Learning Camera-Aware Noise Models
[Supplementary Material]

Anonymous ECCV submission

Paper ID 4617

A Network Architectures

We apply a U-Net [1] architecture to the generator G, and Table 1 shows the detailed configuration. The first five layers correspond to the encoder followed by four residual blocks, and the last five layers are the decoder. The residual block consists of two consecutive convolutions as well as a skip connection across the block. Besides, the latent vector from the camera encoder E is concatenated with the output of the 2$^{\text{nd}}$ residual block.

Table 2 shows the architecture of the discriminator D, which is similar to PatchGAN [2]. The out_{D_f} and out_D are used for the feature matching loss L_{FM} and adversarial loss L_{Adv}, respectively. Note that D determines the score of realness at the scale of 46×46 according to the receptive field.

Finally, the architecture of the camera encoder E is shown in Table 3. To make latent vectors irrelevant to the spatial domain, we perform a global average pooling at the last layer. The latent vector is then concatenated with the middle features of G by expanding the spatial dimension.

B Control of Noise Levels

Recall that the noise level of the final synthetic noise $\tilde{\mathbf{n}}$ can be controlled by adjusting the parameters of Poisson-Gaussian noise model for the initial synthetic noise $\tilde{\mathbf{n}}_{\text{init}}$. For the same camera, these parameters are proportional to the digital gain, which is highly correlated to the ISO. Therefore, different noise levels should be observed in different ISOs. Fig. 1 shows the examples of noise and noisy image pairs from various noise models in a wide range of ISOs. We can find that as the ISO ascends, our noise samples become much noisier obviously. Moreover, our noise model always outperforms the compared methods in terms of Kullaback-Leibler divergence measurement.

C More Qualitative Results

More synthesized noise samples as well as the corresponding noisy images are shown in Figs. 2–5. More qualitative results of real image denoising are shown in Figs. 6–9.

Table 1. Architecture of the Generator. The notation $[\cdot, \cdot]$ represents concatenation and C, RES, T respectively denote convolution, residual block, and transposed convolution. The *SN-IN* indicates a Spectral Normalization [3] followed by an Instance Normalization [4] and LReLU is the Leaky ReLU [5]

	Input	Output	Kernel Size	Channels In	Channels Out	Stride	Norm.	Activ.	Output Size
C	in_G	c1	4×4	8	64	2	-	LReLU	$\frac{h}{2} \times \frac{w}{2}$
C	c1	c2	4×4	64	128	2	*SN-IN*	LReLU	$\frac{h}{4} \times \frac{w}{4}$
C	c2	c3	4×4	128	256	2	*SN-IN*	LReLU	$\frac{h}{8} \times \frac{w}{8}$
C	c3	c4	4×4	256	512	2	*SN-IN*	LReLU	$\frac{h}{16} \times \frac{w}{16}$
C	c4	c5	4×4	512	512	2	*SN-IN*	LReLU	$\frac{h}{32} \times \frac{w}{32}$
RES	c5	res1	3×3	512	512	1	-	-	$\frac{h}{32} \times \frac{w}{32}$
RES	res1	res2	3×3	512	512	1	-	-	$\frac{h}{32} \times \frac{w}{32}$
RES	[res2, out_E]	res3	3×3	1024	1024	1	-	-	$\frac{h}{32} \times \frac{w}{32}$
RES	res3	res4	3×3	1024	1024	1	-	-	$\frac{h}{32} \times \frac{w}{32}$
T	res4	t1	4×4	1024	512	1/2	*SN-IN*	LReLU	$\frac{h}{16} \times \frac{w}{16}$
T	[t1, c4]	t2	4×4	1024	256	1/2	*SN-IN*	LReLU	$\frac{h}{8} \times \frac{w}{8}$
T	[t2, c3]	t3	4×4	512	128	1/2	*SN-IN*	LReLU	$\frac{h}{4} \times \frac{w}{4}$
T	[t3, c2]	t4	4×4	256	64	1/2	*SN-IN*	LReLU	$\frac{h}{2} \times \frac{w}{2}$
T	[t4, c1]	out_G	4×4	128	4	1/2	*SN-IN*	Tanh	$h \times w$

Table 2. Architecture of the Discriminator

	Input	Output	Kernel Size	Channels In	Channels Out	Stride	Norm.	Activ.	Output Size
C	in_D	d1	4×4	8	64	2	-	LReLU	$\frac{h}{2} \times \frac{w}{2}$
C	d1	d2	4×4	64	128	2	*SN-IN*	LReLU	$\frac{h}{4} \times \frac{w}{4}$
C	d2	out_{D_f}	4×4	128	256	2	*SN-IN*	LReLU	$\frac{h}{8} \times \frac{w}{8}$
C	out_{D_f}	out_D	4×4	256	1	1	*SN-IN*	-	$\frac{h}{16} \times \frac{w}{16}$

Table 3. Architecture of the Camera Encoder. Note that POOL represents global average pooling

