

sRGB Real Noise Synthesizing with Neighboring Correlation-Aware Noise Model

Zixuan Fu^{1*}, Lanqing Guo^{1*}, Bihan Wen^{1†}

¹Nanyang Technological University, Singapore

{zixuan.fu, lanqing001, bihan.wen}@ntu.edu.sg

Abstract

Modeling and synthesizing real noise in the standard RGB (sRGB) domain is challenging due to the complicated noise distribution. While most of the deep noise generators proposed to synthesize sRGB real noise using an end-to-end trained model, the lack of explicit noise modeling degrades the quality of their synthesized noise. In this work, we propose to model the real noise as not only dependent on the underlying clean image pixel intensity, but also highly correlated to its neighboring noise realization within the local region. Correspondingly, we propose a novel noise synthesizing framework by explicitly learning its neighboring correlation on top of the signal dependency. With the proposed noise model, our framework greatly bridges the distribution gap between synthetic noise and real noise. We show that our generated “real” sRGB noisy images can be used for training supervised deep denoisers, thus to improve their real denoising results with a large margin, comparing to the popular classic denoisers or the deep denoisers that are trained on other sRGB noise generators. The code will be available at <https://github.com/xuan611/sRGB-Real-Noise-Synthesizing>.

1. Introduction

Real image denoising is one of the most challenging tasks in low-level vision. Deep denoisers that are trained using synthetic noise, *e.g.*, Additive White Gaussian Noise (AWGN), perform poorly on real photography [3, 15], which motivates more realistic noise models, *e.g.*, [1, 5, 14–16]. In general, there are two approaches towards real noise modeling, *i.e.*, modeling in the raw-RGB and standard RGB (sRGB) domains. Popular modeling methods including the physical-based [25, 28] and data-driven methods [1, 6] exploit sophisticated noise models in the raw-RGB domain, which demonstrated promising perfor-

mance as noise in raw-RGB is largely simplified comparing to noise in sRGB [20, 22]. However, raw-RGB images are not usually utilized by common users due to their large sizes. In contrast, most commercial cameras generate sRGB images by default, which are more popular in practice. Unfortunately, the noise generation methods in the raw-RGB domain cannot be directly applied to sRGB images, as the real noise distribution in sRGB is more complicated than raw-RGB noise, caused by the in-camera signal processing (ISP) pipeline [22].

Recent works [5, 15] proposed to generate noise on raw-RGB images and convert them into sRGB images by the ISP pipeline including demosaicing, white balancing, gamma correction, *etc.* While these methods synthesized realistic noise, the requirement of raw-RGB images as well as manually defined ISP pipelines limits their applications. An alternative solution for sRGB real noise modeling is to train the generative models with sRGB noisy-clean images and directly synthesize real noise on sRGB images [16, 17, 20, 26]. However, these models synthesize noise without explicitly modeling the characteristics of sRGB real noise, resulting in degradation of the quality of the synthesized noise.

In this paper, we propose a novel real noise generation network, based on Neighboring Correlation-Aware noise model, dubbed as NeCA, to directly synthesize real noise in the sRGB domain. The proposed real noise synthesis assumes that the sRGB real noise is not only signal-dependent, *i.e.*, noise level partially depends on its underlying clean pixel, but also highly correlated with its neighboring noise realization. Such a real noise model greatly bridges the gap between the synthetic and real noise in sRGB. Furthermore, the synthesized “real” images by the proposed NeCA can be used for training supervised deep denoisers, thus tackling the real image denoising challenges, subjective to only a few real training data. The trained deep denoiser using our synthetic noisy images achieves state-of-the-art denoising performance, compared to the popular classic denoisers as well as deep denoisers that are trained on synthetic pairs from other noise models. To sum up, our main contributions can be concluded as follows:

*Co-first authors contributed equally.

†Corresponding author: Bihan Wen.

This work was supported in part by the MOE AcRF Tier 1 (RG61/22) and Start-Up Grant.

- We introduce a neighboring correlation-aware noise model for sRGB real noise synthesis by explicitly modeling the neighboring correlation of real noise, to bridge the gap between the synthetic and real noise distribution in sRGB.
- Our proposed framework shows a well-generalized ability, which is still capable to improve the real image denoising performance even with limited training data.
- With the synthetic image pairs generated by NeCA, the trained denoisers achieve state-of-the-art denoising performance compared with the deep denoisers trained with other real noise models.

2. Related Work

2.1. Raw-RGB Image Noise Synthesis

Modeling real noise in raw-RGB is challenging as it cannot be simply assumed as Additive White Gaussian Noise (AWGN). Typically, raw-RGB noise models can be classified into two categories: physical-based models and learning-based models. One of the most commonly used physical-based models is the heteroscedastic Gaussian noise [10], which posits noise value, located at pixel i , is dependent on its underlying clean pixel intensity:

$$\mathbf{n}_i \sim \mathcal{N}(0, \sigma_s^2 \cdot \mathbf{x}_i + \sigma_c^2), \quad (1)$$

where \mathbf{n} and \mathbf{x} are noise and clean image in the raw-RGB domain, while σ_s and σ_c denote the noise variance term for signal-dependent and signal-independent components. Such a noise model is also known as the noise level function (NLF) as it describes the relationship between the pixel-wise noise level and image intensity. To better model the camera sensor noise, recent works [25, 28] have proposed that real noise is a sophisticated combination of shot noise, read noise and row noise, *etc.*

Compared to statistical modeling of noise, learning-based models learn the real noise distribution with generative models such as the generative adversarial nets (GANs) [6] and normalization flows [1] from paired noisy-clean images. Although these methods perform well in raw-RGB, they cannot be directly applied to model sRGB real noise since their assumptions are based on the characteristics of raw-RGB noise. For instance, these noise generators synthesize raw-RGB noise from an initialized heteroscedastic Gaussian noise (as described in Equation (1)), which fails to provide an accurate representation of real noise in the sRGB domain [21, 22].

