Supplementary Material *for*
C2N: Practical Generative Noise Modeling for Real-World Denoising

Geonwoon Jang* Wooseok Lee* Sanghyun Son Kyoungmu Lee
ASRI, Department of ECE, Seoul National University
{onwoono, adntjr4}@gmail.com, {thstkdgus35, kyoungmu}@snu.ac.kr

S1. Measurement of KL-Divergence

In Section 4.3 of our main manuscript, the KL-divergence is evaluated with the following definition:

$$D_{\mathrm{KL}}(P_n \| P_{\hat{n}}) = \sum_{i=0}^{255} p_n(i) \log \frac{p_n(i)}{p_{\hat{n}}(i)}, \tag{S1}$$

where P_n and $P_{\hat{n}}$ denote the distribution of the ground truth and generated noise maps, respectively. Also, p_n and $p_{\hat{n}}$ are the probability histogram of all noise maps from P_n and $P_{\hat{n}}$. The histogram p_n is calculated as follows:

$$p_n(i) = \frac{1}{mCHW} \sum_{k,c,h,w}^{m,C,H,W} \mathbf{1}_{\{n_{k,c,h,w}=i\}}, \tag{S2}$$

where k, c, h, w denote indices of noise maps and color channel, height, width of the corresponding image of size H, W, and C and m are the number of color channels and the number of the noise maps we want to evaluate, respectively. $\mathbf{1}_{\{n_{k,c,h,w}=i\}}$ is used as an indicator function which refers the number of pixels with noise intensity of i. We note that the histogram is calculated together between different color channels. The remaining histogram $p_{\hat{n}}(i)$ for generated noise map is also calculated in the same manner as (S2).

S2. Implementation Details

Discriminator. We define the discriminator architecture as a sequence of six ResBlocks with 3×3 convolutions and the following 1×1 convolution, which reduces the number of channels to one. Figure S1 shows the discriminator architecture of our C2N framework. The output values from the discriminator is averaged across spatial dimension to indicate whether the image is real or generated one. We set the number of channels for all 3×3 ResBlocks as 64, which is the same number as the C2N generator.

*Authors contributed equally.

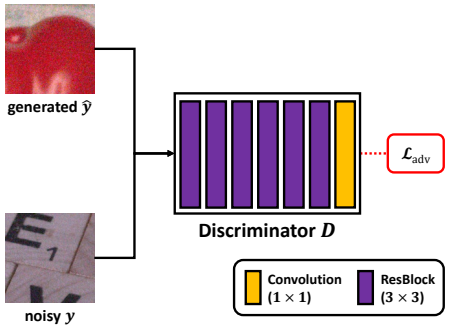


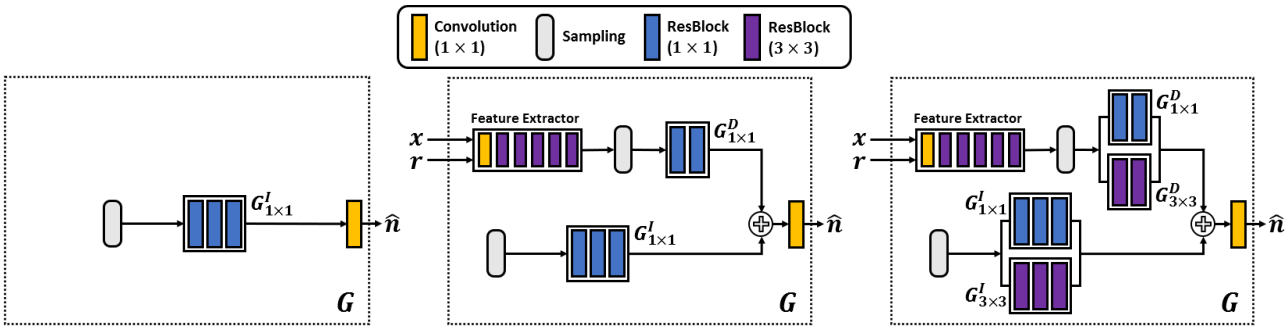
Figure S1: **Discriminator architecture.** The ResBlocks, like those of the generator, are modified for low-level vision problems [7].

Self-ensemble. We apply the self-ensemble technique [8, 7] to acquire final denoised results. For a noisy image, we augment inputs by flipping and 90° rotations and evaluate the denoising model 8 times including the original. We convert each output to the original geometry by the inverse transformations and average all to get the self-ensembled result.

Fine-tuning on the DND In Section 4.4 of our main manuscript, we also report the performance of the model trained on the SIDD and then fine-tuned on the DND. We fine-tune the C2N generator by training it for 16 more epochs on the DND dataset, with initial learning rate of 10^{-5} multiplied by 0.8 for every 3 epochs.