	Input	Output	Kernel Size	Channels In	Channels Out	Stride	Norm.	Activ.	Output Size
C	in_E	e1	7×7	4	64	1	-	LReLU	$h \times w$
C	e1	e2	4×4	64	128	2	*SN-IN*	LReLU	$\frac{h}{2} \times \frac{w}{2}$
C	e2	e3	4×4	128	256	2	*SN-IN*	LReLU	$\frac{h}{4} \times \frac{w}{4}$
C	e3	e4	4×4	256	512	2	*SN-IN*	LReLU	$\frac{h}{8} \times \frac{w}{8}$
POOL	e4	out_E	-	512	512	-	-	-	1×1

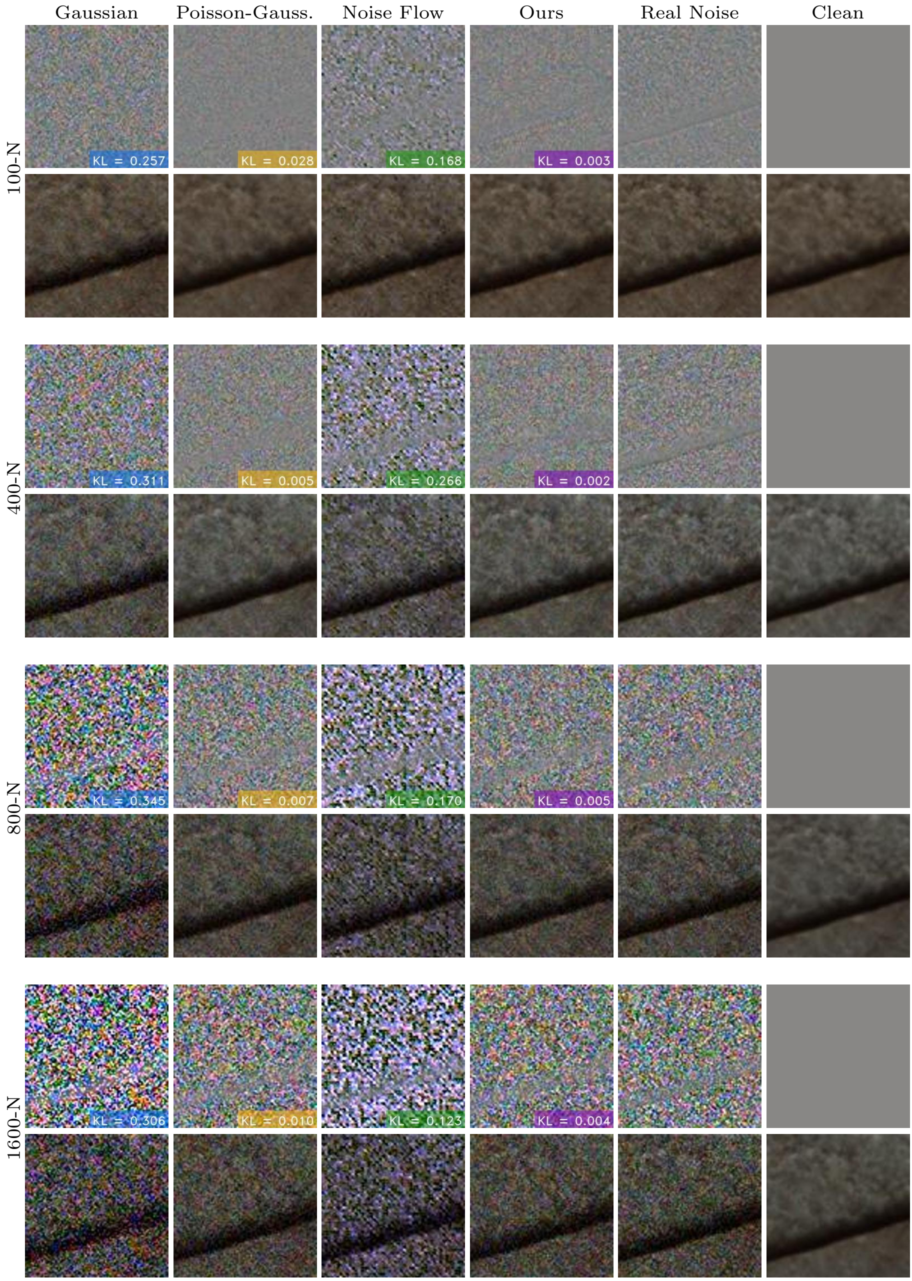

Fig. 1. Different noise levels in different ISOs. Each column represents a noise modeling method, and each two consecutive rows correspond to a pair of noise and noisy image in a specific ISO (from 100-N to 1600-N)