2.2. sRGB Image Noise Synthesis

The camera ISP pipeline, including demosaicing, tone mapping, white balancing, gamma mapping, *etc.*, makes

real noise in the sRGB domain to be more complicated than it is in the raw-RGB domain. To synthesize sRGB real noise, two approaches have been proposed: (1) synthesizing noisy samples in the raw-RGB domain and rendering them into sRGB images by applying the manually defined ISP pipeline [5, 15], and (2) directly synthesizing real noise in the sRGB domain [8, 16, 17, 20, 26].

In CBDNet [15], heteroscedastic Gaussian noise is added on raw-RGB clean images, and images are converted into sRGB using demosaicing and camera response functions. However, CBDNet requires raw-RGB images, which are not commonly used. To address this issue, unprocessing image (UPI) [5] proposes to de-render sRGB images into raw-RGB images using several predefined unprocessing pipelines. Similar procedures used in CBDNet are then applied to the unprocessed raw-RGB images to obtain their sRGB versions.

Despite CBDNet and UPI effectively synthesize sRGB real noise, they still require predefined ISP pipelines, which may not match real ones used in different camera sensors. Therefore, generating real noise directly in the sRGB domain with deep generative models [11, 19] is considered an alternative solution. GCBD [8] proposes a GAN-based model that learns noise distributions by training on noise patches that have been cropped from noisy images. However, the synthesized noise is signal-independent as it is generated from random noise. DANet [26] and GRDN [17] use conditional generative networks to synthesize signal-dependent noise, however, few experiments are conducted to demonstrate the effectiveness of the proposed noise generators. C2N [16] attempts to synthesize the real noise with unpaired clean-noisy images, but the generated noise contains artifacts and color-shift problems due to the unpaired training mode. Recently, Kousha et al. [20] propose a conditional flow-based model for sRGB image noise generation that takes clean images, camera types, and ISO levels as input. However, the denoiser, trained with synthetic data, improves marginally compared to the unpaired noise generation method C2N. Unlike previous attempts that model noise with an end-to-end generator, our proposed method explicitly decomposes signal dependency and neighboring correlation of real noise and learns them with separate networks.

3. Method

3.1. Neighboring Correlation-Aware Noise Model

In this section, we present our proposed noise model for sRGB real noise. We begin by introducing the basic noise model, which defines the signal dependency of pixel-wise noise level and its underlying clean pixels. We then discuss discrepancies between noise synthesized by the basic noise model and sRGB real noise and propose to bridge this gap

by explicitly modeling noise neighboring correlation on top of the signal dependency.

Basic Noise Model. Both raw-RGB and sRGB real noise are dependent on the image signal. In raw-RGB, the noise level can be approximated as a simple function of its underlying clean pixel intensity, *i.e.*, heteroscedastic Gaussian noise described in Equation (1). However sRGB real noise is more complex due to camera settings and signal transformations in the ISP pipeline [20–22]. To address this challenge, we propose a noise model that characterizes the signal dependency of sRGB real noise. Specifically, for an sRGB clean image $\mathbf{x} = (\mathbf{x}_1, \dots, \mathbf{x}_N)$ and its paired noisy version $\mathbf{y} = (\mathbf{y}_1, \dots, \mathbf{y}_N)$, we define noise level at pixel i as a function of the clean image patch $\Omega_{\mathbf{x}}$, centered at clean pixel \mathbf{x}_i , and camera ISO level γ :

$$\sigma_i = f(\Omega_{\mathbf{x}}, \gamma), \quad (2)$$

where $f(\cdot)$ represents the non-linear relationship of $\Omega_{\mathbf{x}}$, γ and the pixel-wise noise level $\sigma_i = (\sigma_{i,r}, \sigma_{i,g}, \sigma_{i,b})$ for three color channels. For the sake of clarity, we omit the location index i in the expression for the local region $\Omega_{\mathbf{x}}$. Then the distribution of noise \mathbf{v} at each pixel is modeled as a Gaussian distribution:

$$\mathbf{v}_{i,c} \sim \mathcal{N}(0, \sigma_{i,c}^2), \quad (3)$$

where c is the index of RGB channels. We further define the noise level map \mathbf{m} , which has the same size as the clean image and the value at pixel i refers to the noise level σ_i . Finally, we can simulate signal-dependent noise as follows:

$$\mathbf{v} = \epsilon \odot \mathbf{m}, \quad \epsilon_{i,c} \sim \mathcal{N}(0, 1^2). \quad (4)$$

Neighboring Correlation Noise Model. The noise synthesized by the basic noise model still exhibits discrepancies with real noise, as shown in Figure 1(b) and (d). We attribute this gap to the improper noise realization defined in Equation (4), where noise is sampled spatially independently from the basic noise model. Specifically, the most commonly used noise models, including the AWGN, heteroscedastic Gaussian noise, and our basic noise model, assume that the noise distribution is independent at each pixel, and the noise is sampled from the noise distribution without considering its neighboring synthesized noise. However, this noise realization method is inadequate to synthesize sRGB real noise as the noise value is assumed to be highly correlated with its neighboring noise values due to the influence of the ISP pipeline such as demosaicing, which introduces neighboring operations. We refer to this characteristic of noise as neighboring correlation and define a neighboring correlation operator $g(\cdot)$ that maps such the correlation onto the synthesized signal-dependent noise \mathbf{v} :

$$\mathbf{n}_i = g(\Omega_{\mathbf{v}}), \quad (5)$$

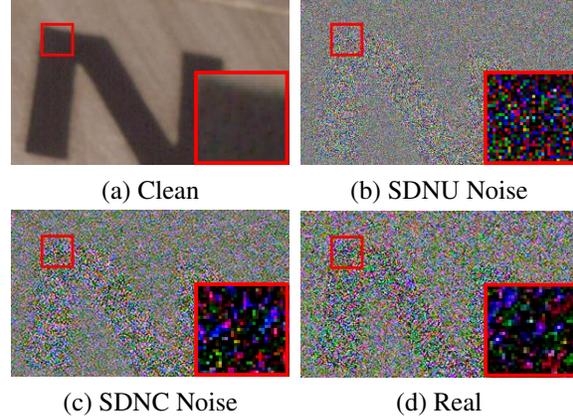


Figure 1. The visualization of modeling signal dependency and neighboring correlation of sRGB real noise. (a) Clean image. (b) Synthetic signal-dependent and neighboring uncorrelated (SDNU) noise. (c) Synthetic signal-dependent and neighboring correlated (SDNC) noise. (d) Real noise. We add a constant value to the noise maps for better visualizing the signal dependency.

