

A Additional experimental details

In this section, we provide experimental details omitted in the main body of the paper.

- *Noise settings.* We simulate 4 different types of noise and 3 intensity levels for each noise type, as detailed below. The generation algorithms follow those of ImageNet-C [14]⁸.
 - Gaussian noise: 0 mean additive Gaussian noise with variance 0.12, 0.18, and 0.26 for low, medium, and high noise levels, respectively;
 - Impulse noise: i.e., salt-and-pepper noise, replacing each pixel with probability $p \in [0, 1]$ into white or black pixel with half chance each. Low, medium, and high noise levels correspond to $p = 0.3, 0.5, 0.7$, respectively;
 - Shot noise: i.e., pixel-wise independent Poisson noise. For each pixel $x \in [0, 1]$, the noisy pixel is Poisson distributed with rate λx , where λ is 25, 12, 5 for low, medium, and high noise levels, respectively.
 - Speckle noise: for each pixel $x \in [0, 1]$, the noisy pixel is $x(1 + \varepsilon)$, where ε is 0-mean Gaussian with a variance level 0.20, 0.35, 0.45 for low, medium, and high noise levels, respectively.
- *Network architecture.* Our AE is based on deep CNNs. The exact architecture is depicted in Table 5.

B Additional experimental results

B.1 Image denoising

Besides DIP, we also verify our methods on DD. As we alluded to in Section 1, although the original DD paper proposes underparameterization as a strategy to tame overfitting, in practice it is tricky to implement and underparameterization can produce inferior results, see, e.g., Fig. 1 (right). Thus, people (including the DD authors in their later papers, e.g., [14]) tend to still use overparametrized DD. Empirically, overparametrized DDs behave very similarly to DIPs. Since in Section 3 we have validated our detection method extensively on DIP with 4 noise types of both low and high noise levels, here we only focus on Gaussian noise with medium noise level. Here, we set all the network width as 512, which is typically used in practice. The learning rate is set to 0.001 here for good numerical stability. Other experimental settings are identical to those of DIP in Section 3.

The denoising results are summarized in Fig. 8. One can observe that in most cases, the detection gap is ≤ 1 in terms of ES-PG, and ≤ 0.1 in terms of ES-SG. However, if we run DD without ES, the overfitting issue is dreadful: most of BASELINE-PGs are ≥ 6 and BASELINE-SGs are ≥ 0.4 . These denoising results reaffirm the effectiveness and generality of our method.

Moreover, Fig. 9 visualizes the reconstruction results of both DIP+AE and DOP. Fig. 9 (left) shows that both DIP+AE and DOP attain similar performance in terms of PSNR while DIP+AE requires far fewer iterations and stops very early. Fig. 9 (right) confirms that visually they also lead to similar reconstruction qualities, as there is almost no perceivable difference.

⁸<https://github.com/hendrycks/robustness>

Table 5: The network architecture of AE.

Nets	Layers	Parameters
Encoder net	Conv2d ¹	(3, 32, 3, 2, 1, False)
	Batch norm, ReLU	N/A
	Conv2d	(32, 64, 3, 2, 1, False)
	Batch norm, ReLU	N/A
	Conv2d	(64, 128, 3, 2, 1, False)
	Batch norm, ReLU	N/A
	Conv2d	(128, 128, 3, 2, 1, False)
	Batch norm, ReLU	N/A
	Conv2d	(128, 128, 3, 2, 1, False)
	Batch norm, ReLU	N/A
	Conv2d	(128, 128, 3, 2, 1, False)
	Batch norm, ReLU	N/A
	Conv2d	(128, 1, 3, 2, 1, False)
	Batch norm, ReLU	N/A
Linear net ³	Linear ²	(16, 16, False)
	Linear	(16, 16, False)
	Linear	(16, 16, False)
	Linear	(16, 16, False)
Decoder net ³	Upsample	bilinear
	Conv2d	(1, 128, 3, 1, 1, False)
	Batch norm, ReLU	N/A
	Upsample	bilinear
	Conv2d	(128, 128, 3, 1, 1, False)
	Batch norm, ReLU	N/A
	Upsample	bilinear
	Conv2d	(128, 128, 3, 1, 1, False)
	Batch norm, ReLU	N/A
	Upsample	bilinear
	Conv2d	(128, 128, 3, 1, 1, False)
	Batch norm, ReLU	N/A
	Upsample	bilinear
	Conv2d	(128, 64, 3, 1, 1, False)
	Batch norm, ReLU	N/A
	Upsample	bilinear
	Conv2d	(64, 32, 3, 1, 1, False)
	Batch norm, ReLU	N/A
	Upsample	bilinear
	Conv2d	(32, 3, 3, 1, 1, False)
	Batch norm, Sigmoid	N/A

¹ The parameters for Conv2d layers: (in_channels, out_channels, kernel_size, stride, padding, bias).

