Supplementary Material for
Self-Calibrated Variance-Stabilizing Transformations for Real-World Image Denoising

A. More results on removal of synthetic and real-world noise

	Method	KODAK	BSD300	SET14
Gaussian $\sigma = 25$	Baseline, N2C [29]	32.91/0.896	31.33/0.892	31.93/0.878
	CBM3D [5]	31.87/0.868	30.48/0.861	30.88/0.854
	N2V [13]	31.81/0.875	30.52/0.870	30.53/0.853
	Nr2N [17]	30.45/0.811	29.34/0.803	29.75/0.815
	ZS-N2N [16]	29.87/0.797	28.93/0.800	28.97/0.789
	S2S [20]	31.28/0.864	29.86/0.849	30.08/0.839
	DBSN [23]	31.64/0.856	29.80/0.839	30.63/0.846
	SSDN [14]	32.40/0.883	30.99/0.877	31.36/0.866
	R2R [18]	32.25/0.880	30.91/0.872	31.32/0.865
	NBR2NBR [11]	32.08/0.879	30.79/0.873	31.09/0.864
	B2UNB [22]	32.27/0.880	30.87/0.872	31.27/0.864
	DCD-Net [32]	32.27/0.881	31.01/0.876	31.29/0.862
	SST-GP [24]	32.75/**0.898**	31.18/0.880	31.68/0.872
	Noise2VST (ours)	**32.85**/0.895	**31.28/0.891**	**31.82/0.874**
	Noise2VST† (ours)	32.83/0.891	31.26/0.884	31.81/0.872
Gaussian $\sigma \in [5, 50]$	Baseline, N2C [29]	33.08/0.887	31.40/0.871	32.41/0.883
	CBM3D [6]	32.02/0.860	30.56/0.847	30.94/0.849
	N2V [13]	31.72/0.863	30.39/0.855	30.24/0.843
	Nr2N [17]	32.17/0.868	30.93/0.862	30.87/0.852
	S2S [20]	31.37/0.860	29.87/0.841	29.97/0.849
	DBSN [23]	30.38/0.826	28.34/0.788	29.49/0.814
	SSDN [14]	32.40/0.870	30.95/0.861	31.21/0.855
	R2R [18]	31.50/0.850	30.56/0.855	30.84/0.850
	NBR2NBR [11]	32.10/0.870	30.73/0.861	31.05/0.858
	B2UNB [22]	32.34/0.872	30.86/0.861	31.14/0.857
	DCD-Net [32]	32.35/0.872	31.09/0.866	31.09/0.855
	SST-GP [24]	31.78/0.880	31.12/0.869	31.38/0.871
	Noise2VST (ours)	**32.93/0.882**	**31.27**/0.869	**32.09**/0.872
	Noise2VST† (ours)	32.88/0.881	31.24/**0.870**	32.08/**0.873**

	Method	KODAK	BSD300	SET14
Poisson $\lambda = 30$	Baseline, N2C + GAT [15, 29]	32.22/0.884	30.59/0.872	31.07/0.865
	CBM3D + GAT [6, 15]	30.53/0.856	29.18/0.842	29.44/0.837
	N2V [13]	31.18/0.864	29.88/0.858	29.79/0.841
	Nr2N [17]	29.43/0.775	28.29/0.764	28.63/0.778
	ZS-N2N [16]	29.06/0.775	28.15/0.782	27.97/0.763
	S2S [20]	30.31/0.857	28.93/0.840	28.84/0.839
	DBSN	30.07/0.827	28.19/0.790	29.16/0.814
	SSDN [14]	31.67/0.874	30.25/0.866	30.47/0.855
	R2R [18]	30.50/0.801	29.47/0.811	29.53/0.801
	NBR2NBR [11]	31.44/0.870	30.10/0.863	30.29/0.853
	B2UNB [22]	31.64/0.871	30.25/0.862	30.46/0.852
	DCD-Net [32]	31.60/0.870	30.22/0.865	30.41/0.855
	SST-GP [24]	31.99/0.879	**30.84/0.897**	30.87/0.867
	Noise2VST (ours)	**32.19/0.883**	30.55/0.871	**30.99**/0.860
	Noise2VST† (ours)	32.13/0.882	30.52/0.871	30.98/**0.861**
Poisson $\lambda \in [5, 50]$	Baseline, N2C + GAT [15, 29]	31.63/0.865	29.92/0.850	30.66/0.854
	CBM3D + GAT [5, 15]	29.40/0.836	28.22/0.815	28.51/0.817
	N2V [13]	30.55/0.844	29.46/0.844	29.44/0.831
	Nr2N [17]	30.31/0.812	29.45/0.821	29.40/0.812
	S2S [20]	29.06/0.834	28.15/0.817	28.83/0.841
	DBSN	29.60/0.811	27.81/0.771	28.72/0.800
	SSDN [14]	30.88/0.850	29.57/0.841	28.94/0.808
	R2R [18]	29.14/0.732	28.68/0.771	28.77/0.765
	NBR2NBR [11]	30.86/0.855	29.54/0.843	29.79/0.838
	B2UNB [22]	31.07/0.857	29.92/0.852	30.10/0.844
	DCD-Net [32]	31.00/0.857	**29.99/0.855**	29.99/0.843
	SST-GP [24]	31.39/**0.872**	29.96/0.853	30.22/0.848
	Noise2VST (ours)	**31.60**/0.865	29.89/0.849	**30.60/0.850**
	Noise2VST† (ours)	31.51/0.862	29.84/0.848	30.57/**0.850**

Table 1. PSNR(dB)/SSIM denoising results on synthetic datasets in sRGB space. The highest PSNR(dB)/SSIM among unsupervised denoising methods is highlighted in **bold**, while the second is underlined.

