Supplementary Material for
Rethinking Deep Image Prior for Denoising

Yeonsik Jo
LG AI Research
yeonsik.jo@lgresearch.ai

Se Young Chun
ECE, INMC, Seoul National University
sychun@snu.ac.kr

Jonghyun Choi
GIST, South Korea
jhc@gist.ac.kr

Note: All blue characters indicate the index of main paper. All red or green characters indicate the reference in this supplementary material.

1. Proof for effective degrees of freedom with ground truth ($\mathbf{x}$) instead of another noisy realization ($\tilde{\mathbf{y}}$) for Eq.(6)

We present a simple proof of the following approximation of $2\sigma^2 \cdot \text{DF}(\mathbf{h})$ to $2\sigma^2 \cdot \text{DF}_{GT}(\mathbf{h})$ (Eq.(6)).

First, the relationship between optimism and effective degrees of freedom in Sec. 3 can be written (combining Eq.(4) and Eq.(5) of the main paper) as:

$$2\sigma^2 \cdot \text{DF}(\mathbf{h}) = \mathbb{E}\left[\mathcal{L}(\tilde{\mathbf{y}}, \mathbf{h}(\cdot)) - \mathcal{L}(\mathbf{y}, \mathbf{h}(\cdot))\right]. \tag{1}$$

Note that $\mathbf{h}(\cdot)$ is an output of the given deep neural network (regardless of the input) where it is fitting to $\mathbf{y}$.

Second, without loss of generality, we model $\tilde{\mathbf{y}} = \mathbf{x} + \tilde{\epsilon}$ where $\tilde{\epsilon}$ is a Gaussian random vector of test data with zero mean and σ standard deviation. Then, we can expand $\mathbb{E}\left[\mathcal{L}(\tilde{\mathbf{y}}, \mathbf{h}(\cdot))\right]$ as follows:

$$\begin{aligned}\mathbb{E}\left[\mathcal{L}(\tilde{\mathbf{y}}, \mathbf{h}(\cdot))\right] &= \mathbb{E}\left[\mathcal{L}(\mathbf{x} + \tilde{\epsilon}, \mathbf{h}(\cdot))\right] \\ &= \mathbb{E}\left[\mathcal{L}(\mathbf{x}, \mathbf{h}(\cdot))\right] + \mathbb{E}\left[\tilde{\epsilon}^T \tilde{\epsilon}\right] + 2\mathbb{E}\left[(\mathbf{x} - \mathbf{h}(\cdot))^T \tilde{\epsilon}\right] \\ &= \mathbb{E}\left[\mathcal{L}(\mathbf{x}, \mathbf{h}(\cdot))\right] + \sigma^2.\end{aligned} \tag{2}$$

Note that $\tilde{\epsilon}$ is independent of $\mathbf{x} - \mathbf{h}(\cdot)$.

Finally, by incorporating Eq.(2) into the first term of Eq.(1) and approximating the asymptotic property by the expectation operation to the deterministic values (between line 2 and 3 in Eq.(3)), we can obtain the following equation:

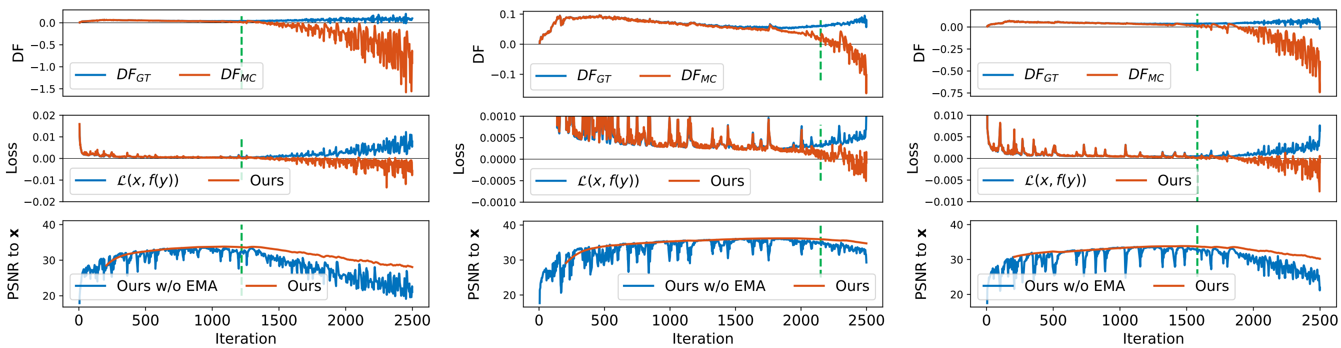
$$\begin{aligned}2\sigma^2 \cdot \text{DF}(\mathbf{h}) &= \mathbb{E}\left[\mathcal{L}(\mathbf{x}, \mathbf{h}(\cdot)) - \mathcal{L}(\mathbf{y}, \mathbf{h}(\cdot)) + \sigma^2\right] \\ &\approx \mathcal{L}(\mathbf{x}, \mathbf{h}(\cdot)) - \mathcal{L}(\mathbf{y}, \mathbf{h}(\cdot)) + \sigma^2 \\ &= 2\sigma^2 \cdot \text{DF}_{GT}(\mathbf{h}).\end{aligned} \tag{3}$$

2. More empirical analyses of the 'zero crossing stopping criterion' with DF$_{GT}$ and DF$_{MC}$.

Fig. 1 shows more examples of DF$_{GT}$ and DF$_{MC}$ with the proposed 'zero crossing stopping' point (green dashed line) (more examples of Fig.3-(c)). We argue that the optimal stopping point, where the 'PSNR to $\mathbf{x}$' is at peak, is roughly aligned with out zero crossing stopping criteria under different noise level and different texture frequency distribution of the images. Fig.3-(c) uses 'Lena' (Fig. 1c), Fig. 1d uses Fig. 1a and Fig. 1e and 1f use Fig. 1b with different σ values. Note that the Image '7' is more colorful than 'Lena' of Fig.3-(c) and Image '10' of Fig. 1b. We observe that, in Image '7', the loss deviated from 0 at earlier iteration (at iteration 1,250 in Fig. 1d) than in 'Lena' (at iteration 1,500 in Fig.3-(c)) and Image '10' with same sigma (at iteration 1,600 in Fig. 1f).

Comparing Fig. 1e and 1f, with different noise level on the same image, we observe that the DF$_{MC}$ with mild noise estimates DF$_{GT}$ well (both are in the similar trajectory) for a slightly longer iteration with small noise ($\sigma = 15$, at iteration 1,850 in top figure of 1e) than when it does with larger noise ($\sigma = 25$, at iteration 1,600 in top figure of 1f). By our zero

(a) Image '7' used in Fig.1-(d) (b) Image '10' used in Fig.1-(e) and (f) (c) 'Lena' used in Fig. 3-(c) of the main paper



(d) Image '7' in Kodak dataset with $\sigma = 25$ (e) Image '10' in Kodak dataset with $\sigma = 15$ (f) Image '10' in Kodak dataset $\sigma = 25$

Figure 1: **More examples of zero-crossing stopping criterion (Fig.3-(c)).** We observe that the optimal stopping point is affected by the intensity of the noise. But still the 'zero-crossing' criterion finds the reasonable point to stop. The green dashed line indicates our proposed stopping point.

crossing stopping criterion, it requires much more iterations to stop with $\sigma = 15$ (at 2,150 iteration in Fig. 1e) than with $\sigma = 25$ (at 1,600 iteration in Fig. 1f). Note that the divergence term in Eq.(8) consists of σ and DF_{MC}. By the zero crossing stopping criterion, we try to monitor when DF_{MC} starts diverging to $-\infty$. But if the σ, is small, then the divergence term, which multiplies the σ to the DF_{MC}, becomes small thus the changes in DF_{MC} are suppressed by the σ as shown in Eq.(8). Since the 'zero' refers to the zero of Eq.(8), when σ is small, the iteration that we 'estimate' when the DF_{MC} start diverging would be delayed as shown in Fig. 1e). Whereas when σ becomes large, the changes in DF_{MC} are amplified by the σ thus the 'zero stopping criterion' senses precisely when DF_{MC} start diverging.