² The parameters for Linear layers: (in_features, out_features, bias).

³ Tensors are reshaped properly to suit the input dimensions.

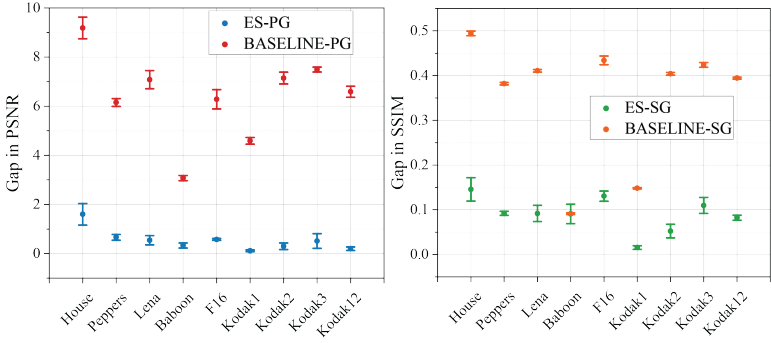


Figure 8: DD+AE for image denoising. (left) The performance measured in PGs. (right) The performance measured in SGs.

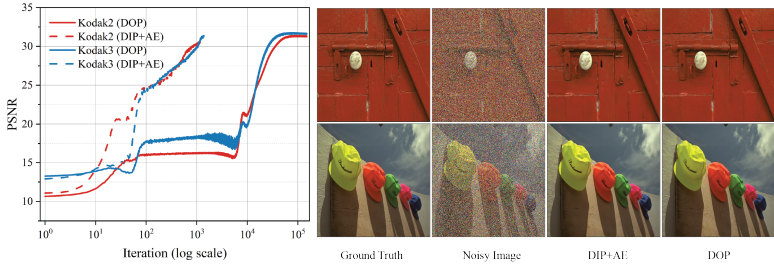


Figure 9: DIP+AE vs DOP. (left) The solid lines and dash lines respectively show the restoration performance for DOP and DIP+AE. (right) Visualizations for Kodak2 (first row) and Kodak3 (second row).

B.2 MRI reconstruction

Here we provide the results for the complete set of random MRI samples that we experiment with. For the notations, *ES* indicates the reconstruction quality detected by our method, *Peak* denotes the peak quality that DD can achieve, and *Overfitting* is the final reconstruction quality without ES. As can be seen from Table 6, for all experimental samples, *ES* and *Peak*

Table 6: The experimental results of MRI reconstruction.

		Sample 1	Sample 2	Sample 4	Sample 6	Sample 9	Sample 10	Sample 16
PSNR \uparrow	ES	29.837 (0.176)	30.077 (0.231)	27.419 (0.250)	28.799 (0.111)	33.395 (0.158)	27.907 (0.124)	28.602 (0.217)
	Peak	31.792 (0.295)	31.700 (0.185)	28.465 (0.282)	29.855 (0.081)	34.444 (0.265)	28.816 (0.107)	30.494 (0.381)
	Overfitting	27.182 (0.379)	26.710 (0.449)	23.934 (0.072)	25.631 (0.191)	31.053 (0.061)	25.545 (0.182)	24.806 (0.232)
SSIM \uparrow	ES	0.602 (0.002)	0.609 (0.000)	0.611 (0.003)	0.612 (0.005)	0.669 (0.002)	0.620 (0.008)	0.646 (0.005)
	Peak	0.642 (0.006)	0.643 (0.004)	0.658 (0.003)	0.637 (0.014)	0.689 (0.008)	0.644 (0.004)	0.671 (0.003)
	Overfitting	0.562 (0.002)	0.571 (0.002)	0.568 (0.005)	0.507 (0.012)	0.591 (0.009)	0.554 (0.004)	0.553 (0.005)

yield close performance in terms of both PSNR and SSIM, which indicates that our method can reliably detect the near-peak performance. On the other hand, the performance (both

PSNR and SSIM) of *ES* is uniformly better than that of *overfitting*, often by a considerable margin, further confirming the effectiveness of our method.