Method	BM3D [5]	DBSN [23]	N2V [13]	B2UNB [22]	FBI-D (DND) [4]	**Noise2VST** (ours)
DND	47.53/0.976	45.41/0.969	45.87/0.964	45.87/0.964	47.53/0.971	**47.94/0.978**

Table 2. PSNR(dB)/SSIM denoising results on Darmstadt Noise Dataset (DND) raw data [19].

B. Display of the learned VST

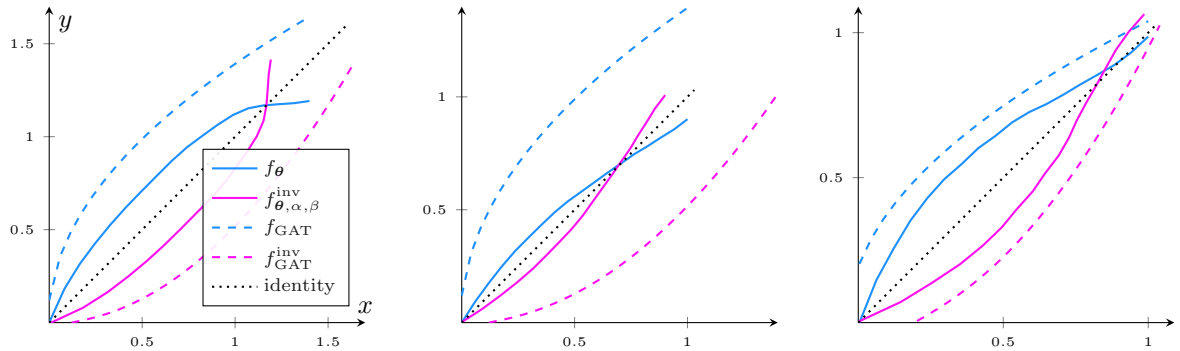

Fig2a Fig2b Fig2c

Figure 1. GAT versus learned VST by the proposed algorithm for Figure 2 of the paper.

C. Ablation study on the number of parameters

Figure 2 presents the results of an ablation study on the number of learnable parameters for Noise2VST. As expected, increasing the number of parameters leads to improved PSNR, though the performance gains begin to plateau beyond a hundred parameters. Our choice of 128 (+ 2) parameters is thus appropriate.

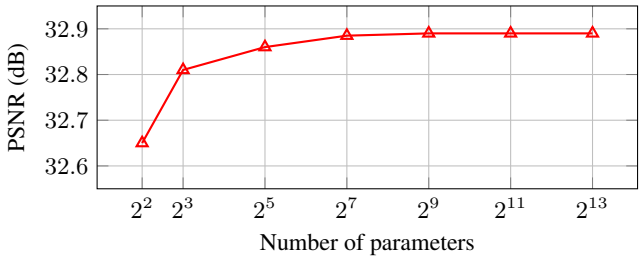

Figure 2. PSNR (in dB) vs. number of parameters (*i.e.*, number of knots for the spline $f_{\boldsymbol{\theta}}$) on the testing set Confocal Fish [30].

D. Why can the noise level σ and the parameter θ_1 be fixed?

Let D be a non-blind denoiser, assumed to be normalization-equivariant [10]:

$$\forall(\boldsymbol{z},\sigma)\in\mathbb{R}^N\times\mathbb{R}_{>0},\forall\lambda\in\mathbb{R}_{>0},\forall\mu\in\mathbb{R},\quad D(\lambda\boldsymbol{z}+\mu\mathbf{1},\lambda\sigma)=\lambda D(\boldsymbol{z},\sigma)+\mu\mathbf{1}\,, \tag{1}$$

Note that a denoiser D can always be constrained to be normalization-equivariant by considering the application $\mathcal{T}^{\text{inv}}\circ D\circ\mathcal{T}$ with:

$$\mathcal{T}:(\boldsymbol{z},\sigma)\mapsto\left(\frac{\boldsymbol{z}-\boldsymbol{z}_{\min}}{\boldsymbol{z}_{\max}-\boldsymbol{z}_{\min}},\frac{\sigma}{\boldsymbol{z}_{\max}-\boldsymbol{z}_{\min}}\right)\text{ and }\mathcal{T}^{\text{inv}}:\boldsymbol{z}\mapsto(\boldsymbol{z}_{\max}-\boldsymbol{z}_{\min})\boldsymbol{z}+\boldsymbol{z}_{\min}\,, \tag{2}$$

(see elements of proof in [10]). The following proposition states that the noise level σ can be arbitrarily set to $\sigma_0=25/255$ and the parameter θ_1 to 0 without loss of generality.