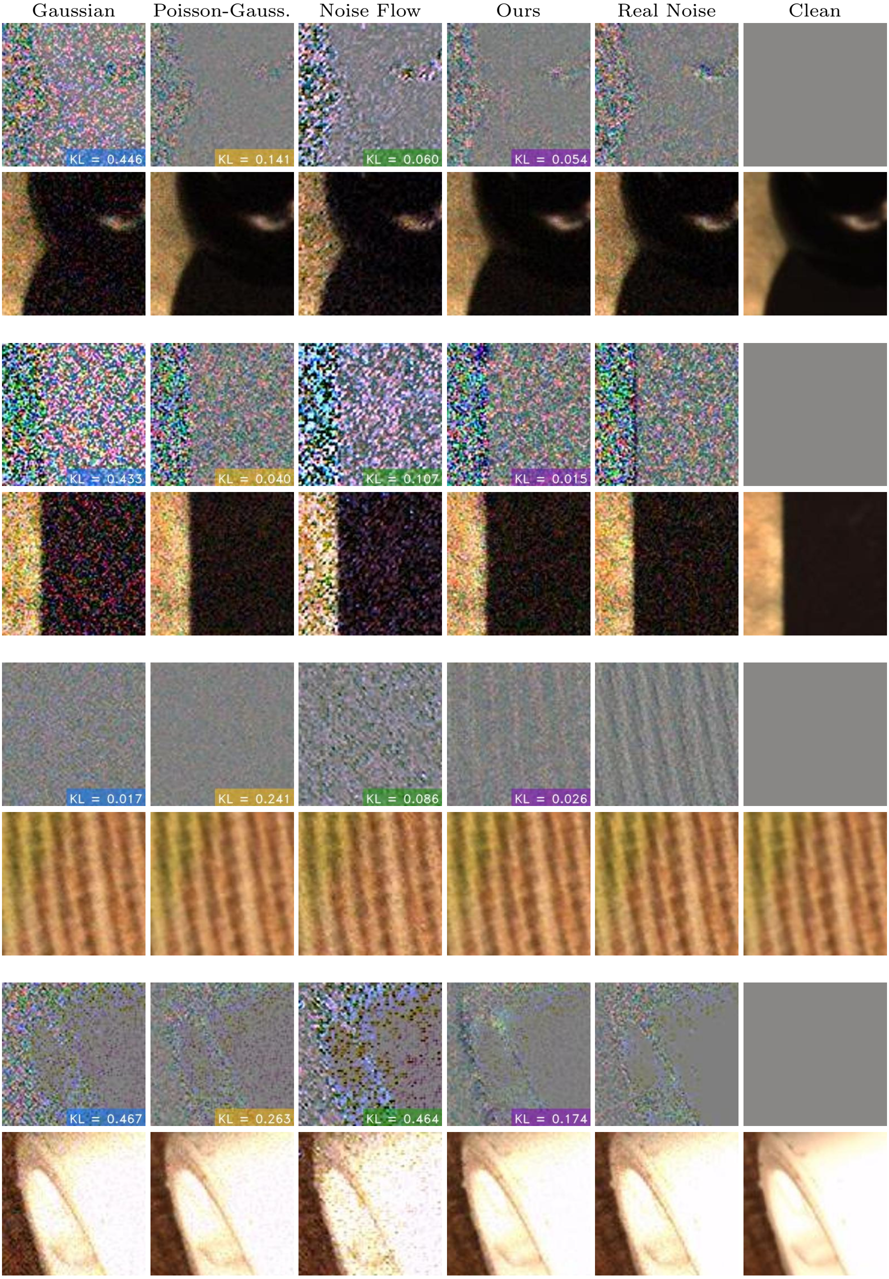

Fig. 2. More Qualitative Results of Noise Modeling (1)