where \mathbf{n} is the neighboring correlated noise and $\Omega_{\mathbf{v}}$ is the local patch of \mathbf{v} , centered at pixel i . By processing the neighboring uncorrelated noise \mathbf{v} with the neighboring correlation operator, which is learned by our proposed noise synthesizing framework in Section 3.2, the final generated noise performs similar characteristics to real noise, as demonstrated in Figure 1(c) and (d). For the purpose of clarity, we use SDNU noise to refer to the intermediate synthesized signal-dependent and neighboring uncorrelated noise \mathbf{v} , and SDNC noise to refer to the final generated signal-dependent and neighboring correlated noise \mathbf{n} . In the following sections, we will introduce the proposed noise synthesizing framework to explicitly learn the neighboring correlation and signal dependency of noise.

3.2. Noise Synthesizing Framework

Given paired sRGB real-world noisy and clean images (\mathbf{y}, \mathbf{x}) , where $\mathbf{y} = \mathbf{x} + \mathbf{n}$, our proposed framework aims to learn the neighboring correlation-aware noise model using paired data. Our proposed framework, as illustrated in Figure 2, comprises three networks: a gain estimation network (GENet), a noise-level prediction network (NPNet), and a neighboring correlation network (NCNet). GENet estimates the gain factor from a noisy image, which serves to amplify the synthesized noise, similar to the ISO level. NPNet synthesizes the SDNU noise by incorporating the estimated gain factor and the clean image as inputs. Finally, NCNet explicitly models the neighboring correlation of sRGB real noise and generates the SDNC noise.

Gain Estimation Network. The gain estimation network (GENet) is designed to estimate the gain factor from a noisy image \mathbf{y} , which serves as guidance to control the overall magnitude of the synthesized noise. The gain factor is de-

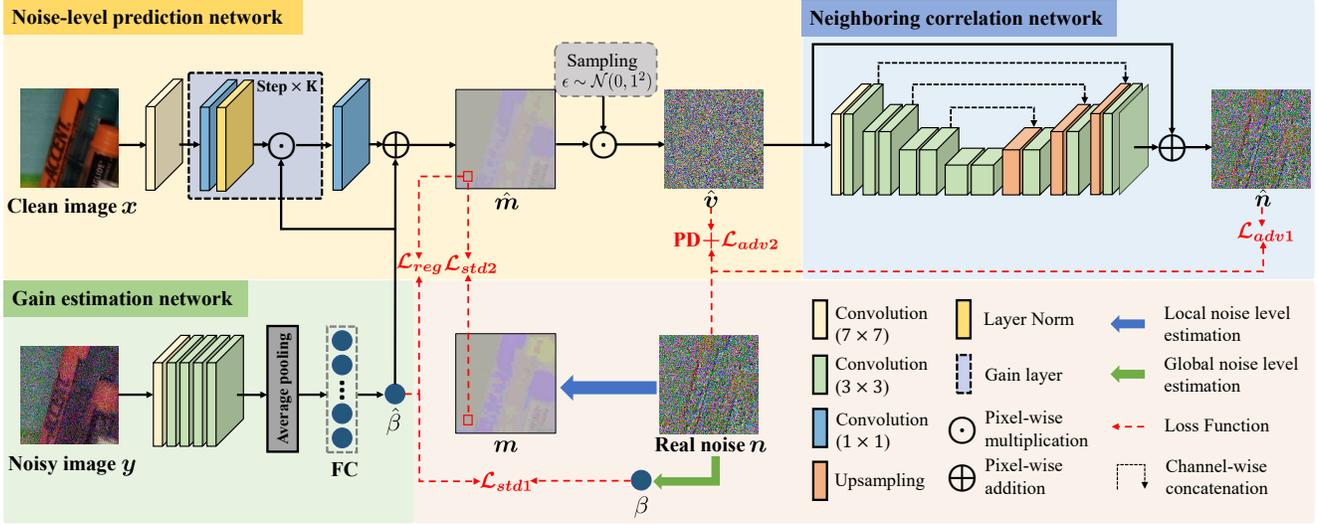


Figure 2. The proposed noise synthesizing framework. Our NeCA contains three networks including the gain estimation network (GENet), noise-level prediction network (NPNet), and neighboring correlation network (NCNet). **PD** denotes the Pixel-shuffle Down-sampling scheme introduced in [29]. Local noise level estimation and global noise level estimation operations are formulated in Equation (13) and (6). The details of the network architecture and PD scheme are described in the supplementary material.

defined as the global noise level of the noisy image, which is the standard deviation calculated by every noise value in its noise \mathbf{n} :

$$\beta = \sqrt{\frac{1}{N} \sum_{i,c} (\mathbf{n}_{i,c} - \bar{n})^2}, \quad (6)$$

where β is the defined global noise level of the noisy image \mathbf{y} , \bar{n} is the mean of the noise \mathbf{n} , and N is the total number of pixels in the noisy image. However, during testing, the calculated gain factor is unavailable. To solve this, we aim to estimate the gain factor from the noisy image using GENet:

$$\hat{\beta} = E(\mathbf{y}), \quad (7)$$

where E represents the GENet, and $\hat{\beta}$ is the estimated gain factor by GENet, which is expected to be as close as the global noise level of the noisy image. The main reason to use the gain factor estimated from the noisy image rather than the ISO level is driven by a crucial factor. ISO levels are typically saved in the metadata of images. The requirement of metadata will limit the application of our noise synthesizing framework.