Proposition 1. *Let $\boldsymbol{z}\in\mathbb{R}^N,\boldsymbol{\theta}\in\mathbb{R}^n,\alpha,\beta\in\mathbb{R}$ and $\sigma>0$. There exists $\alpha',\beta'\in\mathbb{R}$ and $\boldsymbol{\theta'}\in\mathbb{R}^n$ with $\theta'_1=0$ such that:*

$$f^{\text{inv}}_{\boldsymbol{\theta},\alpha,\beta}(D(f_{\boldsymbol{\theta}}(\boldsymbol{z}),\sigma))=f^{\text{inv}}_{\boldsymbol{\theta'},\alpha',\beta'}(D(f_{\boldsymbol{\theta'}}(\boldsymbol{z}),\sigma_0))\,.$$

Proof. Let $\boldsymbol{z}\in\mathbb{R}^N,\boldsymbol{\theta}\in\mathbb{R}^n,\alpha,\beta\in\mathbb{R}$ and $\sigma>0$. By normalization-equivariance of the denoiser D:

$$f^{\text{inv}}_{\boldsymbol{\theta},\alpha,\beta}(D(f_{\boldsymbol{\theta}}(\boldsymbol{z}),\sigma))=f^{\text{inv}}_{\boldsymbol{\theta},\alpha,\beta}(\frac{\sigma}{\sigma_0}D(\frac{\sigma_0}{\sigma}f_{\boldsymbol{\theta}}(\boldsymbol{z})-\frac{\sigma_0}{\sigma}f_{\boldsymbol{\theta}}(\boldsymbol{z}_{\min})\mathbf{1},\sigma_0)+f_{\boldsymbol{\theta}}(\boldsymbol{z}_{\min})\mathbf{1})\,.$$

But it is easy to see that there exists $\boldsymbol{\theta'}\in\mathbb{R}^n$ with $\theta'_1=0$ such that:

$$f_{\boldsymbol{\theta'}}(\boldsymbol{z})=\frac{\sigma_0}{\sigma}f_{\boldsymbol{\theta}}(\boldsymbol{z})-\frac{\sigma_0}{\sigma}f_{\boldsymbol{\theta}}(\boldsymbol{z}_{\min})\mathbf{1}\,.$$

Its algebraic inverse is:

$$f^{-1}_{\boldsymbol{\theta'}}(\boldsymbol{z})=f^{-1}_{\boldsymbol{\theta}}(\frac{\sigma}{\sigma_0}\boldsymbol{z}+f_{\boldsymbol{\theta}}(\boldsymbol{z}_{\min})\mathbf{1}),$$

and, for $\alpha'=\alpha\dfrac{\sigma}{\sigma_0}$ and $\beta'=\alpha f_{\boldsymbol{\theta}}(\boldsymbol{z}_{\min})+\beta$, we have:

$$f^{\text{inv}}_{\boldsymbol{\theta'},\alpha',\beta'}(\boldsymbol{z})=f^{-1}_{\boldsymbol{\theta'}}(\boldsymbol{z})+\alpha'\boldsymbol{z}+\beta'\mathbf{1}=f^{-1}_{\boldsymbol{\theta}}(\frac{\sigma}{\sigma_0}\boldsymbol{z}+f_{\boldsymbol{\theta}}(\boldsymbol{z}_{\min})\mathbf{1})+\alpha'\boldsymbol{z}+\beta'\mathbf{1}=f^{\text{inv}}_{\boldsymbol{\theta},\alpha,\beta}(\frac{\sigma}{\sigma_0}\boldsymbol{z}+f_{\boldsymbol{\theta}}(\boldsymbol{z}_{\min})\mathbf{1})\,.$$

Finally, $f^{\text{inv}}_{\boldsymbol{\theta},\alpha,\beta}(D(f_{\boldsymbol{\theta}}(\boldsymbol{z}),\sigma))=f^{\text{inv}}_{\boldsymbol{\theta'},\alpha',\beta'}(D(f_{\boldsymbol{\theta'}}(\boldsymbol{z}),\sigma_0))\,.$

□

Table 3 experimentally confirms what was theoretically proven: the noise level σ and the parameter θ_1 can be fixed during training.

θ_1 \ σ	$\sigma = 15/255$	$\sigma = 25/255$	$\sigma = 50/255$	Learned
$\theta_1 = 0$	32.882	32.882	32.880	32.878
Learned	32.880	32.883	32.879	32.881

Table 3. PSNR(dB) results of Noise2VST with different combinations of θ_1 and σ on the testing set Confocal Fish.