B.3 Image inpainting

Image inpainting is another common IR task that SIDGPs have excelled in and hence popularly evaluated on; see, e.g., the DIP paper [41]. In this task, a clean image x_0 is contaminated by additive Gaussian noise ε , and then only partially observed to yield the observation $y = (x_0 + \varepsilon) \odot m$, where $m \in \{0, 1\}^{H \times W}$ is a binary mask and \odot denotes the Hadamard point-wise product. Here both y and m are known, and the goal is to reconstruct x_0 . We consider the natural formulation

$$E(x) = \|(x - y) \odot m\|_2^2. \quad (4)$$

We parametrize x using the DIP model. The mask m is generated according to an iid Bernoulli model, with a rate of 50%, i.e., 50% of pixels not observed in expectation. The noise ε is set to the medium level.

We also compare the performance of our method (DIP+AE) with the two competing methods: DIP+TV [46, 47] and SGLD [6]. To ensure convergence, we run 60K and 200K iterations for DIP+TV and SGLD, respectively, and report their final results. We repeat all experiments 3 times and obtain the mean and standard deviation for each instance.

Table 7 summarizes the results. Our method significantly outperforms the other two in terms of both PSNR and SSIM. It should be noted that here we test a medium noise level, rather than the very low noise levels experimented with in the DIP+TV and SGLD papers. Although in their original papers overfitting seems to be gone, here we see a strike-back with a different noise level. So the two competing methods are at best sensitive to hyperparameters, which are tricky to set.

Table 7: DIP+AE, DIP+TA, and SGLD for image inpainting. The best PSNRs are colored as red; the best SSIMs are colored as blue.

	PSNR \uparrow			SSIM \uparrow		
	DIP+AE	DIP+TV	SGLD	DIP+AE	DIP+TV	SGLD
Barbara	21.878 (0.101)	17.790 (0.024)	15.326 (0.062)	0.522 (0.010)	0.259 (0.000)	0.259 (0.003)
Boat	23.268 (0.445)	18.071 (0.040)	15.211 (0.020)	0.534 (0.010)	0.259 (0.001)	0.227 (0.001)
House	28.883 (0.300)	18.362 (0.022)	15.566 (0.059)	0.767 (0.025)	0.157 (0.000)	0.171 (0.002)
Lena	25.052 (0.246)	18.264 (0.040)	15.481 (0.084)	0.644 (0.004)	0.218 (0.001)	0.204 (0.003)
Peppers	26.251 (0.187)	18.400 (0.022)	15.610 (0.036)	0.738 (0.016)	0.203 (0.000)	0.199 (0.001)
C.man	26.194 (0.423)	18.571 (0.049)	15.861 (0.031)	0.732 (0.009)	0.204 (0.001)	0.215 (0.001)
Couple	22.619 (0.154)	18.115 (0.007)	15.313 (0.062)	0.512 (0.011)	0.280 (0.000)	0.241 (0.002)
Finger	21.396 (0.119)	17.714 (0.024)	15.150 (0.027)	0.795 (0.003)	0.601 (0.000)	0.490 (0.001)
Hill	24.216 (0.254)	18.274 (0.017)	15.514 (0.107)	0.518 (0.009)	0.242 (0.000)	0.214 (0.003)
Man	23.687 (0.302)	18.159 (0.022)	15.394 (0.109)	0.532 (0.007)	0.252 (0.001)	0.222 (0.004)
Montage	27.290 (0.282)	19.005 (0.022)	16.334 (0.017)	0.799 (0.007)	0.193 (0.000)	0.221 (0.001)

B.4 Performance on clean images

For SIDGPS, when there is no noise, the target clean image is a global optimizer to Eq. (2). So there is no overfitting issue in these scenarios, and ES is not strictly necessary. But, in practice, one does not know if noise is present apriori, and finite termination has to be made. In this section, we experiment with “denoising” *clean* images with DIP. The setup is exactly as that of our typical denoising, except that here we do not report the PSNR gap, as whenever one makes a stop after finite iterations, the theoretical PSNR gap is infinity. We report the absolute PSNR detected by our method instead; for most applications, PSNR greater than 30 is good enough for practical purposes.

The AE error curve tends to fluctuate when the quality is already high. To improve the detection performance, we find that comparing running average to determine ES points performs better than the stopping criterion described in Algorithm 1. We use this slightly modified version here; we leave reconciling the two versions as future work—we suspect that this modified version will likely improve our previous detection performance.