Noise-level Prediction Network. The noise-level prediction network (NPNet) learns a parametric model for the noise distribution defined in Equation (3). To achieve this, NPNet predicts the pixel-wise noise level $\hat{\sigma}_i$ using the clean local patch Ω_x and estimated gain factor $\hat{\beta}$:

$$\hat{\sigma}_i = G_1(\Omega_x, \hat{\beta}), \quad (8)$$

where G_1 denotes the NPNet, which has three output channels to predict noise levels for each pixel. To effectively

incorporate the gain factor into the NPNet, we first apply the layer normalization [4] to the feature map of convolution and then multiply the normalized feature map by the gain factor. In practice, NPNet directly outputs the predicted noise level map $\hat{\mathbf{m}}$ by utilizing a clean image and gain factor:

$$\hat{\mathbf{m}} = G_1(\mathbf{x}, \hat{\beta}). \quad (9)$$

Once the noise level map $\hat{\mathbf{m}}$ is obtained, the SDNU noise $\hat{\mathbf{v}}$ can be synthesized by using the sampling trick defined in Equation (4).

Neighboring Correlation Network. The neighboring correlation network (NCNet) performs as the neighboring correlation operator, described in Equation (5). By taking the noise value and its neighboring noise realization as input, NCNet generates the SDNC noise $\hat{\mathbf{n}}$:

$$\hat{\mathbf{n}}_i = G_2(\Omega_{\hat{\mathbf{v}}}), \quad (10)$$

where $\Omega_{\hat{\mathbf{v}}}$ is the noise patch of $\hat{\mathbf{v}}$ located at pixel i and G_2 denotes the NCNet. The SDNC noise can be directly generated by taking the SDNU noise into the network:

$$\hat{\mathbf{n}} = G_2(\hat{\mathbf{v}}). \quad (11)$$

3.3. Loss Functions

To jointly train the proposed networks, five loss functions are introduced: (1) standard deviation losses \mathcal{L}_{std1} and \mathcal{L}_{std2} , (2) adversarial losses \mathcal{L}_{adv1} and \mathcal{L}_{adv2} , (3) the regularization loss \mathcal{L}_{reg} . The details of these loss functions will be introduced later.

Standard Deviation Loss. We introduce \mathcal{L}_{std1} to enforce the estimated gain factor $\hat{\beta}$ by GENet to be close to the

global noise level β of the noisy image, which is defined as follows:

$$\mathcal{L}_{std1} = \mathbb{E}_{\mathbf{y}} [(\hat{\beta} - \beta)^2], \quad (12)$$

where β and $\hat{\beta}$ are obtained by Equation (6) and (7).

The objective of NPNet is to predict the noise level map $\hat{\mathbf{m}}$ by taking the clean image and gain factor as input. However, since the groundtruth noise level map is not available, we propose to use a simple local noise level estimation method to approximate the noise level map \mathbf{m} from the noise, which is calculated as follows:

$$\mathbf{m}_{i,c} = \sqrt{\mathcal{MF}(\Omega_n^2) - \mathcal{MF}^2(\Omega_n)}, \quad (13)$$

where Ω_n denotes the 7×7 noise patch located at pixel i , channel c of noise map \mathbf{n} , and $\mathcal{MF}(\cdot)$ represents the mean filter. Then the \mathcal{L}_{std2} is defined as follows:

$$\mathcal{L}_{std2} = \mathbb{E}_{\mathbf{x}, \mathbf{y}} [||\hat{\mathbf{m}} - \mathbf{m}||_2^2]. \quad (14)$$

Adversarial Loss. In order to guarantee that the generated noise shares the similar distribution with real noise, we introduce two adversarial losses. Our first adversarial loss \mathcal{L}_{adv1} is imposed between the final synthetic SDNC noise and real noise ($\hat{\mathbf{n}}, \mathbf{n}$) to enforce the highly neighboring correlation in the generated noise, similar to that of the real noise. Our second adversarial loss \mathcal{L}_{adv2} is calculated by using Pixel-shuffle Down-sampling [29] versions of synthesized intermediate noise $\hat{\mathbf{v}}$ and real noise \mathbf{n} . Specifically, \mathcal{L}_{adv2} serves as a complementary loss for \mathcal{L}_{std2} because estimating the approximate noise level map using Equation (13) may not be reliable, as this method struggles to differentiate between noise originating from different intensities. However, directly calculating the adversarial loss between noise $\hat{\mathbf{v}}$ and \mathbf{n} is unreasonable since $\hat{\mathbf{v}}$ is neighboring uncorrelated. To address this problem, we utilize the Pixel-shuffle Down-sampling (PD) scheme proposed in [29] to obtain down-sampled versions ($(\hat{\mathbf{v}}) \downarrow_s, (\mathbf{n}) \downarrow_s$) of both synthetic noise $\hat{\mathbf{v}}$ and real noise \mathbf{n} . Here \downarrow_s denotes the PD operation with a stride of s (in this paper, s is set to 3). According to [29], the neighboring correlation in the PD real noise $(\mathbf{n}) \downarrow_s$ will be greatly attenuated. This allows us to calculate the adversarial loss between the two down-sampled versions. We utilize WGAN-GP [13] to compute adversarial losses, while \mathcal{L}_{adv1} is defined as follows:

$$\mathcal{L}_{adv1} = -\mathbb{E}_{\hat{\mathbf{n}}} [D_1(\hat{\mathbf{n}})], \quad (15)$$

where D_1 is the discriminator for NCNet, which scores the realism of synthesized noise. Similarly, \mathcal{L}_{adv2} is computed as follows:

$$\mathcal{L}_{adv2} = -\mathbb{E}_{(\hat{\mathbf{v}}) \downarrow_s} [D_2((\hat{\mathbf{v}}) \downarrow_s)], \quad (16)$$

where D_2 is the discriminator for NPNet. More detail about the PD scheme and the discriminator losses will be discussed in the supplementary material.