E. Choice of the Gaussian denoiser

E.1. Blind denoisers are helpless for real-world noise

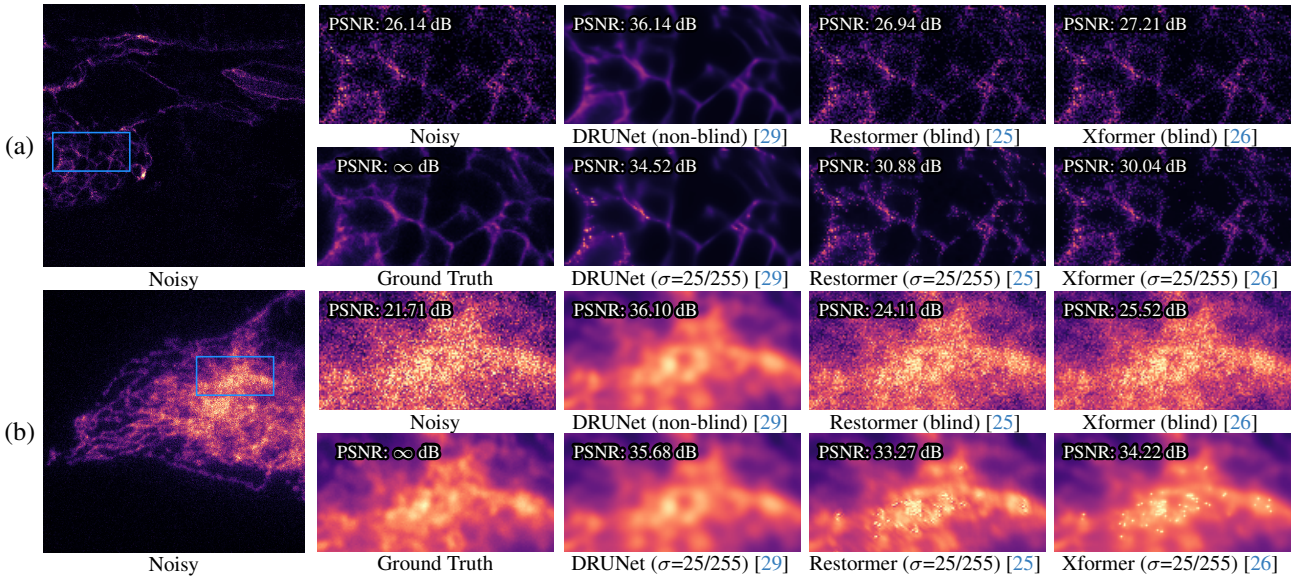

Figure 3. Qualitative real-world image denoising results within the proposed Noise2VST framework for different types of neural networks. (a): image from FMD dataset [30], (b): image from W2S dataset [31]. Only the non-blind denoising network (*i.e.,* with a noise level map as additional input [28, 29]) is able to process the noise correctly, a phenomenon already observed in [28] for real-world noise. The pre-trained weights are taken from [10, 25, 26, 29].

E.2. Noise2VST can handle a wide range of (non-blind) denoisers

Non-blind Gaussian denoisers are recommended within our framework due to their better generalization abilities. We tested our method with three others:

- FDnCNN: the unpublished non-blind version of DnCNN,
- GS-DRUNet [12]: a "gradient step" denoiser,
- NL-Ridge [9]: a "traditional" denoiser written in PyTorch (thus supporting automatic differentiation).

Table 4 presents the experimental results, which highlight the ability of Noise2VST to handle a wide range of denoisers.

Training \ Inference	Same as training	DRUNet
DRUNet [29]	32.88	32.88
FFDNet [28]	32.62	32.88
FDnCNN [27]	32.59	32.88
GS-DRUNet [12]	32.87	32.88
NL-Ridge [9]	31.42	32.71

Table 4. PSNR (in dB) results of Noise2VST with different combinations of Gaussian denoisers on the testing set Confocal Fish [30].

F. How does the learned VST stabilize the variance in the case of synthetic Poisson noise?

In this experiment, we artificially generate noisy images by adding Poisson noise to clean images. The noise model is:

$$z_i = \mathcal{P}(\lambda s_i)/\lambda$$

where λ controls to level of noise. Giving the information of λ to GAT [15], we know that that the variance of the random variable $f_{\text{GAT}}(z_i)$ is approximately constant, whatever the value of s_i. But what about the learned VST $f_{\boldsymbol{\theta}}$? Figure 4 shows that, even though the learned VST is less precise than GAT in stabilizing variance, they do achieve almost the same performance for image denoising in the end. This surprising phenomenon can be attributed to the fact that powerful denoisers such as [28, 29] can adapt to a noise level which was slightly over or under-estimated, without much consequences with regard to the outcome. Of course, it is recommended to use GAT for synthetic Poisson noise over the learned VST when the oracle parameter λ is given. However, as shown in the paper, the proposed method finds its true interest in the case of real noise.