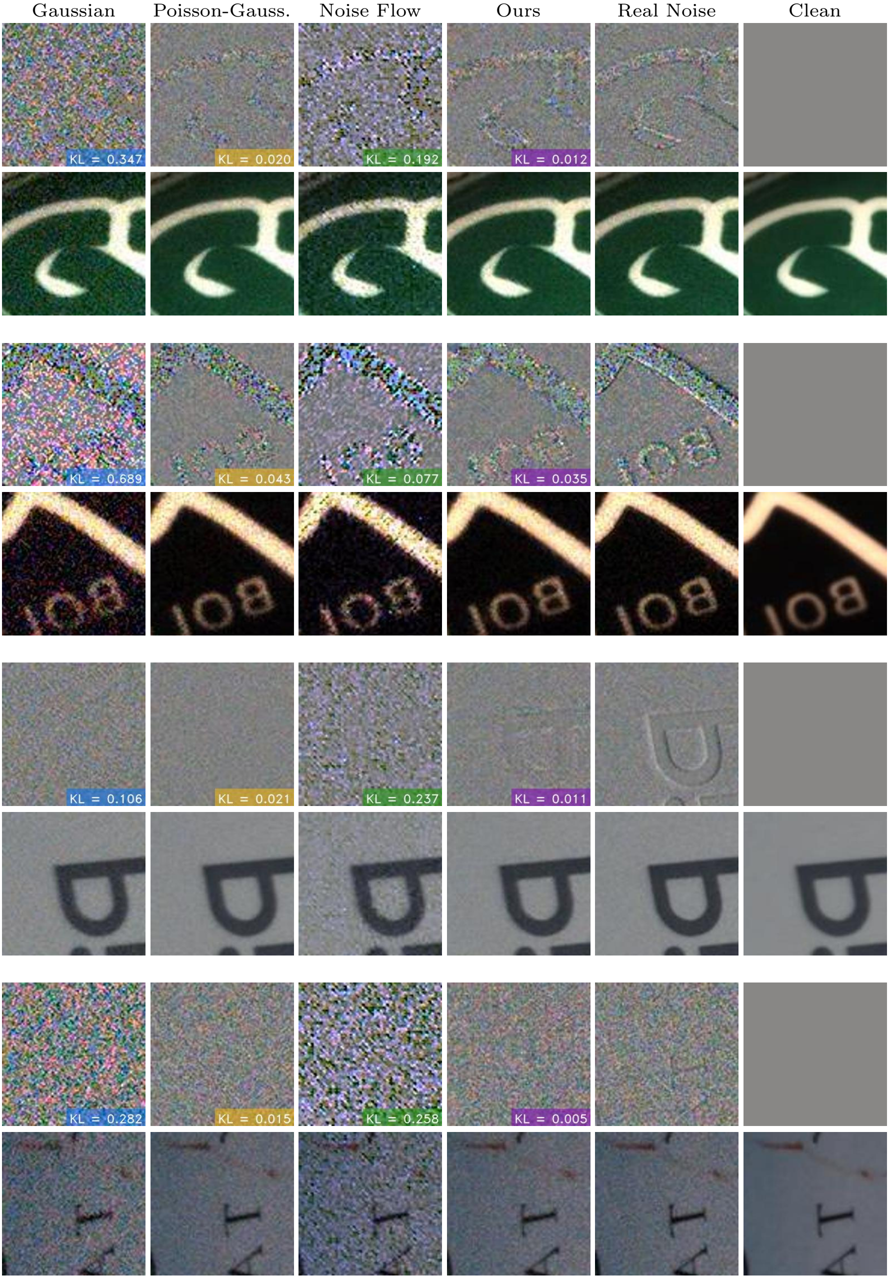

Fig. 3. More Qualitative Results of Noise Modeling (2)