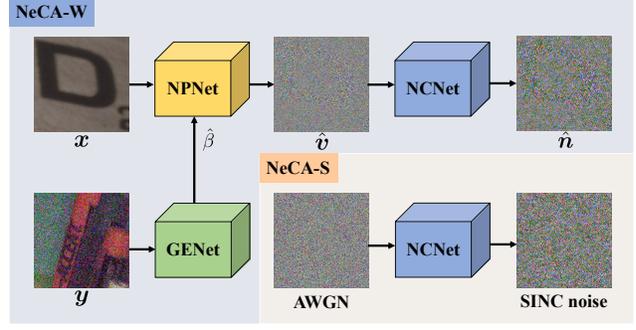


Figure 3. The designed two inference versions. NeCA-W utilizes the whole framework to synthesize SDNC noise. NeCA-S only adopts NCNet to synthesize signal-independent neighboring correlated (SINC) noise by taking the AWGN as input.

Regularization Loss. Besides the losses mentioned above, a regularization loss \mathcal{L}_{reg} is utilized to stabilize training. It is imposed between the estimated gain factor $\hat{\beta}$ and the predicted noise level map $\hat{\mathbf{m}}$:

$$\mathcal{L}_{reg} = \mathbb{E}_{\mathbf{x}, \mathbf{y}} [||\mathbf{w}||_2^2], \quad (17)$$

where $\mathbf{w}_{i,c} = \hat{\mathbf{m}}_{i,c} - \hat{\beta}$.

Finally, the full loss functions of the framework are described as follows:

$$\mathcal{L} = \mathcal{L}_{reg} + \lambda_1 \mathcal{L}_{adv1} + \lambda_2 \mathcal{L}_{adv2} + \lambda_3 \mathcal{L}_{std1} + \lambda_4 \mathcal{L}_{std2}, \quad (18)$$

where $\lambda_1, \lambda_2, \lambda_3$ and λ_4 are hyperparameters to balance the importance between different losses.

3.4. Inference Stage

We provide two inference versions to generate noise, as illustrated in Figure 3: (1) NeCA-W applies the entire framework to synthesize "real" noise. It first estimates the gain factor from an arbitrary noisy image and synthesizes noise by conditioning on a clean image and the estimated gain factor. (2) NeCA-S is the simplified version of NeCA-W which uses only NCNet for inference. In this method, AWGN is synthesized and then NCNet maps it with the neighboring correlation. We refer to this synthetic noise as signal-independent neighboring correlated (SINC) noise. Notably, NeCA-S still enhances the performance of deep denoiser on real noise, even though the denoiser is trained using the synthesized SINC noise. In the meantime, this inference version only requires minimal data to train the NCNet, which we will demonstrate in our experiments.

4. Experiments

4.1. Experimental Setup

To assess the effectiveness of our proposed noise synthesizing framework, we conduct experiments in two parts:

First, we assess the quality of the generated noise. Second, we examine the performance of NeCA on the downstream image denoising task. The details of the experiments will be discussed in the following subsections.

Dataset. We evaluate our NeCA on the medium version of Smartphone Image Denoising Dataset (SIDDD) [2], which comprises 320 noisy-clean image pairs captured by five different smartphone cameras, including Samsung Galaxy S6 Edge (S6), iPhone 7 (IP), Google Pixel (GP), Motorola Nexus 6 (N6), and LG G4 (G4). These images are collected in ten different scenes with varying ISO levels and lighting conditions. The SIDDD provides both raw-RGB and sRGB images, with the sRGB version obtained by rendering the captured raw-RGB images through the manually defined ISP pipeline provided in [2]. In our experiments, we use the sRGB version to evaluate the proposed method.

Metrics. We evaluate the performance of NeCA using three metrics: Discrete Kullback-Leibler (KL) divergence, Signal-to-Noise Ratio (PSNR) and Structural Similarity (SSIM) [24]. The KL divergence is used to measure the similarity of histograms between real noise and generated noise. The histogram range is set from -0.1 to 0.1 with 64 intervals. The PSNR and SSIM are used to evaluate the performance of deep denoisers. A higher PSNR and SSIM reflect better denoising performance, while a smaller KL divergence represents better noise synthesizing quality.

Implementation Details. All the networks are optimized using Adam optimizer [18] with a batch size of 32. Images are cropped to a size of 96×96 pixels for training. For noise generation, we train individual networks for 300 epochs with the learning rate of 10^{-4} . For denoising, we select the DnCNN [27] as the default deep denoiser for comparison and train it for 300 epochs with the learning rate of 10^{-3} . The λ_1 , λ_2 , λ_3 and λ_4 in the loss functions are set to 0.1, 0.1, 50, 10 respectively.

4.2. Noise Synthesis on SIDDD

Compared Baselines. We compare NeCA with several noise models, including Additive White Gaussian Noise (AWGN), C2N [16], and the NLF (described in Equation (1)). To synthesize AWGN, we estimate the noise level from each noisy image by applying a noise estimation method introduced in [7] and add it to its corresponding clean image. To synthesize noise using the C2N, we directly utilize the pretrained model provided by the authors. For the NLF, we synthesize heteroscedastic Gaussian noise on the raw-RGB clean images from SIDDD, where the signal-dependent term σ_s^2 and signal-independent term σ_c^2 are obtained from the metadata provided by SIDDD. We then apply the same ISP pipeline as used in the SIDDD to render them to sRGB. We refer to this model as NLF-ISP for simplicity.

