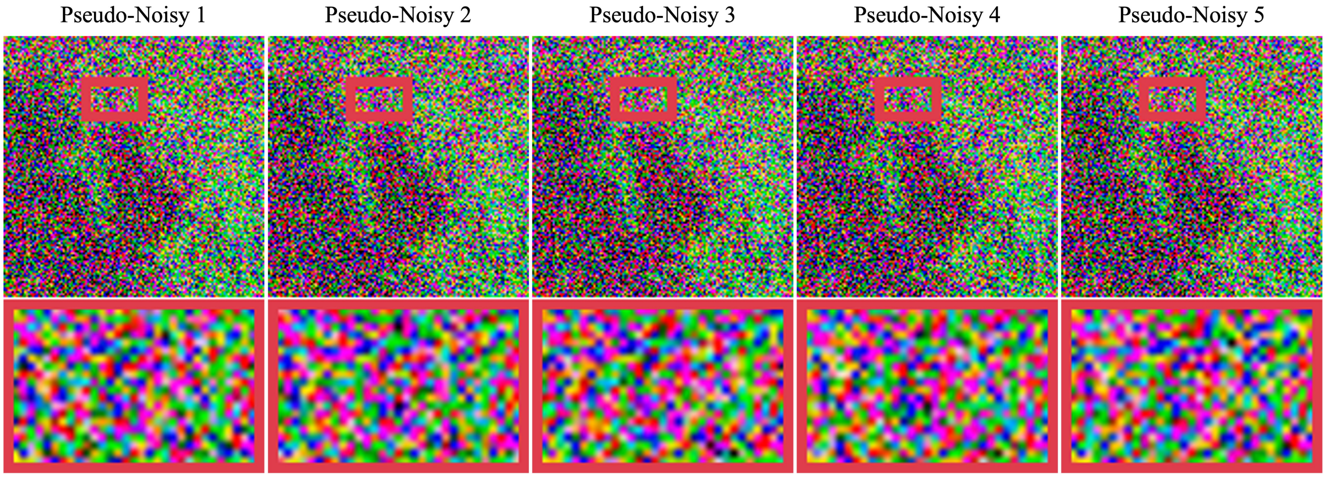


Figure 3. Noise samples generated by our method in five different forward passes. While the distribution of each sample remains constant, distinct discrepancies can be noticed among the generated noise pixels.