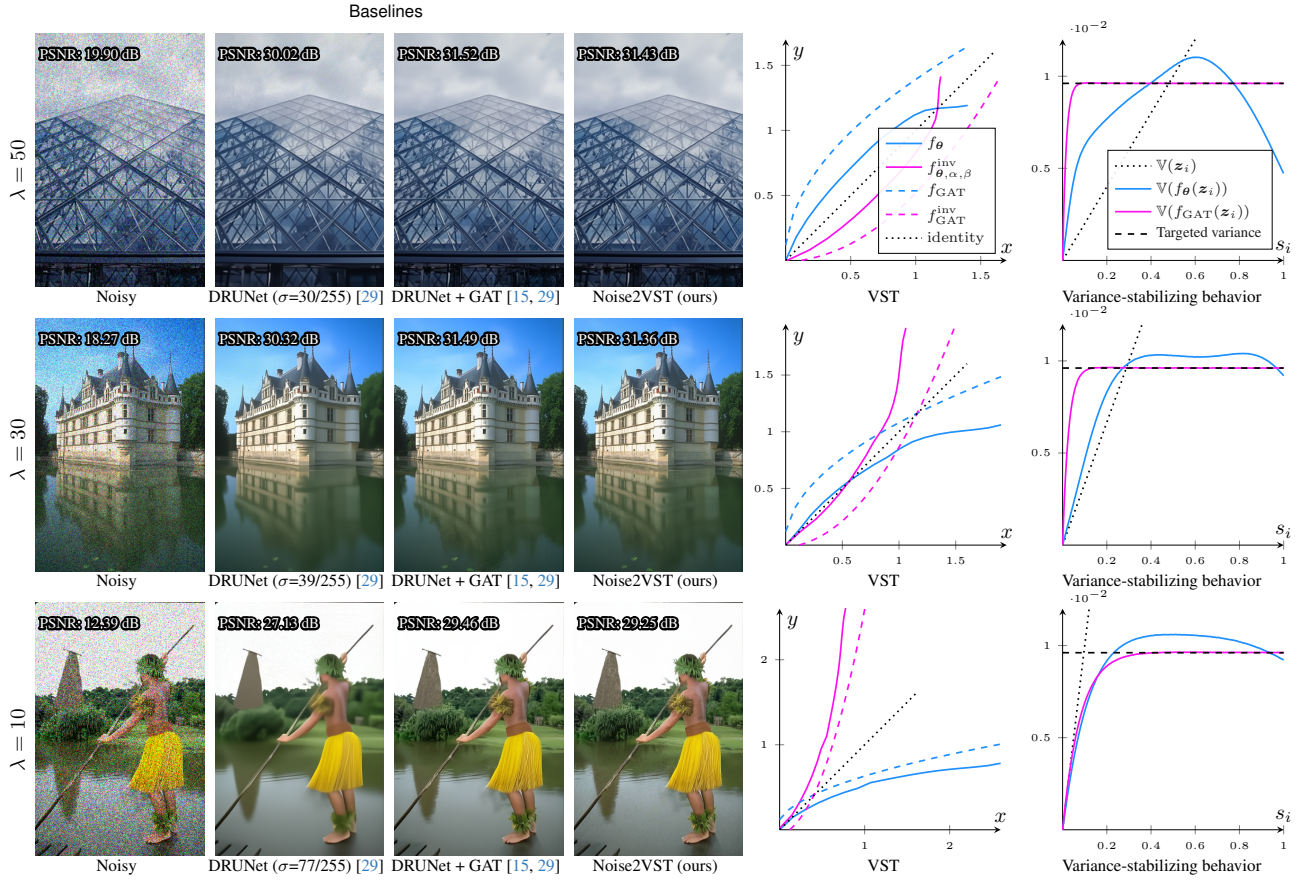

Figure 4. GAT versus learned VST for synthetic Poisson noise. The learned transform $f_{\boldsymbol{\theta}}$ is less precise than f_{GAT} in stabilizing the variance of synthetic Poisson noise with parameter λ (assuming that GAT knows the oracle parameter λ) but the consequences in terms of denoising performance are very limited. As a baseline, we also compare with the case where no VST is applied (the best noise level σ is shown in each case).

G. How does the learned VST stabilize the variance in the case of synthetic Rayleigh noise?

In this experiment, we artificially generate noisy images by adding Rayleigh noise to clean images. The noise model is:

$$z_i = \sqrt{\frac{2}{\pi}}\,\mathrm{Rayleigh}(s_i)\,, \tag{3}$$

where the multiplicative constant is added to guarantee that $\mathbb{E}(z_i) = s_i$ (zero-mean noise). Interestingly, there exists transformations that stabilize exactly the variance, namely:

$$f_{\text{exact}} : z \mapsto \frac{\sqrt{24}}{\pi}\log(z) + c\,. \tag{4}$$

where $c \in \mathbb{R}$ can be any constant. This is because the log-Rayleigh distribution [21] has a variance of $\pi^2/24$, which does not depend on its parameter s_i. Note however that, contrary to mixed Poisson-Gaussian noise, $f_{\text{exact}}(z_i)$ is not close to a normal distribution. The exact unbiased inverse [15] is given by the implicit mapping $\mathbb{E}[f_{\text{exact}}(z)|s] \mapsto s$, where z denotes the random variable following (3). According to [21], $\mathbb{E}[f_{\text{exact}}(z)|s] = \frac{\sqrt{24}}{\pi}(\log(s) + (\log(2) + \log(2/\pi) - \gamma)/2) + c$, hence:

$$f^{\text{inv}}_{\text{exact}} : z \mapsto \exp\left(\frac{\gamma - \log(2) - \log(2/\pi)}{2}\right)\exp\left(\frac{\pi}{\sqrt{24}}(z - c)\right)\,. \tag{5}$$

where γ is the Euler constant defined by $-\int_0^{+\infty}\log(x)\exp(-x)dx$.