Preparation. We evaluate the results of generated noise on each camera in SIDDD, where 80% of image pairs are allo-

Camera	Metrics	AWGN	C2N [16]	NeCA	NLF-ISP	Real
G4	KL	1.9755	0.1660	0.0242	0.0102	-
	PSNR	28.15	37.81	38.85	38.51	40.60
GP	KL	1.8351	0.1315	0.0432	0.0126	-
	PSNR	28.45	37.08	37.72	37.74	38.33
IP	KL	1.8562	0.0581	0.0410	0.0475	-
	PSNR	28.01	39.12	39.46	39.53	39.45
N6	KL	2.1465	0.3524	0.0206	0.0063	-
	PSNR	26.31	33.59	35.54	34.84	35.56
S6	KL	0.4517	0.4517	0.0302	0.0902	-
	PSNR	27.22	33.18	35.56	35.99	36.85
Average	KL	2.0062	0.2129	0.0342	0.0414	-
	PSNR	27.90	36.37	37.58	37.59	38.27

Table 1. Quantitative results of synthetic noise. The results are computed on the validation sets of five SIDDD cameras with KL divergence and PSNR (dB). The best results are highlighted in bold.

cated for training the noise synthesizing framework, while the rest 20% are reserved for validation. The quality of the synthesized noise was evaluated using two metrics: KL divergence and PSNR. We calculate the KL divergence between the histograms of ground truth noise in the validation set and the noise synthesized by NeCA with clean images and corresponding gain factors from the validation set. Notably, the gain factors used for evaluation are estimated by GENet from the noisy images paired with the clean images, as they cannot be set to random values for evaluation. Besides, we also use the PSNR to further evaluate the quality of synthesized noisy images. We train the DnCNN with the synthesized noisy-clean image pairs on the training set and apply it to denoise the noisy images from the validation set. We calculate the PSNR between the denoised images and corresponding clean images to evaluate the denoising performance. In order to maintain consistency between the training and validation sets, we ensure that both sets contain the same set of ISO levels.

Noise Synthesis Results. Table 1 shows the KL divergence and PSNR results computed on validation sets of five devices. For the results of average KL divergence over all five cameras, our method exhibits the best performance among all noise models. Additionally, our method lags slightly behind NLF-ISP by 0.01 dB on the average PSNR. It is worth noting that noise samples generated by NLF-ISP are first synthesized in the raw-RGB domain and then rendered to sRGB using the same ISP pipelines as in SIDDD, suggesting the minimal discrepancies between noise samples from NLF-ISP and real data. The similar results on each camera between NLF-ISP and our NeCA model demonstrate the promising performance of the proposed model. Figure 4 shows generated noise maps from compared methods. Remarkable visual similarities observed between generated noise maps and real noise maps indicate that our framework is capable to synthesize realistic noise.

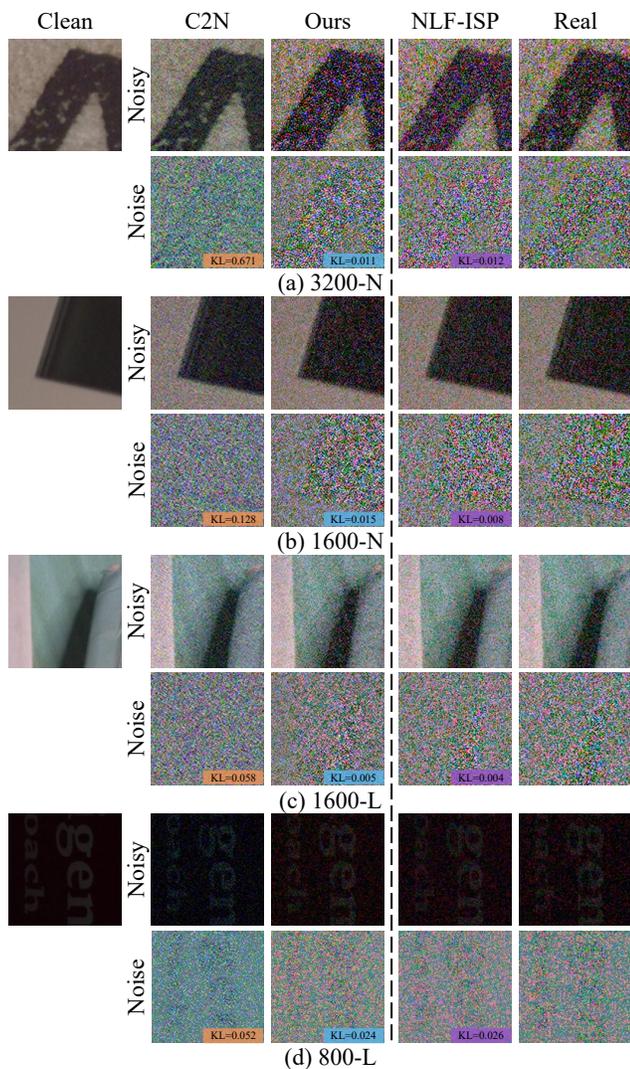


Figure 4. Visualization of synthetic noise samples under different ISO-lighting conditions on SIDD [2]. The displayed images, from left to right, correspond to clean image, C2N, Our method, NLF-ISP and real noisy image.

4.3. Applications on Real Image Denoising

Compared Baselines. Various noise generation methods are evaluated to demonstrate the effectiveness of these baselines performed on the downstream real image denoising task, including GCBD [8], C2N [16], Flow-sRGB [20], NeCA-S and NeCA-W. When assessing denoising performance, classical denoisers such as BM3D [9] and WNNM [12] are also included in the experiments.

Preparation. We establish the synthetic SIDD where clean images are from the original SIDD and noisy images are synthesized by using NeCA-W and NeCA-S. Specifically, the proposed framework is trained on the entire SIDD for each camera and the whole framework (NeCA-W) is used to

Method	SIDD		DND	
	PSNR(dB)	SSIM	PSNR(dB)	SSIM
BM3D [9]	25.65	0.685	34.51	0.851
WNNM [12]	25.78	0.809	34.67	0.865
GCBD [8]	-	-	35.58	0.922
C2N* [16]	33.76	0.901	36.08	0.903
Flow-sRGB* [20]	34.74	0.912	-	-
NeCA-S*	36.10	0.927	36.96	0.938
NeCA-W*	36.82	0.932	37.53	0.940
Real*	37.12	0.934	37.89	0.942