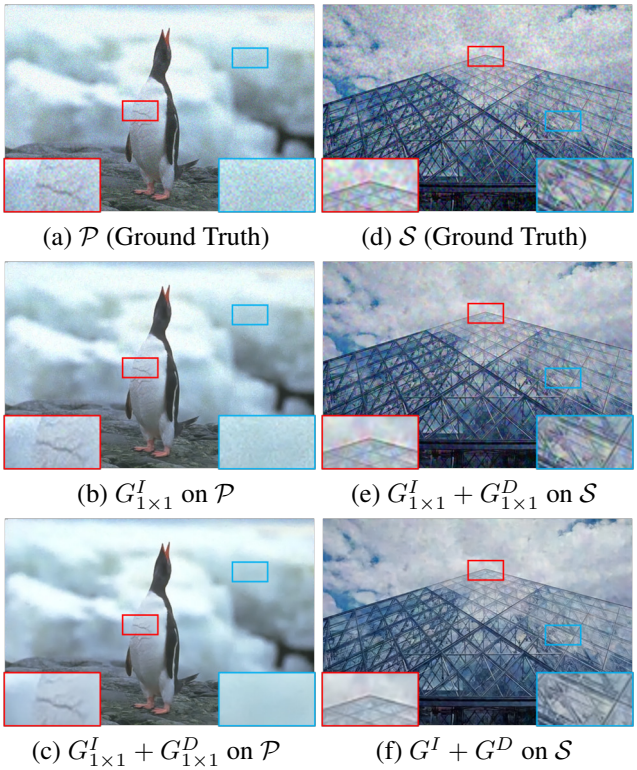
S3. Architectures in the Model Analysis

For the model ablation study on synthetic noise in Section 4.2 of our main manuscript, we use the notations $G^I_{1\times1}$, $G^I_{1\times1} + G^D_{1\times1}$, and $G^I + G^D$ to refer the variants of our C2N. Figure S2 illustrates detailed diagrams of those variants. The model $G^I_{1\times1}$ in Figure S2a consists of a signal-independent transformation module with 1×1 convolutions that do not take clean image x and random vector r as input. In contrast, the $G^I_{1\times1} + G^D_{1\times1}$ in Figure S2b has both modules to produce signal-independent and signal-dependent



(a) $G^I_{1\times1}$ (b) $G^I_{1\times1} + G^D_{1\times1}$ (c) $G^I + G^D$

Figure S2: **Generator architectures in synthetic experiment.** (a) signal-independent and spatially uncorrelated noise generator. (b) signal-dependent but spatially uncorrelated noise generator. (c) Our whole C2N generator, which can generate signal-dependent and spatially correlated noise. Notations are same with Figure 3 in our main manuscript. $\hat{n}$ denotes generated noise map and replication procedure of random vector r is skipped for visualization.



(a) $\mathcal{P}$ (Ground Truth) (d) $\mathcal{S}$ (Ground Truth)

(b) $G^I_{1\times1}$ on $\mathcal{P}$ (e) $G^I_{1\times1} + G^D_{1\times1}$ on $\mathcal{S}$

(c) $G^I_{1\times1} + G^D_{1\times1}$ on $\mathcal{P}$ (f) $G^I + G^D$ on $\mathcal{S}$

Figure S3: **Denoising results from model ablation study on various synthetic noise.** (a, d) Synthetic ground-truth noisy image of Poisson $\mathcal{P}$ and spatially correlated Gaussian noise $\mathcal{S}$. (b, c) Denoising results of the C2N variant trained on the Poisson noise and its following denoiser. (e, f) Denoising results of the C2N variant trained on the spatially correlated Gaussian noise and its following denoiser.

noise terms but only with 1×1 convolutions. Lastly, the $G^I + G^D$ in Figure S2c consists of all modules for the proposed C2N, including $G^I_{3\times3}$ and $G^D_{3\times3}$.

C2N Model	Test Noise Type $\mathcal{G}$	$\mathcal{P}$	$\mathcal{S}$
$G^I_{1\times1}$	30.69	35.21	26.09
$G^I_{1\times1} + G^D_{1\times1}$	30.66	**35.80**	28.91
$G^I + G^D$	30.40	35.23	**31.03**

Table S1: **Denoising performance on various synthetic noise.** PSNR(dB) is calculated on the CBSD68 dataset.