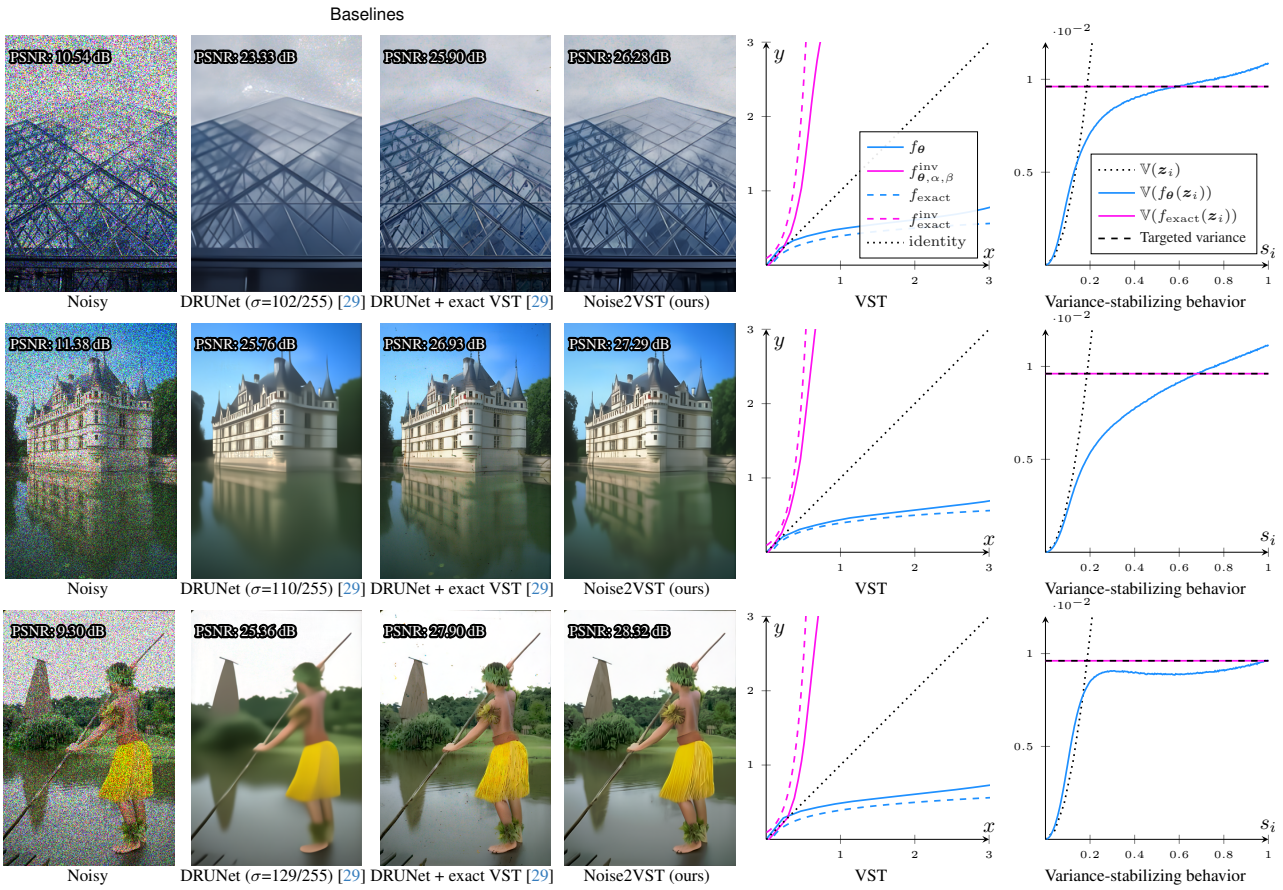

Figure 5. Exact versus learned VST for synthetic Rayleigh noise. The learned transform $f_{\boldsymbol{\theta}}$ is less precise than f_{exact} in stabilizing the variance of synthetic Rayleigh noise but it actually gives slightly better results. As a baseline, we also compare with the case where no VST is applied (the best noise level σ is shown in each case).

H. On the benefit of constraining the inverse transform to be closed to the algebraic one

	Inverse VST	Kodak [8] (Poisson λ=30)	FMD [30] Confocal Fish	SIDD [1] Validation
$g_{\boldsymbol{\theta}}$	(unconstrained)	31.68/0.866	**32.91/0.906**	50.92/0.991
$f_{\boldsymbol{\theta}}^{-1}$	(algebraic inverse)	31.95/0.878	32.37/0.896	51.59/0.991
$f_{\boldsymbol{\theta},\alpha,\beta}^{\text{inv}}$	(proposed)	**32.19/0.883**	32.88/0.905	**51.66/0.992**

Table 5. Ablation study: PSNR(dB)/SSIM denoising results of Noise2VST depending on the nature of the inverse transform.