Table 2. Quantitative evaluation of denoising performance on SIDD and DND benchmark. * denotes the DnCNN denoiser is trained on either the synthetic or real image pairs with the SIDD. (red: the best result, blue: the second best)

generate noise for each clean image from the SIDD, where the gain factor is estimated from its paired noisy image. On the other hand, We train NeCA with a few paired images, e.g., three image pairs with varying ISO levels (800, 1600, 3200) from camera N6 and use only NCNet (NeCA-S) to generate signal-independent neighboring correlated (SINC) noise for clean images from SIDD, as seen in Figure 3. The synthesized SINC noise is added to the clean image. For each clean image, the noise level of AWGN is randomly selected from a range of [0, 75]. Our experiments with NeCA-S aim to demonstrate the advantages of explicitly modeling the neighboring correlation of real noise. Other sRGB real noise generation baselines, including C2N [16] and Flow-sRGB [20], also follow the same experimental settings with NeCA-W. With the synthetic noisy-clean image pairs, we train the DnCNN on either synthetic or real pairs of SIDD. Then the denoising performances are evaluated on both the SIDD and DND [23] benchmarks.

Results and Discussions. Table 2 shows the denoising results of the compared denoisers. Obviously, DnCNN trained on the synthetic samples from NeCA-W, achieves the best results among all compared methods in terms of both PSNR and SSIM. Specifically, NeCA-W gets 2.08 dB gains from Flow-sRGB on the SIDD benchmark, where Flow-sRGB is an end-to-end flow model which implicitly synthesizes real noise. The improvement of denoising performance obtained by NeCA-W indicates the accuracy of our noise model. Moreover, even though the denoising performance of NeCA-W still does not surpass the denoiser trained on the real data, the slight PSNR and SSIM discrepancies between them suggest our model does shrink this gap. Furthermore, the most impressive thing is that NeCA-S still achieves comparable denoising results on both the SIDD and DND benchmarks, outperforming the Flow-sRGB by a large margin. Note that the synthetic noise from NeCA-S is signal-independent. The superior performance of NeCA-S further verifies explicitly modeling neighboring correlation benefits the sRGB real noise synthesis.

Figure 6 and 7 show the denoised images from the SIDD

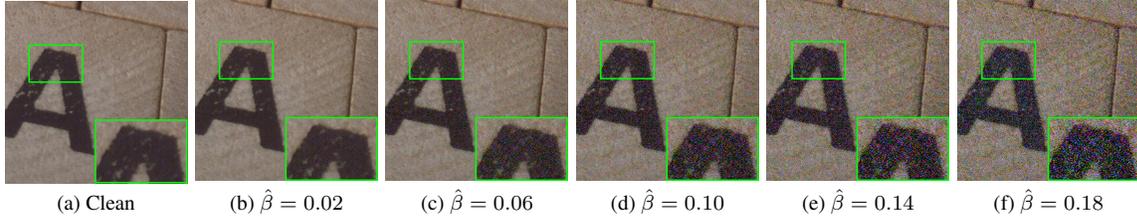


Figure 5. Results of controllable noise synthesis. The gain factor ranges from 0.02 to 0.18 with intervals of 0.04.

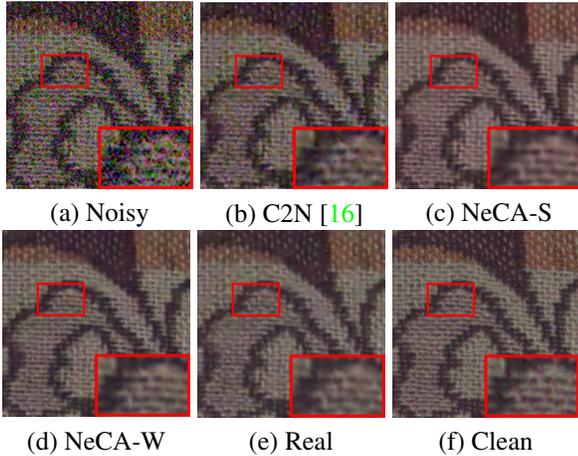


Figure 6. Denoising results on the SIDD dataset. DnCNN denoisers are trained on the noisy images from (b) C2N, (c, d) our models, and (e) real noisy images of the SIDD.

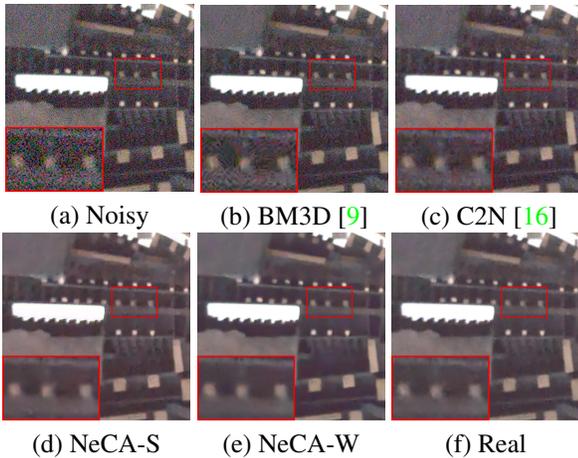


Figure 7. Denoising results on the DND dataset. DnCNN denoisers are trained on the noisy images from (c) C2N, (d, e) our models, and (f) real noisy images of the SIDD.

and DND datasets. The results indicate that the denoisers trained on the synthetic image pairs from NeCA-W and NeCA-S achieve similar denoising results compared to the denoiser trained on real image pairs. In contrast, the denoiser trained on noisy samples from C2N, which employs an unpaired training scheme, fails to suppress the noise effectively, partly due to its unpaired train scheme.

Loss	w/o \mathcal{L}_{std2}	w/o \mathcal{L}_{adv1}	w/o \mathcal{L}_{reg}	all
KL	0.052	0.048	0.108	0.041

Table 3. Ablation study on the effectiveness of different loss functions. We train the framework on the training set of camera IP and calculate KL divergence on its validation set.

4.4. Customized Generation

Our proposed noise synthesizing framework allows for controlling the generated noise with multiple noise levels by manipulating the gain factors. Figure 5 illustrates the controllable synthesizing results, which are generated by varying the gain factor within the range of 0.02 to 0.18 with intervals of 0.04. The results demonstrate that an increase in the gain factor value leads to a proportional increase in the magnitude of the generated noise.