Table 8 shows the preliminary results. Except for the challenging case Baboon, the PSNR scores are near 30 or above. So our method is performing reasonable detection. We suspect using more advanced smoothing techniques such that Gaussian smoothing can suppress the fluctuation better and hence lead to better performance; we leave this as future work.

Table 8: Performance of DIP+AE on denoising clean images.

	PSNR \uparrow	SSIM \uparrow
House	36.569	0.921
Peppers	30.407	0.797
Lena	31.927	0.857
Baboon	20.186	0.423
F16	33.065	0.911
Kodak1	29.243	0.852
Kodak2	31.064	0.826
Kodak3	30.155	0.861
Kodak12	31.757	0.851

B.5 Bell-shape examples under different learning rate

As we shown in Table 4, our ES detection method is stable in terms of different learning rates of DIP. Here we further demonstrate that the bell-shape of PSNR curve of DIP is holds under different learning rates. We randomly select two images—F16 and Peppers—and visualize their PSNR curves under different learning rates $\{0.01, 0.001, 0.0001\}$ in Fig. 10. We can observe that different rates would perturb the curves but would not distort the overall bell shape.

B.6 Analysis of failure mode

To qualitatively understand the failing cases, we select 3 positive images that enjoy consistent good detection and 3 negative images that see consistent failure, and visualize their Fourier spectra in Figs. 11 and 12. For better visualization, we take the $x \mapsto \log(1+x)$ transform of the Fourier magnitudes, as is commonly done in image processing. Visually, the positive

Table 9: The performance gaps of BRISQUE [23], NIQE [24], NIMA [33], and DIP+AE on shot and speckle noises. For NIMA, we report both technical quality assessment (the number before “/”) and aesthetic assessment (the number after “/”). The best PSNR gaps are colored as red; the best SSIM gaps are colored as blue.

	Shot noise								Speckle noise							
	Gap in PSNR ↓				Gap in SSIM ↓				Gap in PSNR ↓				Gap in SSIM ↓			
	BRISQUE	NIQE	NIMA	DIP+AE	BRISQUE	NIQE	NIMA	DIP+AE	BRISQUE	NIQE	NIMA	DIP+AE	BRISQUE	NIQE	NIMA	DIP+AE
House	6.713	8.629	10.873/0.662	0.294	0.389	0.491	0.598/0.024	0.002	9.848	8.970	12.879/ 1.394	1.847	0.457	0.424	0.591/0.027	0.010
Peppers	5.538	5.975	1.863/6.013	0.417	0.267	0.289	0.128/0.295	0.018	6.414	6.085	8.987/4.861	0.311	0.233	0.227	0.316/0.193	0.011
Lena	7.976	6.191	9.697/1.545	1.281	0.448	0.375	0.516/0.075	0.020	8.797	5.133	9.352/0.912	0.445	0.402	0.240	0.436/0.039	0.013
Baboon	0.508	0.562	2.767/3.061	1.920	0.026	0.009	0.391/0.404	0.284	0.387	1.507	1.178/2.063	2.318	0.027	0.053	0.144/0.231	0.314
F16	5.329	8.448	0.938 /6.404	1.016	0.403	0.555	0.011 /0.168	0.013	6.760	7.136	0.418 /7.757	0.418	0.488	0.498	0.001 /0.229	0.021
Kodak1	3.099	3.741	3.287/5.387	0.953	0.086	0.118	0.321/0.485	0.059	1.363	4.332	3.215/6.871	0.719	0.013	0.106	0.258/0.556	0.056
Kodak2	15.693	9.055	10.208/1.071	0.154	0.433	0.269	0.435/0.025	0.006	9.755	7.921	8.169/0.634	0.340	0.272	0.177	0.158/ 0.013	0.013
Kodak3	8.429	8.546	2.501/21.796	0.895	0.429	0.427	0.092/0.619	0.006	6.484	6.509	13.457/13.130	1.183	0.236	0.239	0.584/0.263	0.016
Kodak12	6.009	8.941	6.746/20.054	2.118	0.401	0.490	0.422/0.704	0.021	8.545	8.204	5.398/0.703	0.562	0.459	0.450	0.343/0.011	0.010

