C2N: Practical Generative Noise Modeling for Real-World Denoising

Response to Reviewer 1 and 2

R1&2-1.Generalization of noise level and camera models. R2 asks whether a specific noise level or camera model is used in our experiments. We use the entire samples in the SIDD [2] dataset obtained from five different camera models under various illuminations, ISO levels, and shutter speeds. Thus we confirm that our model is not fitted to a specific noise level or camera type but is more general. We also note that results in Figure 4, Figure 5, and Table 1 demonstrate that our C2N faithfully learns different types of synthetic and real noise distributions and support the generalizability of C2N. We will include more thorough ablations and analysis in the revised manuscript.

R1&2-2. Control of generated noise level. Since the noise level and camera model are not given in the real-world cases, we do not explicitly control them. Instead, we adopt the random latent vector r to implicitly generate noise of various intensities. Figure S3 in our Supple. illustrates that changing the r results to the different noise levels.

Response to Reviewer 1

R1-1. Evaluation of the generated noisy images. Our C2N is designed to facilitate reliable denoising models for real-world images. Therefore, we evaluate the validity of C2N indirectly by the performance improvement of the denoising models coupled with C2N. An additional experiment shows that the KL-divergence metric is less correlated with better denoising performance [1]. For example, our generators with KL divergence of 0.202 and 0.108 show denoising performances of 32.63dB and 29.61dB in PSNR on the SIDD validation set, respectively. Still, we agree that a concrete evaluation of the generated noisy images is also essential and will cover it in our final manuscript.

R1-2. Comparison with the previous noise generators. We mainly focus on introducing a novel noise generator with individual modules to express corresponding noise components explicitly. On the other hand, most of the existing noise generators are designed for specific purposes. For example, the GCBD [13] and GAN2GAN [11] methods mainly focus on handling signal-independent synthetic noise. While some methods have the capability of imitating real-world noise, they are available for particular situations; The NoiseFlow [1] model requires paired data, and the NT-GAN [52] method utilizes the known CRFs, i.e., camera response function. Since several generation-based methods are developed on their own formulations, it is not easy to fairly compare the proposed C2N model with the others. The UIDNet [23] formulation shares a similar assumption to the proposed method, but no codes and results are released for a fair comparison. Still, we appreciate the suggestion that extensive comparison between several methods is important, and we will include it in our revised manuscript.

R1-3. About the heteroscedastic Gaussian and Poisson-Gaussian noise model. We adopt the formulation of the heteroscedastic Gaussian noise model [16] from several previous works [1, 20]. The Gaussian distribution with signal-dependent variance in this model replaces the Poisson part of the Poisson-Gaussian model, which we misrepresented due to the similarity in signal-dependent properties. We will correct the term in the final manuscript.

Response to Reviewer 2

R2-1. The learning procedure. Our feature extractor produces local mean and standard deviation of the noise before transformations, which can be jointly trained in an end-to-end manner with the entire C2N framework. We define this extractor as five ResBlocks as L430-441 in our main manuscript. As for the input of this extractor, we concatenate the input clean image and the random vector r. We used r as a latent vector that encode the information of the capturing condition(ISO, shutter speed, camera model, etc.) of each scenes, for such condition affect in the noise distribution [2, 27] but not available in realistic situations. Thus we sample the r from the normal distribution similarly to the conventional generative models [40, 42], as we describe in L314-323 of our main manuscript.

R2-2. More details in the model ablation. We find that there could be confusion on Table 2. We note that G^I and G^D include $G^I_{3\times3}$ and $G^D_{3\times3}$, respectively, as shown in Figure 3. That means if G^* is off, $G^*_{3\times3}$ is off consequently. With such logic, the third column '$G^I_{3\times3}$, $G^D_{3\times3}$' in Table 2 denotes the existence of each 3×3 convolution module when the corresponding G^I or G^D are enabled. Therefore, the 4th and 5th rows of Table 2 represent the cases of without G^D (and $G^D_{3\times3}$), and without G^I (and $G^I_{3\times3}$), respectively, which are the cases that R2 inquires. The effect of the signal-dependent component is verified by comparing the 2nd and 3rd, or 4th and 6th rows, where they refer that using G^I and G^D together produces better results, along with Figure 4. We will fix the misleading expressions and include more detailed descriptions in the final manuscript.

Response to Reviewer 3

R3-1. Related works to be included. We will discuss about the mentioned papers in our revised manuscript.

R3-2. Comparison with the DIDN baseline. We have trained the DIDN [49] model without our C2N framework, using the same synthetic Gaussian noise generated as described in L148-151 in the Supple. The result show that DIDN overfits to the Gaussian noise and gets denoising performance of 23.97dB in PSNR, which is 11.05dB lower than that of C2N+DIDN. We will include the result, also with the performance of DIDN trained on real data pairs and more optimized synthetic noise in the revised manuscript to show the effectiveness of our method further.