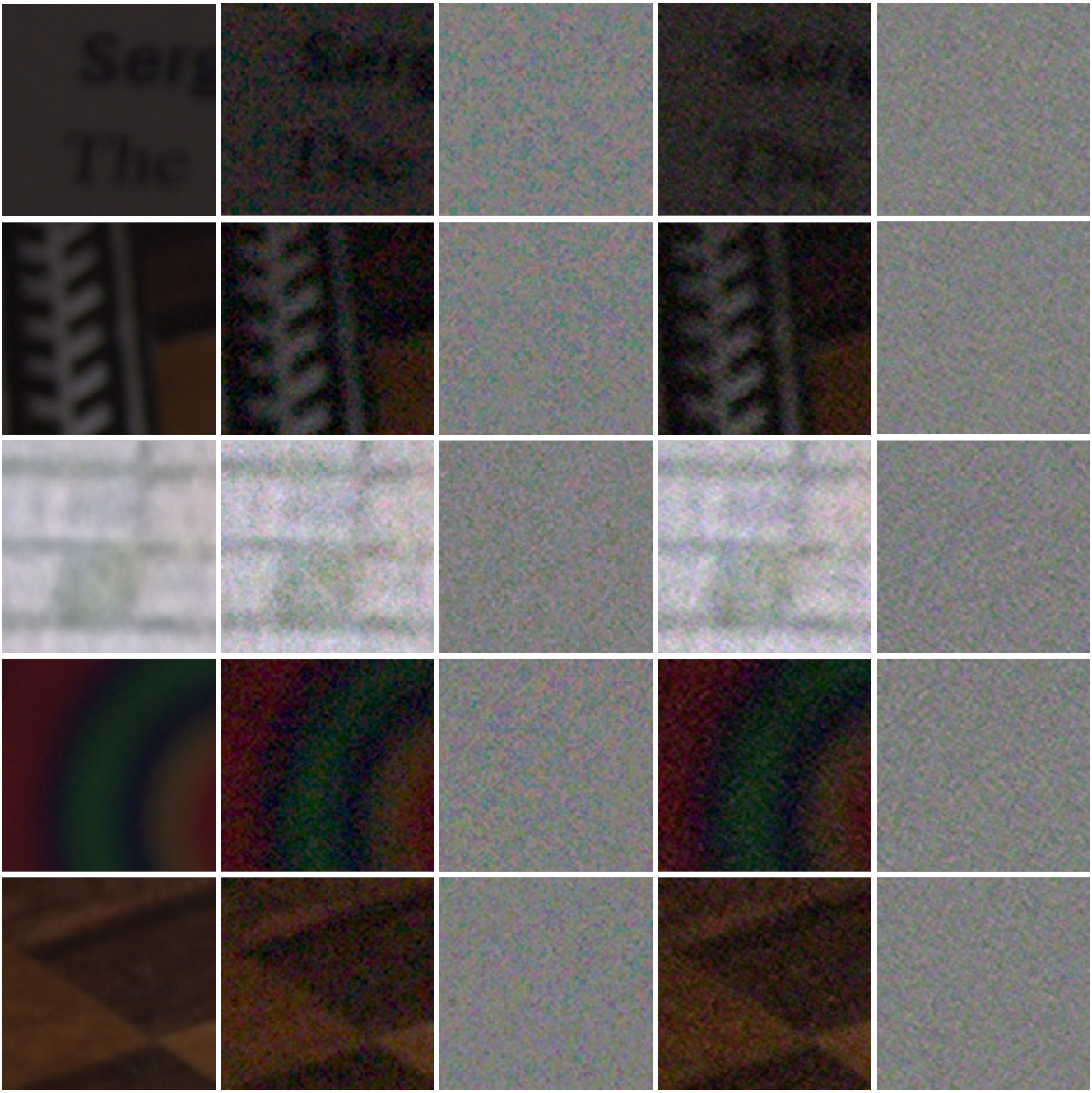
S4. Denoising on Synthetic Noise

The primary purpose of the model ablation study on synthetic noise in Section 4.2 of our main manuscript is to demonstrate how our C2N can generate noise with various characteristics. Still, we can also train the denoising networks followed by each C2N variant and evaluate their performance on various synthetic noise, as shown in Table S1 and Figure S3. $\mathcal{G}$ stands for Gaussian noise of $\sigma = 25$, $\mathcal{P}$ stands for Poisson noise $n \sim Poi(x) - x$ where $Poi(x)$ denotes the Poisson distribution similar to [1], and $\mathcal{S}$ stands for spatially correlated Gaussian noise. The same notations of $\mathcal{P}$ and $\mathcal{S}$ are used in Section 4.2 of our main manuscript.