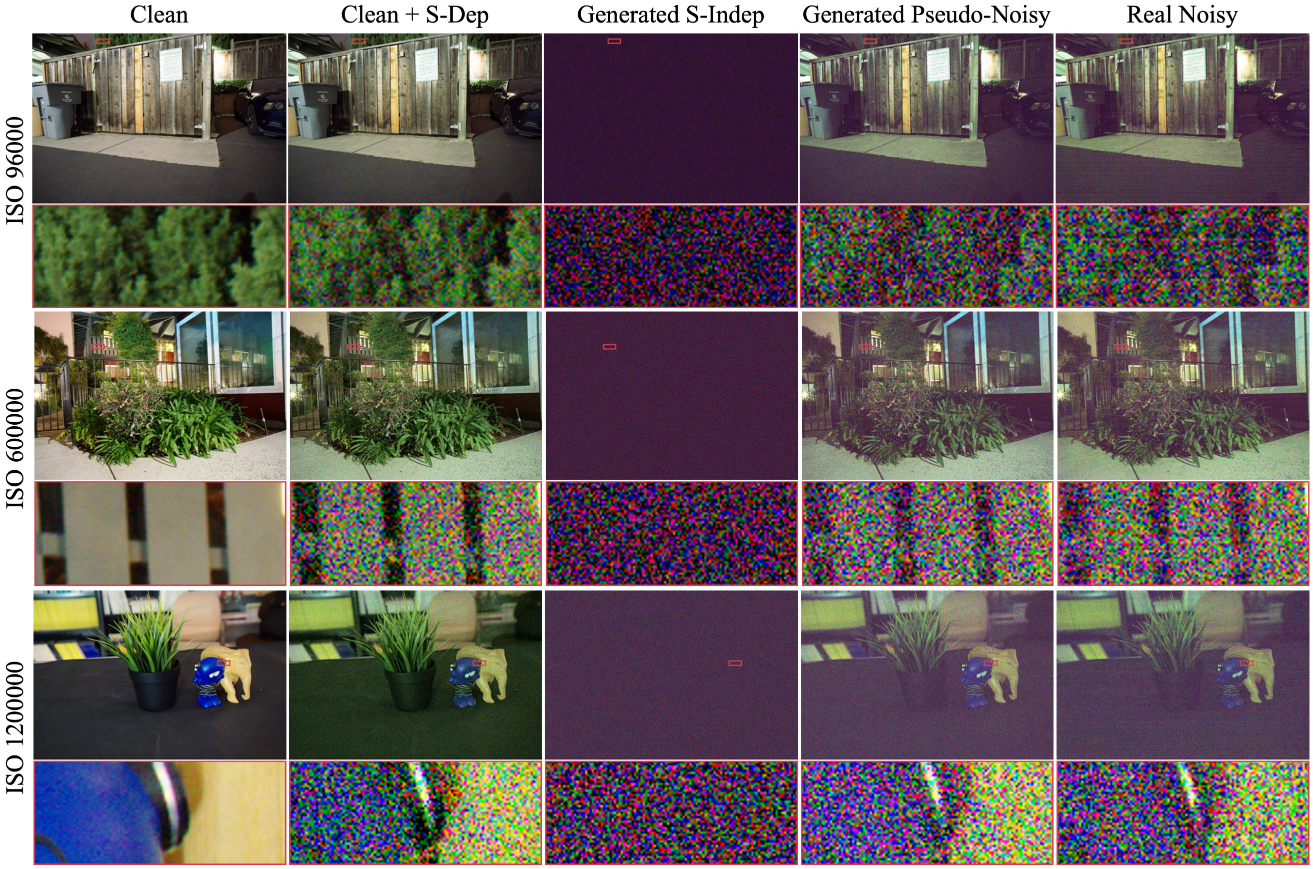
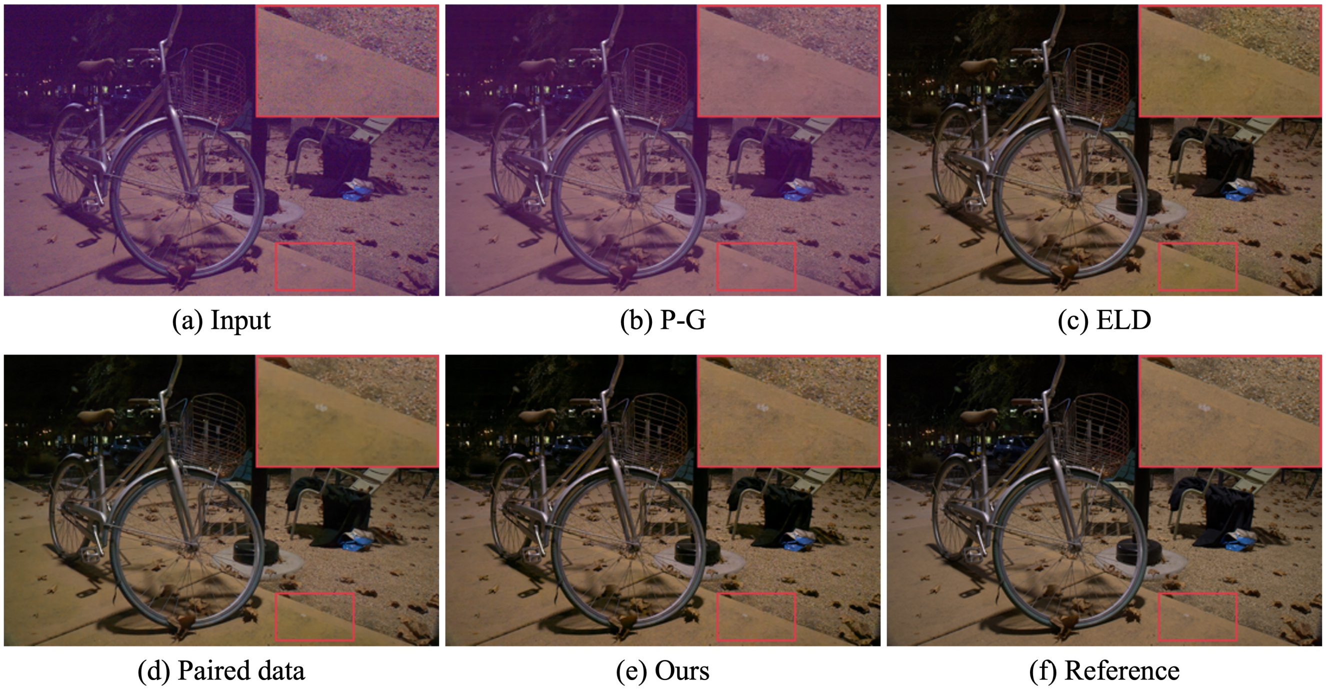