Overall, our 'zero-crossing stopping criterion' (green dashed line in the figures) gives a solution quite close to the optimal stopping points, where the 'PSNR to $\mathbf{x}$' is at peak and even when it misses the optimal points, the denoised images show excellent LPIPS and PSNR scores.

3. Dataset details

CSet9 consists of various sized color images and is used in [2].

Kodak image dataset is widely used in the literature [5, 6, 13]. It contains 24 full color images and the spatial size is 768×512.

McMaster (McM) dataset consists of more saturated image than Kodak to represent digital color images for applications like Color de-mosaicing.

CBSD68 and BSD68 are validation dataset of BSD dataset [1]. They are widely used to evaluate denoising algorithm [3, 7, 11–13].

Set12 is also one of the widely used gray image datasets in denoising literature [11, 13].

4. Network architecture

For a fair comparison to Self2Self, we use the same network architecture to Self2Self [8] with minor modifications; replacing the "Dropout+PConv+LReLU" block with "Conv+Batchnorm+Softplus" since our method does not use dropout and partial convolution (PConv). Note that Soltanayev *et al*. [9] show that the Softplus activation function makes Monte-Carlo estimation more stable. We will release our code in a public repository soon.

5. Comparison to learning-based methods

Although it is not fair to compare our method to learning based methods (our method uses only one noisy observation *without* training while they use a large set of noisy-clean image pairs), it is interesting to compare them as a reference. Specifically, we compare our method with various learning-based methods including DnCNN [11], N2N [5], HQ-N2V [4] and IRCNN [12] on CSet9 dataset with $\sigma = 25$ in Table 1. Note that we use the pre-trained weights of the authors' code to obtain the results of the prior work.

In the table, it is observed that when the size of training data increases, the PSNR increases. In addition, the performance gap in SSIM between ours and other methods is smaller than PSNR as it is computed in the linear scale whereas PSNR uses log scale. Our method exhibits best LPIPS among the all methods despite no training required. We argue that the learning-based methods tend to lose texture details when denoise images while our method is able to preserve delicate textures when it denoise images.

Method	Trainset size	PSNR (↑)	SSIM (↑)	LPIPS (↓)
DnCNN [11]	432	31.17	0.955	0.157
IRCNN [12]	5k	31.72	0.958	0.144
N2N [5]	50k	31.92	**0.960**	0.137
HQ-N2V [4]	50k	**31.98**	**0.960**	0.134
Ours	N/A	31.54	0.953	**0.107**
Ours*	N/A	31.88	**0.960**	0.118

Table 1: **Comparison to learning-based method on CSet9 dataset ($\sigma = 25$).** (↑): higher the better, (↓): lower the better. The best values are in **bold** and second best values are underlined.

6. More qualitative results

Fig. 2 shows examples of the results of BSD68 and Set12 datasets. Fig. 3 shows examples of the results of CBSD68, Set9 and Kodak dataset. In the enlarged portion of the images in Fig. 2 and Fig. 3, it is observed that ours restores the texture better than the others while suppresses the noise (*e.g.*, decent PSNR), which results in much better LPIPS score among the methods with decent PSNR. We argue that our method tries to find better trade-off between PSNR and LPIPS scores.

Specifically, in Fig. 2, all methods except ours fail to preserve wrinkles in elephant. It is partly because other denoising methods have focused on creating an image as smooth as possible thus have high PSNR. In contrast, our method preserves the details as it aims to estimatedly fit to clean images.