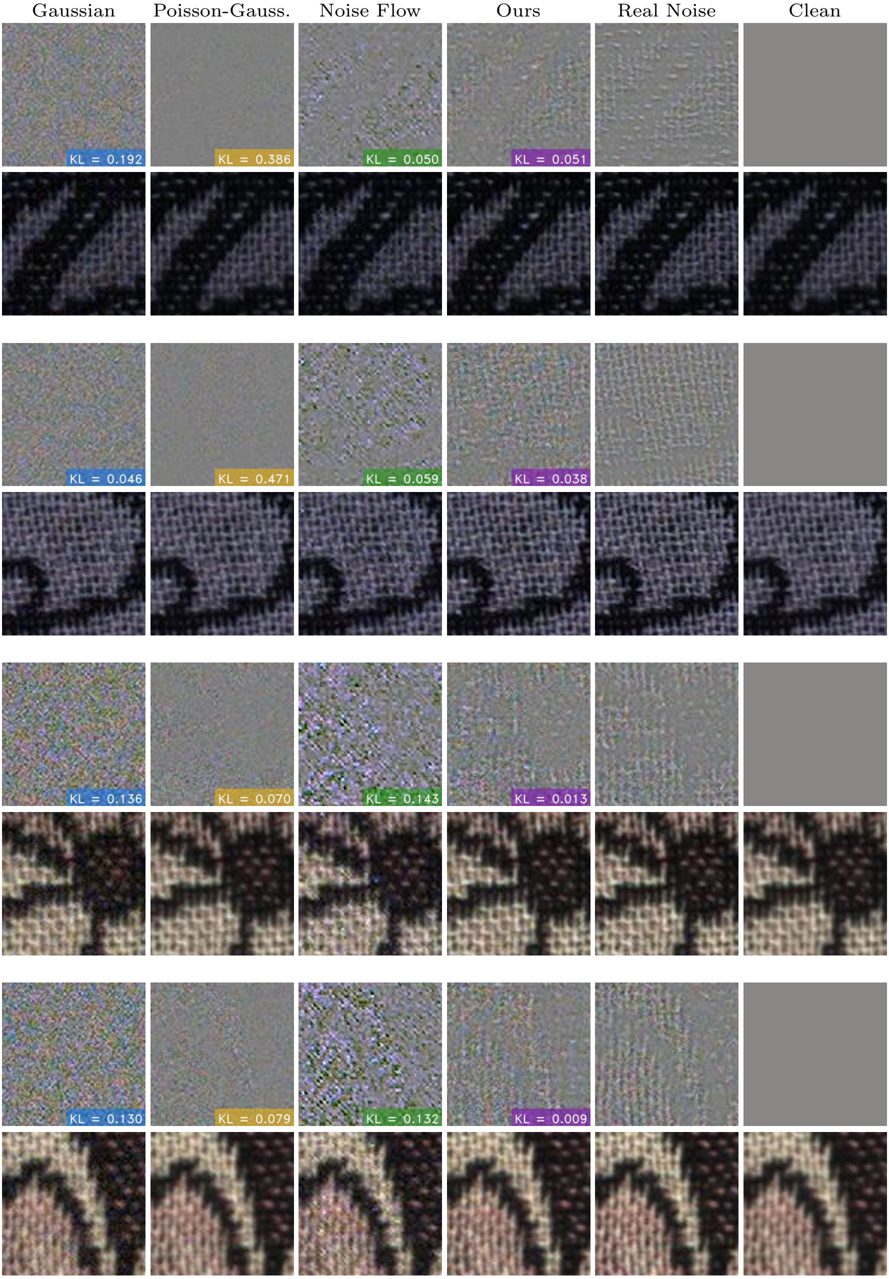

Fig. 4. More Qualitative Results of Noise Modeling (3)

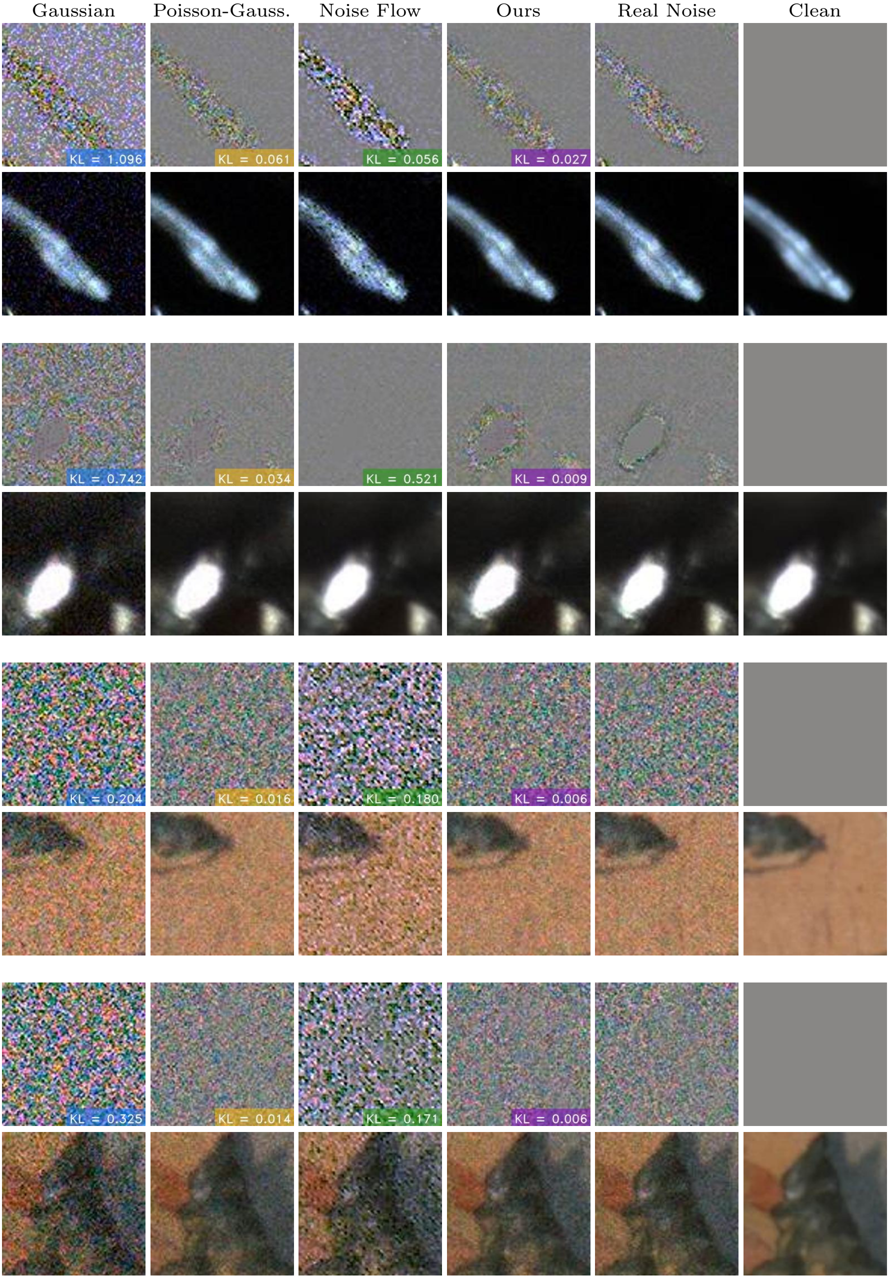

Fig. 5. More Qualitative Results of Noise Modeling (4)