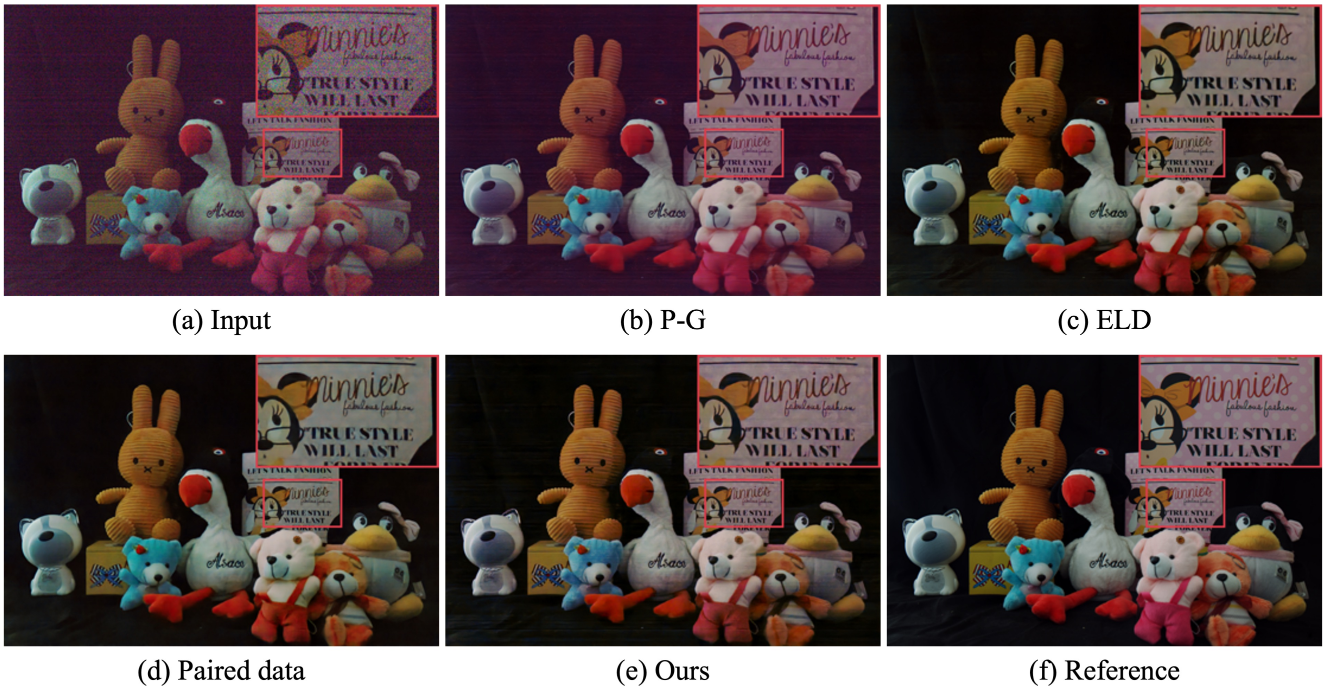
Figure 4. A single learned model generates noise at different ISO levels. First column: clean raw image. Second column: sampled image contaminated by signal-dependent noise. Third column: synthesized signal-independent noise. Fourth column: synthesized pseudo-noisy raw image (second column $\bigoplus$ third column). Fifth column: real low-light noisy raw image.