In Fig. 3, the wrinkle in the face (the first row) and stripe pattern in the hat (the second row) shows that our method preserves the high frequency textural details while suppresses noise. The rough part of the marble (third row of Fig. 3) is challenging to be recovered by other methods as they tend to oversmooth textures; S2S [8] and CBM3D [2] smooth out this regions (remove some textures). DIP [10] preserves the textures at the expense of resulting in many noisy colored pixels (*i.e.*, pixels colored other than gold). Ours preserves textures relatively better than other methods while suppresses the noises (*i.e.*, not much of pixel colored other than gold).

References

[1] Pablo Arbelaez, Michael Maire, Charless Fowlkes, and Jitendra Malik. Contour detection and hierarchical image segmentation. *IEEE Trans. PAMI*, 2011. 2

[2] Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian. Image denoising by sparse 3d transform-domain collaborative filtering. *IEEE TIP*, 2007. 2, 3

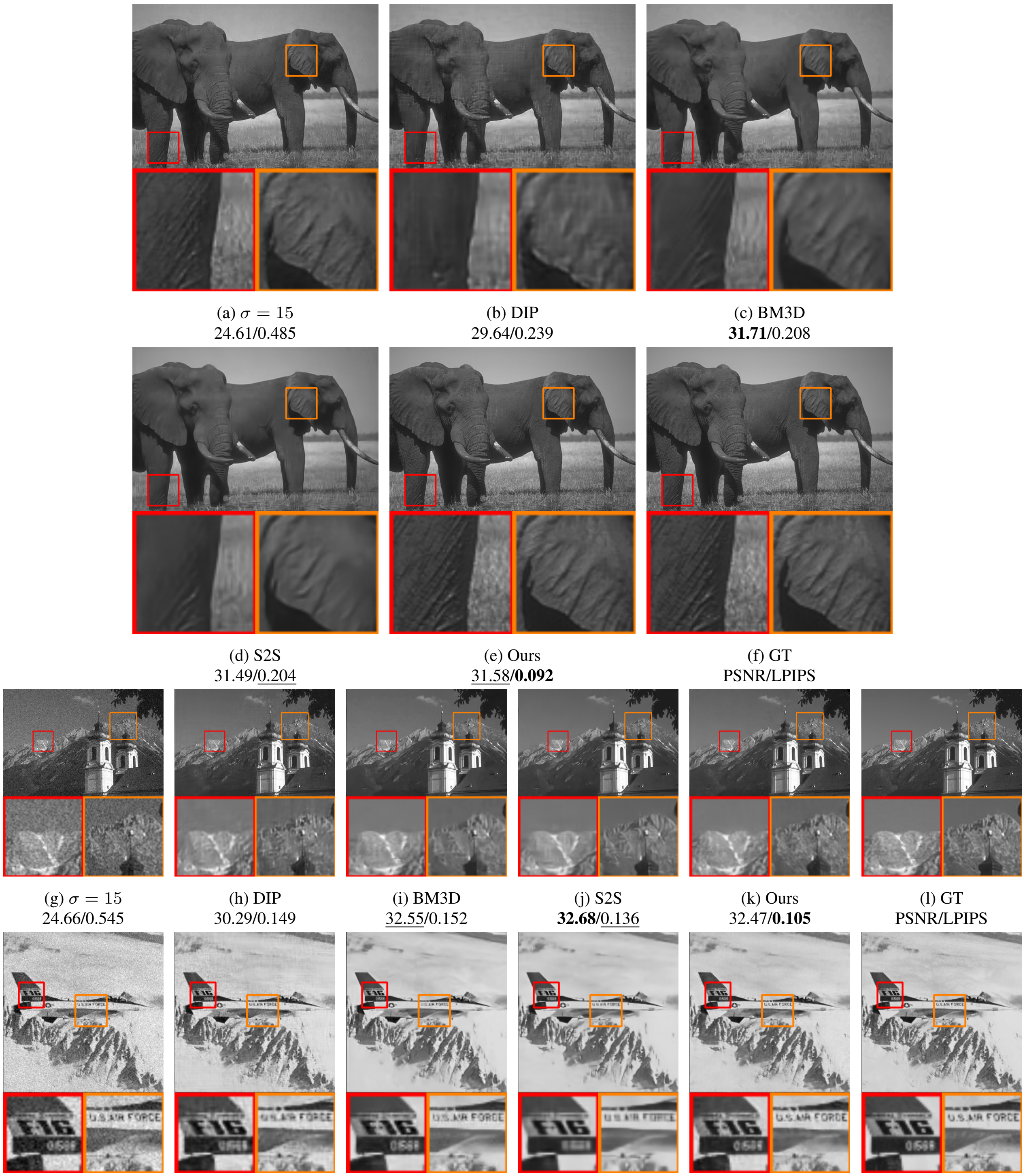
[3] Xixi Jia, Sanyang Liu, Xiangchu Feng, and Lei Zhang. Focnet: A fractional optimal control network for image denoising. In *CVPR*, 2019. 2