Unlike $G^I_{1\times1} + G^D_{1\times1}$, $G^I_{1\times1}$ cannot handle signal-dependent noise level as shown in Figure 5 of our main manuscript. As a result, the denoiser followed by $G^I_{1\times1}$ generator in Figure S3b is not appropriate to remove non-uniform Poisson noise. Meanwhile, in Figure S3e the denoiser followed by $G^I_{1\times1} + G^D_{1\times1}$ outputs images that still contain noise term of $\mathcal{S}$, since the $G^I_{1\times1} + G^D_{1\times1}$ tends to generate artifacts instead of spatially correlated noise, as shown in Figure 4 of our main manuscript.

S5. Visualizing Generated Noise

Comparison between generated samples and the real-world noise. Figure S4 visually compares more pseudo-noisy samples generated by our C2N and ground-truth noise maps. The proposed C2N can synthesize samples that



(a) Clean (b) GT (c) GT res (d) C2N (e) C2N res

Figure S4: **Examples of ground truth noisy image and generated image from our C2N.** (a) Clean image, (b) Ground truth noisy image and (c) its residual noise map, (d) Generated noisy image from the proposed C2N and (e) its residual noise map. Best with zoomed.

closely resemble ground-truth noise without significant artifacts.

Latent space interpolation. We also provide a qualitative study on the effects of r, the input random vector of the generator. We sample two r vectors which generate low and high level noise and visualize the generated images with interpolated r as following equation, $r = (1 - \lambda)r_1 + \lambda r_2$. Here, r_1 and r_2 are the two sampled vectors and $\lambda \in [0, 1]$ is the interpolation factor. The qualitative results in Figure S5 illustrates that the random vector determines the property of

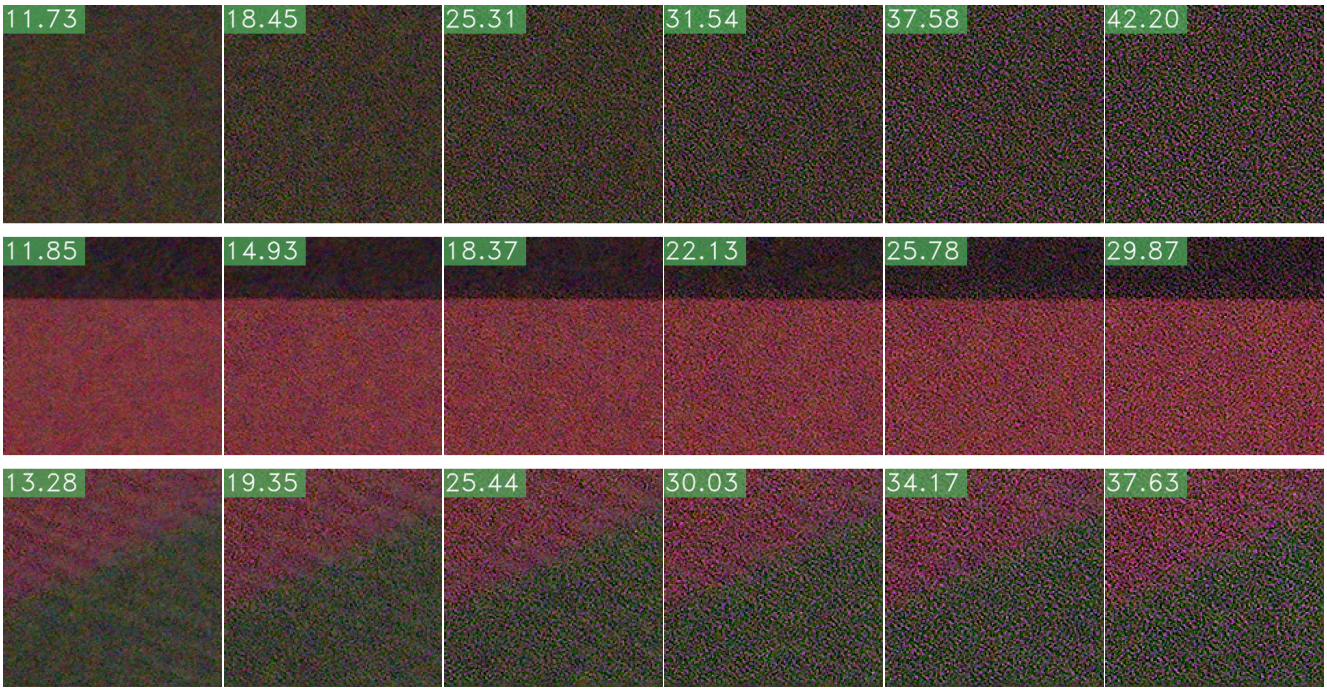


Figure S5: **Image generation with interpolated r .** For the same clean image patch, we interpolate two different r vectors with a factor λ and obtain the resulting images. Each column is generated using same vector r. The standard deviation of each residual noise map is displayed in each image. Best with zoomed.

synthesized noise in our C2N framework. In other words, our approach can learn to generate real-world noise that corresponds to varying conditions, such as strong or weak noise from various camera types in the SIDD [2].