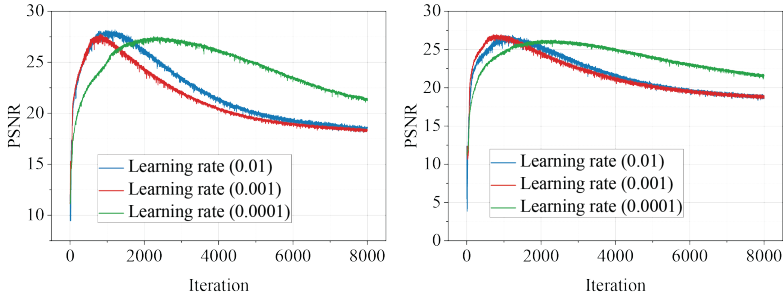


Figure 10: The PSNR curves of DIP under different learning rates. (left) the PSNR curves for F16; (right) the PSNR curves for Peppers.

examples can be well characterized as being piecewise smooth, and the negative ones invariably contain fine details that correspond to high frequency components. Indeed, the positive spectra are concentrated in the low-frequency bands, whereas the negative spectra are much more scattered into high-frequency bands. We leave a more quantitative analysis of this as future work.

B.7 ES criterion based on other principles?

During the peer review, one reviewer kindly pointed to us the possibility of formulating ES criteria based on the *whiteness* or *discrepancy* principles in image denoising [11, 14]. In this section, we briefly discuss the possibility of implementing them. We want to quickly remind that we target practical denoising, and so assume very little knowledge about noise types, levels, or whatsoever. Particularly, the noise level σ^2 is unknown to us and possibly also hard to estimate due to the generality we strive for. Also, to avoid confusion, we will switch our notations slightly here.

Let $\mathbf{X} \in \mathbb{R}^{n \times n}$ be the target image, and $\mathbf{Y} = f(\mathbf{X}) + \mathbf{W}$ be the noisy measurement, where

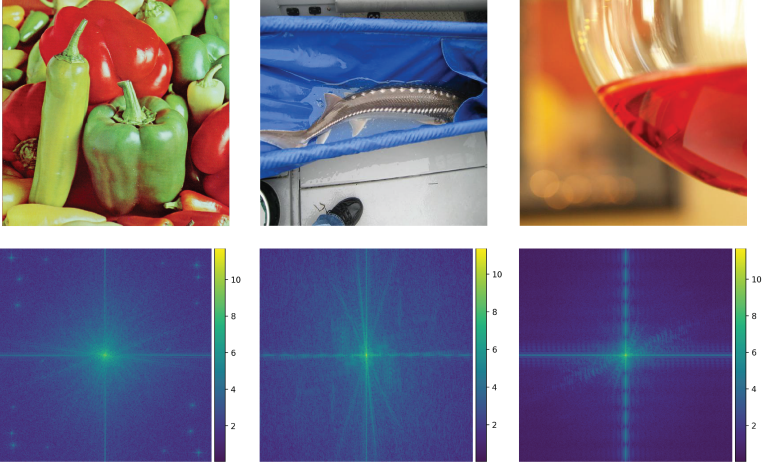


Figure 11: Positive images and their spectra.

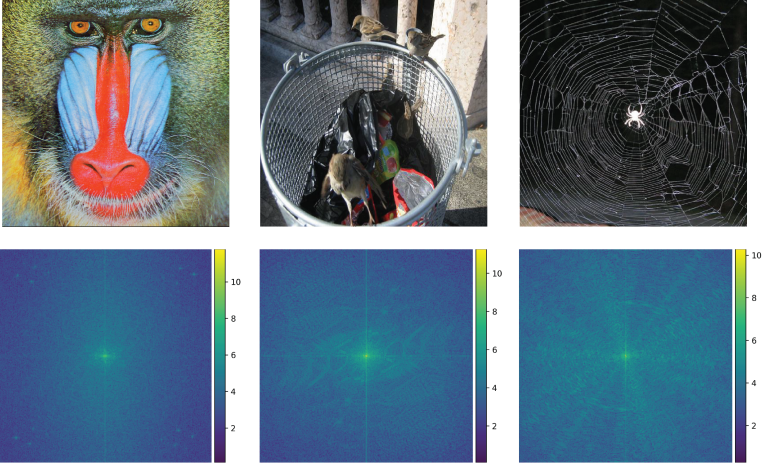


Figure 12: Negative images and their spectra.

$\mathbf{W} \in \mathbb{R}^{n \times n}$ is white noise with variance level $\sigma^2 < +\infty$, i.e.