[4] Samuli Laine, Tero Karras, Jaakko Lehtinen, and Timo Aila. High-quality self-supervised deep image denoising. In *Neurips*, 2019. 3

[5] 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. 2, 3

[6] Nick Moran, Dan Schmidt, Yu Zhong, and Patrick Coady. Noisier2noise: Learning to denoise from unpaired noisy data. In *CVPR*, 2020. 2

[7] Tobias Plötz and Stefan Roth. Neural nearest neighbors networks. In *NeurIPS*, 2018. 2

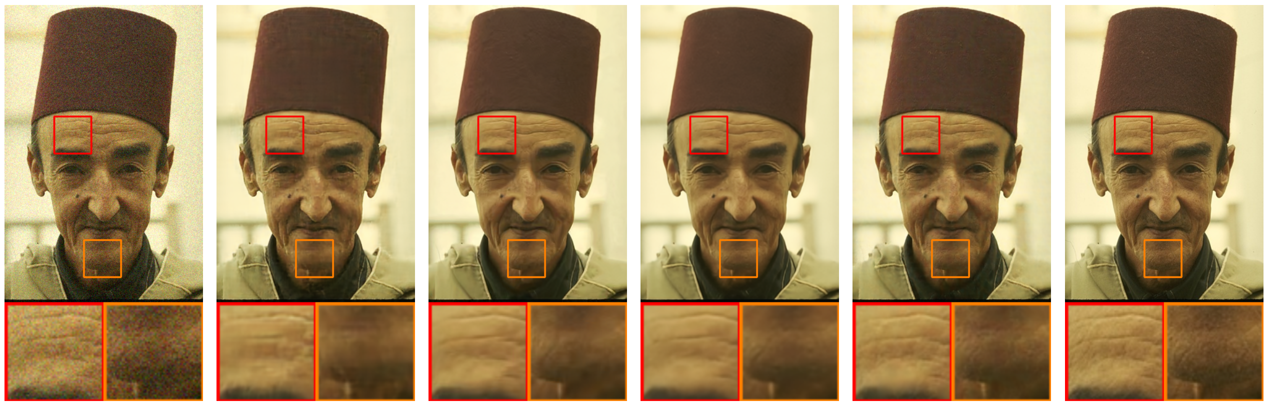


(a) $\sigma = 15$ 24.61/0.485
(b) DIP 29.64/0.239
(c) BM3D **31.71**/0.208
(d) S2S 31.49/0.204
(e) Ours 31.58/**0.092**
(f) GT PSNR/LPIPS
(g) $\sigma = 15$ 24.66/0.545
(h) DIP 30.29/0.149
(i) BM3D 32.55/0.152
(j) S2S **32.68**/0.136
(k) Ours 32.47/**0.105**
(l) GT PSNR/LPIPS
(m) $\sigma = 15$ 24.63/0.393
(n) DIP 28.68/0.130
(o) BM3D 31.05/0.105
(p) S2S 30.21/0.141
(q) Ours **31.09**/**0.052**
(r) GT PSNR/LPIPS

Figure 2: **Qualitative comparison on gray images.** The best performance is in bold. Second best is underlined.