4.5. Ablation Study

In this section, we conduct ablation studies to verify the effectiveness of individual loss functions in our framework, including \mathcal{L}_{std2} , \mathcal{L}_{adv2} and \mathcal{L}_{reg} . We exclude \mathcal{L}_{std1} and \mathcal{L}_{adv1} from evaluation since they are indispensable for framework training. As indicated in Table 3, the model achieves optimal performance in KL divergence with complete loss functions, suggesting all the components contribute to the final synthetic noise. However, removing \mathcal{L}_{reg} significantly reduces the KL divergence, suggesting the importance of stabilizing the training process. Moreover, both \mathcal{L}_{adv2} and \mathcal{L}_{std2} improve the quality of synthetic noise, supporting our claim that \mathcal{L}_{adv2} serves as a complementary loss for \mathcal{L}_{std2} , enabling the NPNet to predict more accurate noise levels.

5. Conclusion

In this paper, we propose a neighboring correlation-aware noise model for sRGB real noise generation. Our proposed method effectively bridges the gap between synthetic noise and real noise by explicitly modeling the signal dependency and neighboring correlation of real noise. The experimental results demonstrate the proposed noise model achieves superior performance on both real noise synthesis and downstream real image denoising tasks.

References

- [1] Abdelrahman Abdelhamed, Marcus A Brubaker, and Michael S Brown. Noise flow: Noise modeling with conditional normalizing flows. In *ICCV*, 2019. 1, 2
- [2] Abdelrahman Abdelhamed, Stephen Lin, and Michael S Brown. A high-quality denoising dataset for smartphone cameras. In *CVPR*, 2018. 6, 7
- [3] Saeed Anwar and Nick Barnes. Real image denoising with feature attention. In *ICCV*, 2019. 1
- [4] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016. 4
- [5] Tim Brooks, Ben Mildenhall, Tianfan Xue, Jiawen Chen, Dillon Sharlet, and Jonathan T Barron. Unprocessing images for learned raw denoising. In *CVPR*, 2019. 1, 2
- [6] Ke-Chi Chang, Ren Wang, Hung-Jin Lin, Yu-Lun Liu, Chia-Ping Chen, Yu-Lin Chang, and Hwann-Tzong Chen. Learning camera-aware noise models. In *ECCV*, 2020. 1, 2
- [7] Guangyong Chen, Fengyuan Zhu, and Pheng Ann Heng. An efficient statistical method for image noise level estimation. In *ICCV*, 2015. 6
- [8] Jingwen Chen, Jiawei Chen, Hongyang Chao, and Ming Yang. Image blind denoising with generative adversarial network based noise modeling. In *CVPR*, 2018. 2, 7
- [9] Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian. Image denoising by sparse 3-d transform-domain collaborative filtering. *TIP*, 2007. 7, 8
- [10] Alessandro Foi. Clipped noisy images: Heteroskedastic modeling and practical denoising. *Signal Processing*, 2009. 2
- [11] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 2020. 2
- [12] Shuhang Gu, Lei Zhang, Wangmeng Zuo, and Xiangchu Feng. Weighted nuclear norm minimization with application to image denoising. In *CVPR*, 2014. 7
- [13] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. *NIPS*, 2017. 5
- [14] Lanqing Guo, Siyu Huang, Haosen Liu, and Bihan Wen. Fino: Flow-based joint image and noise model. *arXiv preprint arXiv:2111.06031*, 2021. 1
- [15] Shi Guo, Zifei Yan, Kai Zhang, Wangmeng Zuo, and Lei Zhang. Toward convolutional blind denoising of real photographs. In *CVPR*, 2019. 1, 2
- [16] Geonwoon Jang, Wooseok Lee, Sanghyun Son, and Kyoung Mu Lee. C2n: Practical generative noise modeling for real-world denoising. In *ICCV*, 2021. 1, 2, 6, 7, 8
- [17] Dong-Wook Kim, Jae Ryun Chung, and Seung-Won Jung. Grdn: Grouped residual dense network for real image denoising and gan-based real-world noise modeling. In *CVPRW*, 2019. 1, 2
- [18] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6
- [19] Durk P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. *NIPS*, 2018. 2
- [20] Shayan Kousha, Ali Maleky, Michael S Brown, and Marcus A Brubaker. Modeling srgb camera noise with normalizing flows. In *CVPR*, 2022. 1, 2, 3, 7
- [21] Ce Liu, William T Freeman, Richard Szeliski, and Sing Bing Kang. Noise estimation from a single image. In *CVPR*, 2006. 2, 3
- [22] Seonghyeon Nam, Youngbae Hwang, Yasuyuki Matsushita, and Seon Joo Kim. A holistic approach to cross-channel image noise modeling and its application to image denoising. In *CVPR*, 2016. 1, 2, 3
- [23] Tobias Plotz and Stefan Roth. Benchmarking denoising algorithms with real photographs. In *CVPR*, 2017. 7
- [24] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *TIP*, 2004. 6
- [25] Kaixuan Wei, Ying Fu, Jiaolong Yang, and Hua Huang. A physics-based noise formation model for extreme low-light raw denoising. In *CVPR*, 2020. 1, 2
- [26] Zongsheng Yue, Qian Zhao, Lei Zhang, and Deyu Meng. Dual adversarial network: Toward real-world noise removal and noise generation. In *ECCV*, 2020. 1, 2
- [27] Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *TIP*, 2017. 6
- [28] Yi Zhang, Hongwei Qin, Xiaogang Wang, and Hongsheng Li. Rethinking noise synthesis and modeling in raw denoising. In *ICCV*, 2021. 1, 2
- [29] Yuqian Zhou, Jianbo Jiao, Haibin Huang, Yang Wang, Jue Wang, Honghui Shi, and Thomas Huang. When awgn-based denoiser meets real noises. In *AAAI*, 2020. 4, 5