S6. Practical Data Constraints in C2N

To apply the generative noise modeling methods in practical situations, two kinds of data constraints should be further considered for better usability. First, due to several physical limitations [6], it is not feasible to capture an ideal clean image from the wild. Rather, a long sequence of aligned noisy images must be captured beforehand [2] to synthesize the pseudo-clean reference. Thus, only a few clean images are available from the same scene distribution of the noisy images in a real situation. Secondly, once the noise generator is trained on the desired noisy image distribution P_N and clean image distribution P_C, it should be able to produce pseudo-noisy images paired to any clean image x from different clean image distribution P'_C to train a denoising model. Various real-world noisy image datasets have scenes that differ in many points, such as types and scales of the contents or illumination, making a model hard to learn the noise distribution distinct from the domain of scenes.

The existing generative noise modeling methods [3, 1, 4] used a large number of samples in P_C, and assumed the external clean image distribution P'_C for training denoising network to be the same as P_C. Such a setting is possible only if a sufficiently large noisy and clean image dataset is given, which is not a practical situation. We examine whether our method can maintain its usability under this problem that have not been explored before.

Table S2 shows that our method fairly preserves its performance without collapsing under two data constraints, (1) where not enough clean images in P_C are given to train the noise generator, (2) where the clean images from different scene distribution P'_C are used to train the following denoising model. Our method already uses surprisingly small amount of samples for training the C2N model, compared to $\sim$500K image patches of 64×64 size used in the previous generative noise modeling methods [1, 4]. The C2N model trained with much smaller amount of clean images in P_C still shows performance comparable to previous unsupervised denoising methods. Also for the case of P'_C to be different to P_C, our C2N can still train the following denoising model with its generated pseudo-noisy data. Although our method shows generalization ability in these situations with data constraints, we believe that further improvement to resolve such problems entirely would be an essential topic to handle in future work.

Number of Samples $\sim P_C$	P'_C	PSNR(dB)	SSIM
36K (100%)	S	34.08	0.909
	D	31.72	0.826
	B	31.74	0.825
	U	31.32	0.803
18K (50%)	S	33.53	0.882
	D	30.68	0.760
	B	29.96	0.742
	U	29.72	0.741
720 (2%)	S	31.98	0.847
	D	29.36	0.745
	B	29.27	0.735
	U	29.21	0.738
360 (1%)	S	31.84	0.849
	D	29.35	0.740
	B	29.08	0.733
	U	29.24	0.739

Table S2: **Denoising performance of our C2N under data constraints.** We use the Urban100 [5] dataset along with the other datasets mentioned in main manuscript as the samples of P'_C. S, D, B, U denote the SIDD, the DIV2K high-resolution images, the BSD traning images, and the Urban100, respectively. P_C is fixed to S for all experiments. Evaluation is done on the SIDD validation set.

References

[1] Abdelrahman Abdelhamed, Marcus A Brubaker, and Michael S Brown. Noise flow: Noise modeling with conditional normalizing flows. In *ICCV*, 2019. 2, 4

[2] Abdelrahman Abdelhamed, Stephen Lin, and Michael S Brown. A high-quality denoising dataset for smartphone cameras. In *CVPR*, 2018. 4

[3] Jingwen Chen, Jiawei Chen, Hongyang Chao, and Ming Yang. Image blind denoising with generative adversarial network based noise modeling. In *CVPR*, 2018. 4

[4] Zhiwei Hong, Xiaocheng Fan, Tao Jiang, and Jianxing Feng. End-to-end unpaired image denoising with conditional adversarial networks. In *AAAI*, 2020. 4

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

[6] Jaakko Lehtinen, Jacob Munkberg, Jon Hasselgren, Samuli Laine, Tero Karras, Miika Aittala, and Timo Aila. Noise2Noise: Learning image restoration without clean data. In *ICML*, 2018. 4

[7] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Enhanced deep residual networks for single image super-resolution. In *CVPRW*, 2017. 1

[8] Radu Timofte, Rasmus Rothe, and Luc Van Gool. Seven ways to improve example-based single image super resolution. In *CVPR*, pages 1865–1873, 2016. 1