I. On the benefit of a model-free VST

In the proposed framework, the VST is searched among the family of continuous piecewise linear (CPWL) functions which allows high flexibility. But what if the VST is searched among the family of the generalized Anscombe transforms (GAT) instead, parameterized by only two positive parameters a and b? More precisely, the direct and inverse transforms, parameterized by a and b, are given by:

$$f_{\text{GAT}} : z \mapsto \frac{2}{a}\sqrt{\max\left(az + \frac{3}{8}a^2 + b, 0\right)}\,, \tag{6}$$

and

$$f_{\text{GAT}}^{\text{inv}} : z \mapsto a\left(\frac{1}{4}z^2 - \frac{1}{8} + \frac{1}{4}\sqrt{\frac{3}{2}}z^{-1} - \frac{11}{8}z^{-2} + \frac{5}{8}\sqrt{\frac{3}{2}}z^{-3}\right) - \frac{b}{a}\,, \tag{7}$$

respectively. Note that the inverse transform $f_{\text{GAT}}^{\text{inv}}$ is the closed-form approximation of the unbiased inverse transform recommended by [15]. Table 6 shows that it is possible to use the proposed framework to find the GAT parameters in the case of synthetic noise. However, in real-world noise conditions, the proposed model-free VST achieves better results than the learned model-based VST, which assumes a mixed Poisson-Gaussian model, perhaps not entirely accurate. Nevertheless, it is worth noticing that the estimation of parameters a and b within our framework is still better than the parameters provided by the authors of the FMD dataset [30] or the ones estimated by the zero-shot method [7] for W2S dataset [31] (see Table 2 and 3 of the paper). Finally, note that for real-noise experiments, we utilized the raw W2S dataset [31] rather than the normalized version provided by the authors, as normalization can alter the noise distribution in a way that may negatively impact GAT.

	Model for VST	Number of parameters	Kodak [8] (Poisson λ=30)	FMD [30] Confocal Fish	FMD [30] Confocal Mice	W2S ch0 [31] *avg1*
f_{GAT}	(model-based)	2	**32.19/0.883**	32.78/0.897	38.23/**0.965**	35.04
$f_{\boldsymbol{\theta}}$	(model-free)	130	**32.19/0.883**	**32.88/0.905**	**38.27/0.965**	**35.65**

Table 6. Ablation study: PSNR(dB)/SSIM denoising results of Noise2VST depending on the model for the VST.

J. Limitations

There are obviously no performance guarantees when the noise properties deviate from our assumptions, namely the noise is *zero-mean*, *spatially independent* and *Gaussianizable* (*i.e.* there exists a transformation from the input noise distribution to a Gaussian-like one). Figure 6 shows two artificial (and non-realistic) examples in which Noise2VST encounters difficulties due to the out-of-distribution nature of the noise. We notice that the spatial independence of noise appears to be the most crucial property of all. However, we would like to emphasize that the noise assumptions underlying Noise2VST are already quite broad and have proven to be sufficient for most applications, especially in fluorescence microscopy. Furthermore, these are the same assumptions commonly made in the literature, aligning with works such as Noise2Void [13], Noise2Self [3], and their variants [11, 16, 18, 20, 22].
Of course, this is not speficic to Noise2VST. For example, the traditional Anscombe transform [2] also has troubles when the noise assumption (namely Poisson noise) does not align with the input noise distribution (see Fig. 7).

References

[1] Abdelrahman Abdelhamed, Stephen Lin, and Michael S. Brown. A high-quality denoising dataset for smartphone cameras. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 6

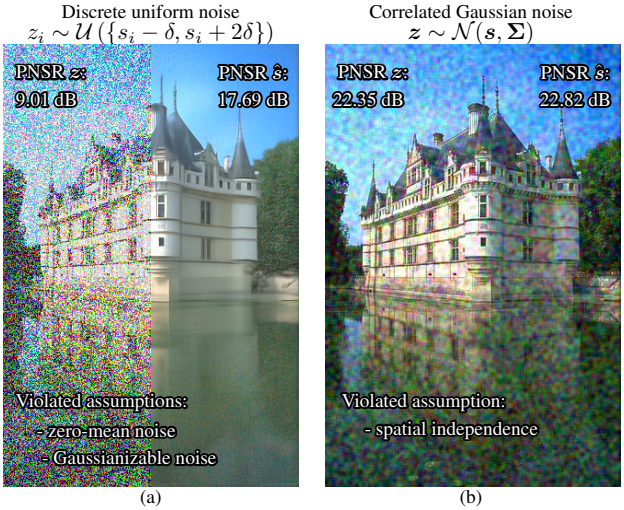

Figure 6. Examples of out-of-distribution noises for Noise2VST.