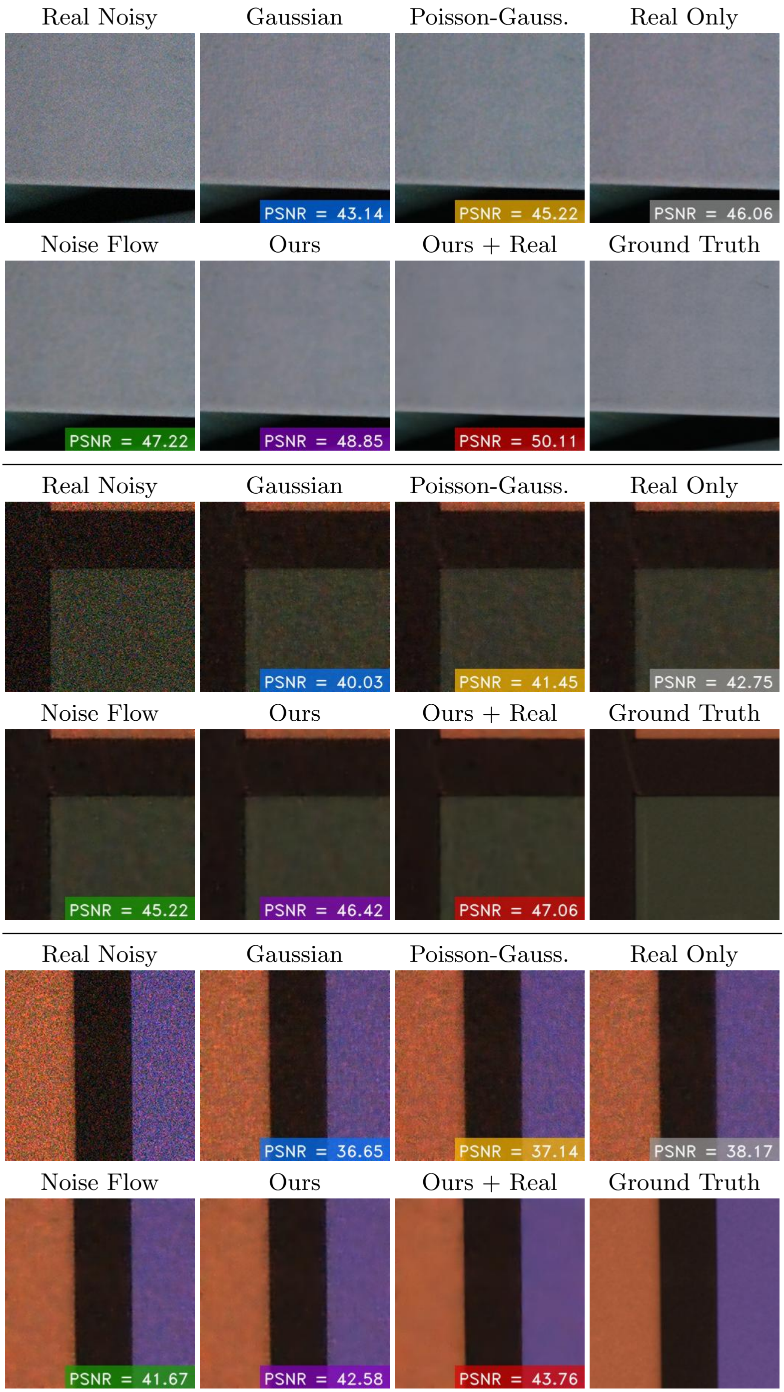

Fig. 6. More Qualitative Results of Denoising (1)