(a) Input (b) P-G (c) ELD

(d) Paired data (e) Ours (f) Reference

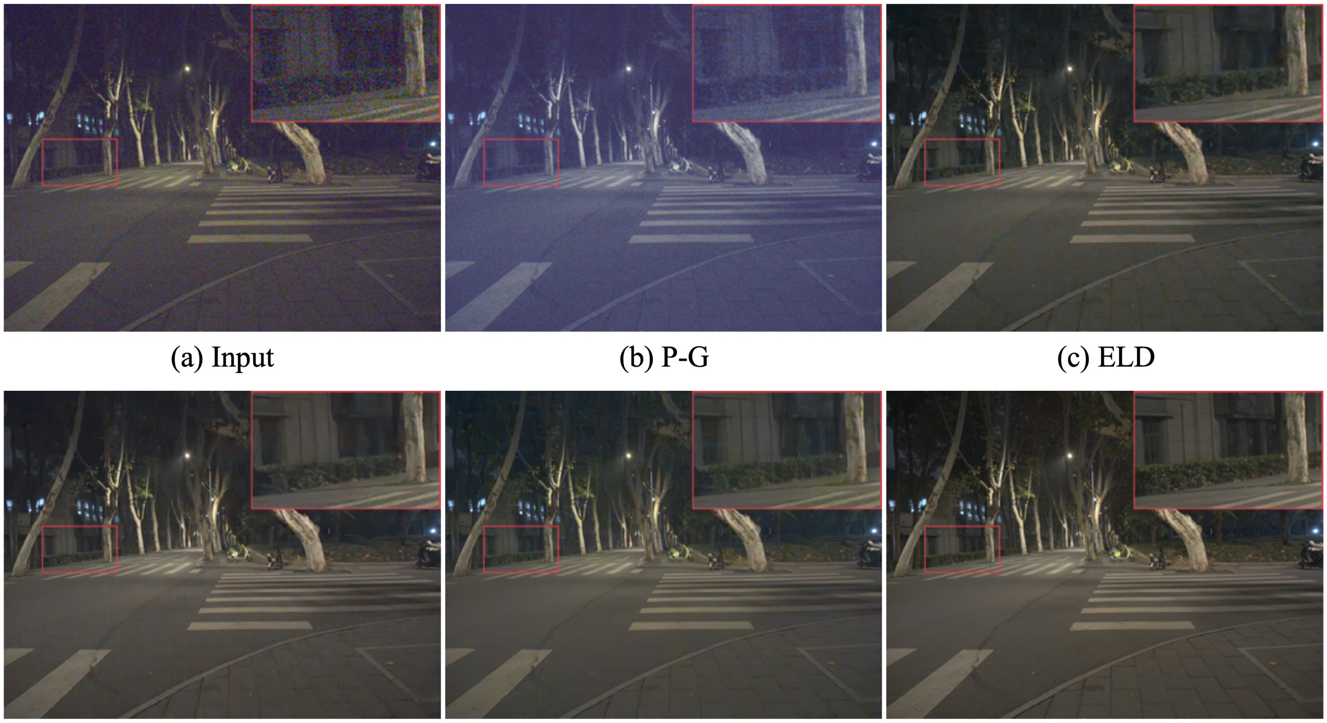
Figure 5. Raw image denoising comparison with state-of-the-art methods on low-light noisy raw images from the SID dataset [2]. Best viewed in color and by zooming in.



(a) Input (b) P-G (c) ELD

(d) Paired data (e) Ours (f) Reference

Figure 6. Raw image denoising comparison with state-of-the-art methods on low-light noisy raw images from the ELD dataset [5]. Best viewed in color and by zooming in.



(a) Input
(b) P-G
(c) ELD
(d) Paired data
(e) Ours
(f) Reference

Figure 7. Raw image denoising comparison with state-of-the-art methods on low-light noisy raw images from our LRD dataset. Best viewed in color and by zooming in.