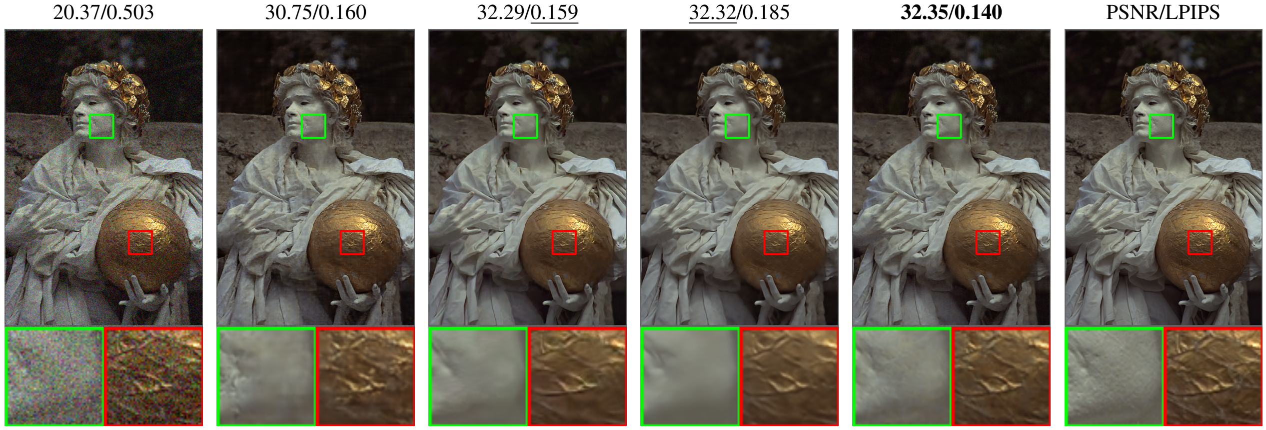
[8] Yuhui Quan, Mingqin Chen, Tongyao Pang, and Hui Ji. Self2self with dropout: Learning self-supervised denoising from single image. In *CVPR*, 2020. 2, 3

[9] S. Soltanayev, R. Giryes, S. Y. Chun, and Y. C. Eldar. On divergence approximations for unsupervised training of deep denoisers



(a) $\sigma = 15$	(b) DIP	(c) CBM3D	(d) S2S	(e) Ours	(f) GT
24.82/0.361	34.61/0.178	**36.40**/<u>0.156</u>	36.02/0.185	35.91/**0.109**	PSNR/LPIPS

(g) $\sigma = 25$	(h) DIP	(i) CBM3D	(j) S2S	(k) Ours	(l) GT
20.37/0.503	30.75/0.160	32.29/<u>0.159</u>	<u>32.32</u>/0.185	**32.35**/**0.140**	PSNR/LPIPS



(m) $\sigma = 25$	(n) DIP	(o) CBM3D	(p) S2S	(q) Ours	(r) GT
20.63/0.745	30.75/0.175	<u>32.92</u>/<u>0.156</u>	**32.96**/0.158	32.49/**0.107**	PSNR/LPIPS

Figure 3: **Qualitative comparison on color images.** The best performance is in bold. Second best is underlined.

based on stein's unbiased risk estimator. In *ICASSP*, 2020. 2

[10] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Deep Image Prior. *IJCV*, 2020. 3

[11] Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang. Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising. *IEEE TIP*, 2017. 2, 3

[12] Kai Zhang, Wangmeng Zuo, Shuhang Gu, and Lei Zhang. Learning deep cnn denoiser prior for image restoration. In *CVPR*, 2017. 2, 3

[13] Kai Zhang, Wangmeng Zuo, and Lei Zhang. Ffdnet: Toward a fast and flexible solution for CNN based image denoising. *IEEE TIP*, 2018. 2