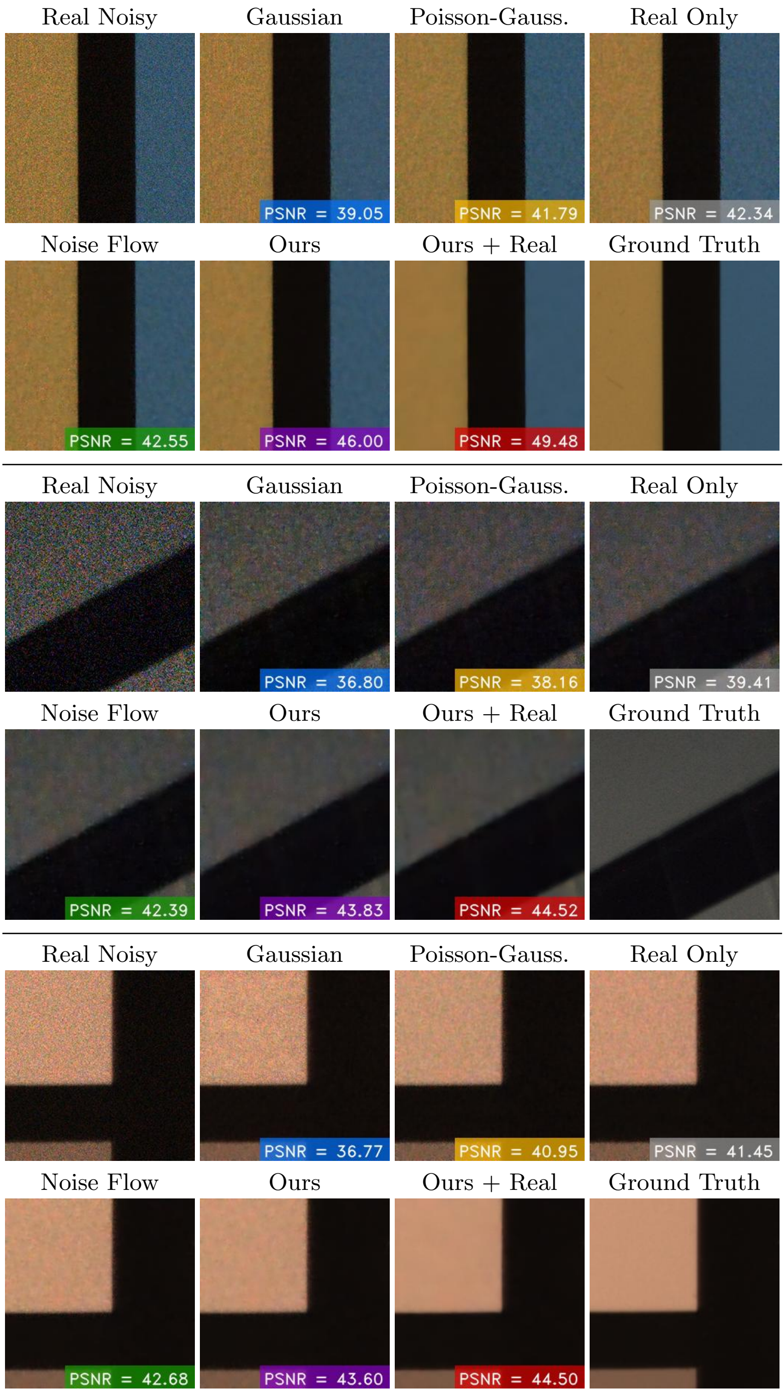

Fig. 7. More Qualitative Results of Denoising (2)

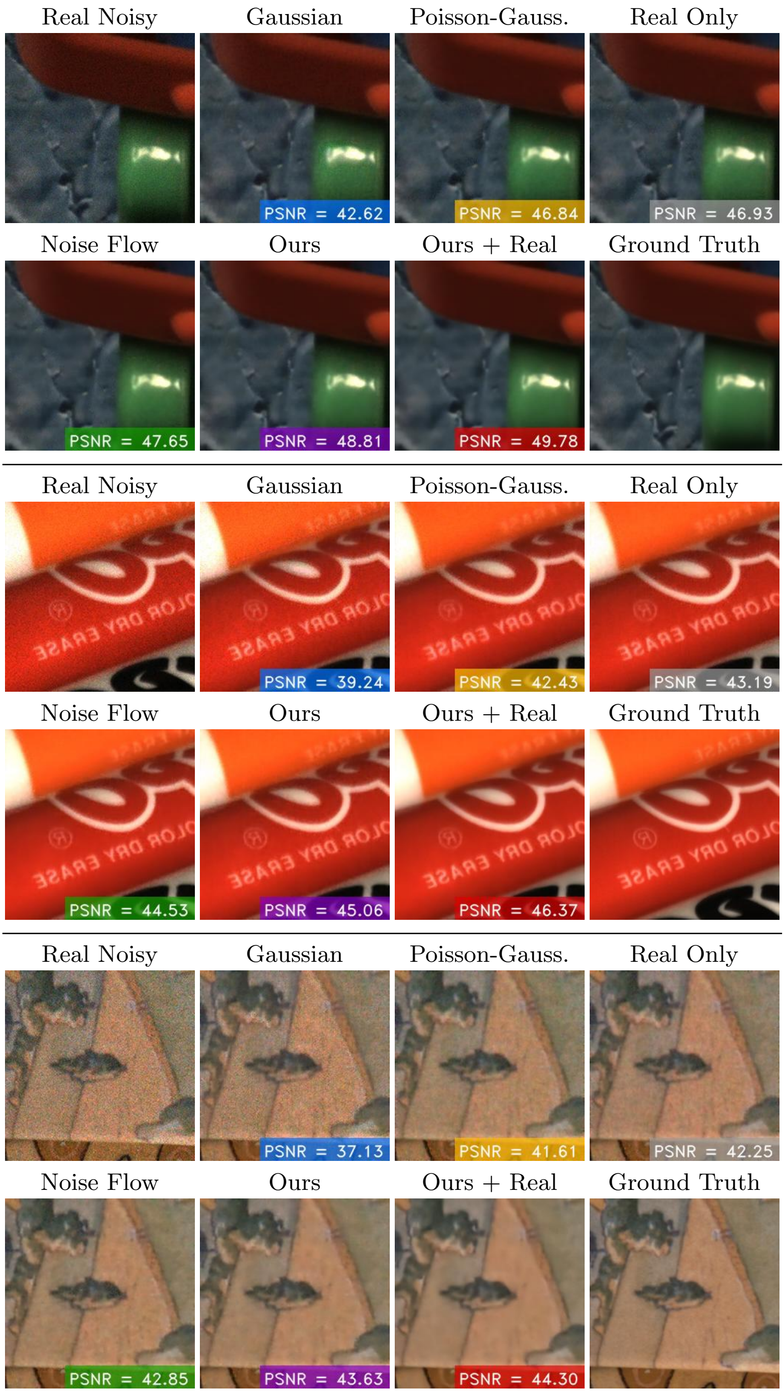

Fig. 8. More Qualitative Results of Denoising (3)