- $\mathbb{E} w_{ij} = 0$ and $\mathbb{E} w_{ij}^2 = \sigma^2 \quad \forall i, j$;
- any pair of distinct elements in \mathbf{W} are *uncorrelated*: for two different (in location) elements w, w' of \mathbf{W} , $\mathbb{E}[(w - \mathbb{E} w)(w' - \mathbb{E} w')] = 0 \implies \mathbb{E}[ww'] = 0$ (the implication uses the 1st property).

The 2nd property has an interesting implication:

$$\mathbb{E}[\mathbf{W} \star \mathbf{W}] = n^2 \sigma^2 \delta_{n \times n} = \begin{bmatrix} n^2 \sigma^2 & \mathbf{0}_{n-1}^\top \\ \mathbf{0}_{n-1} & \mathbf{0}_{(n-1) \times (n-1)} \end{bmatrix}, \quad (5)$$

where \star denotes the 2D cross-correlation (for convenience, we assume the circular version), and $\delta_{n \times n}$ is the 2D delta function (we assume the top-left corner corresponds to no-shift alignment). Denote our estimated image as $\hat{\mathbf{X}}$. If we have perfect recovery, then

$$\mathbf{Y} - f(\hat{\mathbf{X}}) = \mathbf{Y} - f(\mathbf{X}) = \mathbf{W}, \quad (6)$$

and so

$$\mathbb{E} \left[\left(\mathbf{Y} - f(\hat{\mathbf{X}}) \right) \star \left(\mathbf{Y} - f(\hat{\mathbf{X}}) \right) \right] = \mathbb{E} [\mathbf{W} \star \mathbf{W}] = n^2 \sigma^2 \delta_{n \times n}. \quad (7)$$

This is the *whiteness principle* in [19]⁹: in particular, except for the top-left corner, all other elements should be zero.

In practice, we are not able to take the expectation as we only observe one realization. But note that except for the perfectly aligned case which produces $n^2 \sigma^2$, each element of $\frac{1}{n^2} \mathbf{W} \star \mathbf{W}$ can be approximated by $\mathcal{N}\left(0, \frac{\sigma^4}{n^2}\right)$ due to central limit theorem when n^2 is large—this is valid for images. So typical element of $\mathbf{W} \star \mathbf{W}$ should lie in the range

$$[-c n \sigma^2, c n \sigma^2], \quad (8)$$

where $c > 1$ is a large enough constant, say 5.

This is explicitly used as a constraint in [19] to improve image restoration quality. If the variance level σ^2 is known, we may be able to use distance to this set as a measure of image quality. However, for our applications, we do not assume known σ^2 , and also the constant c chosen can impact the result also.

A more reasonable metric for us would be the quantity

$$\|(\mathbf{W} \star \mathbf{W})_{-0}\|, \quad (9)$$

i.e., the norm of the $\mathbf{W} \star \mathbf{W}$ with the top-left corner removed, which ideally should be as sufficiently small. To be more explicit,

$$\left\| \left(\left(\mathbf{Y} - f(\hat{\mathbf{X}}) \right) \star \left(\mathbf{Y} - f(\hat{\mathbf{X}}) \right) \right)_{-0} \right\|. \quad (10)$$

But obviously the minimum is achieved when we overfit the noisy image \mathbf{Y} .

The *discrepancy principle* (or local constraint) in [10] is a refinement to the obvious constraint

$$\frac{1}{n^2} \left\| f(\hat{\mathbf{X}}) - \mathbf{Y} \right\|_F^2 \leq \sigma^2 \quad (11)$$

for a good estimate $\hat{\mathbf{X}}$ to satisfy. This only enforces that *globally* the noise level of the residual matches the known noise level, but does not ensure *uniformly*. To ensure the latter, a natural idea is to enforce the noise level consistently everywhere locally:

$$G * \left(f(\hat{\mathbf{X}}) - \mathbf{Y} \right)_{i,j}^2 \leq \sigma^2 \quad \forall i, j. \quad (12)$$

⁹Our presentation here does not use the discrete-time stochastic process language as in the original paper—which seems overly technical than necessary, but they are equivalent.

Here, G is a Gaussian filter of appropriate size, and the left side of the equality is the (Gaussian) weighted mean variance around the pixel location (i, j) .

Similar to the situation for the whiteness principle, if the variance level σ^2 is known, we may also use the distance to this set as a metric to measure the reconstruction quality. When σ^2 is unknown, it is unclear how to make easy modification to do this, unlike the case of the whiteness principle above.