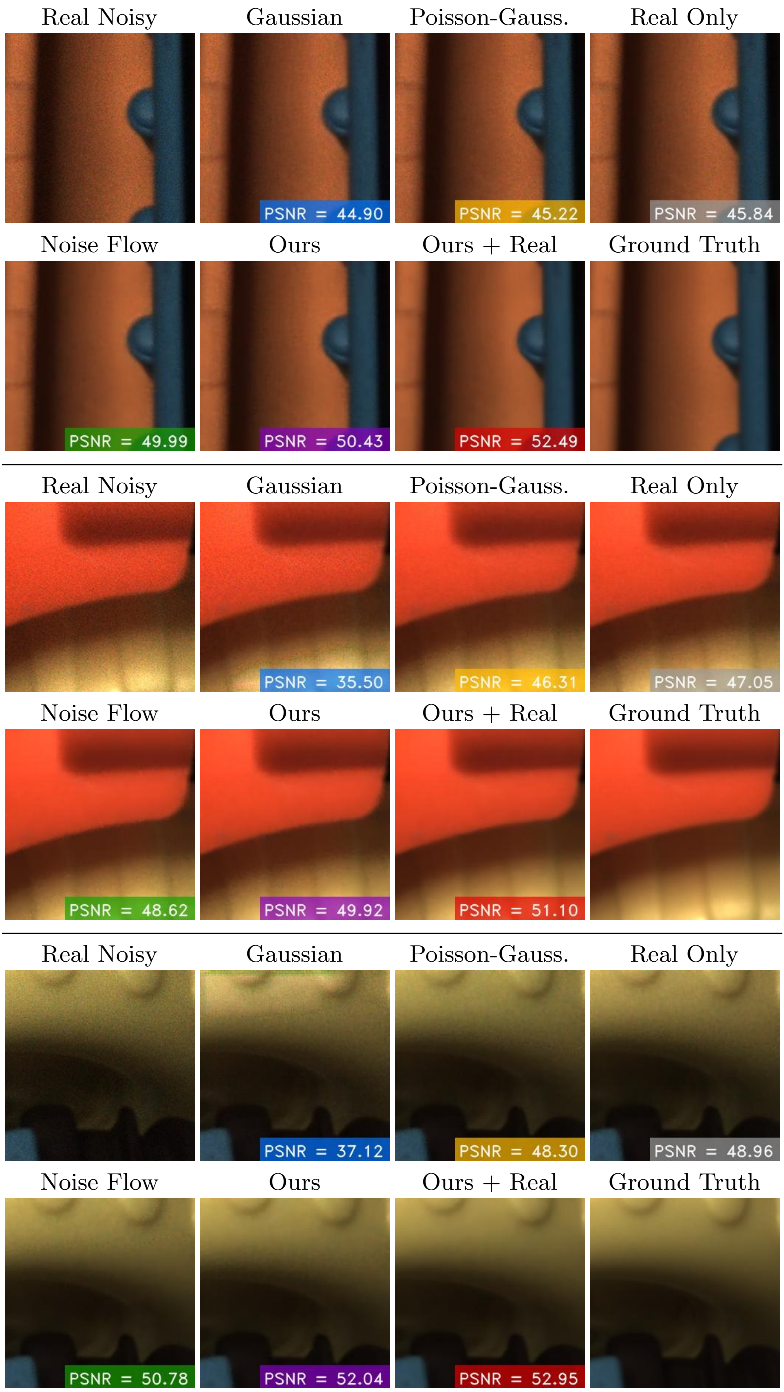

Fig. 9. More Qualitative Results of Denoising (4)

References

1. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. CoRR **abs/1505.04597** (2015)
2. Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2017) 1125–1134
3. Miyato, T., Kataoka, T., Koyama, M., Yoshida, Y.: Spectral normalization for generative adversarial networks. arXiv preprint arXiv:1802.05957 (2018)
4. Ulyanov, D., Vedaldi, A., Lempitsky, V.: Instance normalization: The missing ingredient for fast stylization. arXiv preprint arXiv:1607.08022 (2016)
5. Xu, B., Wang, N., Chen, T., Li, M.: Empirical evaluation of rectified activations in convolutional network. arXiv preprint arXiv:1505.00853 (2015)