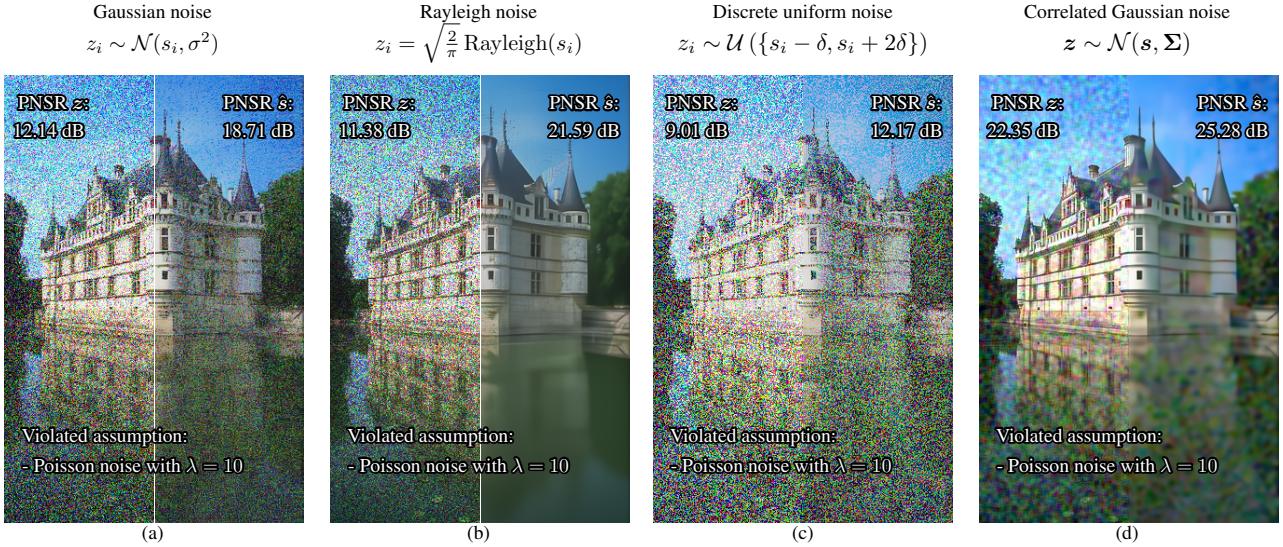

Figure 7. Examples of out-of-distribution noises for a model-based VST (Anscombe transform [2]).

[2] Francis J. Anscombe. The transformation of Poisson, binomial and negative-binomial data. *Biometrika*, 35(3/4):246–254, 1948. 6, 7

[3] Joshua Batson and Loic Royer. Noise2Self: Blind denoising by self-supervision. In *International Conference on Machine Learning (ICML)*, pages 524–533, 2019. 6

[4] Jaeseok Byun, Sungmin Cha, and Taesup Moon. FBI-denoiser: Fast blind image denoiser for Poisson-Gaussian noise. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5768–5777, 2021. 1

[5] Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian. Image denoising by sparse 3-D transform-domain collaborative filtering. *IEEE Transactions on Image Processing*, 16(8):2080–2095, 2007. 1

[6] Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian. Color image denoising via sparse 3D collaborative filtering with grouping constraint in luminance-chrominance space. In *IEEE International Conference on Image Processing (ICIP)*, pages I–313. IEEE, 2007. 1

[7] Alessandro Foi, Mejdi Trimeche, Vladimir Katkovnik, and Karen Egiazarian. Practical Poissonian-Gaussian noise modeling and fitting for single-image raw-data. *IEEE Transactions on Image Processing*, 17(10):1737–1754, 2008. 6

[8] Rich Franzen. Kodak lossless true color image suite. *source: http://r0k. us/graphics/kodak*, 4(2):9, 1999. 6

[9] Sébastien Herbreteau and Charles Kervrann. Towards a unified view of unsupervised non-local methods for image denoising: The NL-Ridge approach. In *IEEE International Conference on Image Processing (ICIP)*, pages 3376–3380, 2022. 3

[10] Sébastien Herbreteau, Emmanuel Moebel, and Charles Kervrann. Normalization-equivariant neural networks with application to image denoising. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 5706–5728, 2023. 2, 3

[11] Tao Huang, Songjiang Li, Xu Jia, Huchuan Lu, and Jianzhuang Liu. Neighbor2Neighbor: Self-supervised denoising from single noisy images. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14781–14790, 2021. 1, 6

[12] S. Hurault, A. Leclaire, and N. Papadakis. Gradient step denoiser for convergent plug-and-play. In *International Conference on Learning Representations (ICLR)*, 2022. 3

[13] Alexander Krull, Tim-Oliver Buchholz, and Florian Jug. Noise2Void - Learning denoising from single noisy images. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2124–2132, 2019. 1, 6

[14] Samuli Laine, Tero Karras, Jaakko Lehtinen, and Timo Aila. High-quality self-supervised deep image denoising. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019. 1

[15] Markku Makitalo and Alessandro Foi. Optimal inversion of the generalized Anscombe transformation for Poisson-Gaussian noise. *IEEE Transactions on Image Processing*, 22(1):91–103, 2012. 1, 4, 5, 6

[16] Youssef Mansour and Reinhard Heckel. Zero-Shot Noise2Noise: Efficient image denoising without any data. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14018–14027, 2023. 1, 6

[17] Nick Moran, Dan Schmidt, Yu Zhong, and Patrick Coady. Noisier2Noise: Learning to denoise from unpaired noisy data. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12061–12069, 2020. 1

[18] Tongyao Pang, Huan Zheng, Yuhui Quan, and Hui Ji. Recorrupted-to-Recorrupted: Unsupervised deep learning for image denoising. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2043–2052, 2021. 1, 6

[19] Tobias Plötz and Stefan Roth. Benchmarking denoising algorithms with real photographs. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2750–2759, 2017. 1

[20] Yuhui Quan, Mingqin Chen, Tongyao Pang, and Hui Ji. Self2Self with dropout: Learning self-supervised denoising from single image. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1887–1895, 2020. 1, 6

[21] Bertrand Rivet, Laurent Girin, and Christian Jutten. Log-Rayleigh distribution: A simple and efficient statistical representation of log-spectral coefficients. *IEEE Transactions on Audio, Speech, and Language Processing*, 15(3):796–802, 2007. 5

[22] Zejin Wang, Jiazheng Liu, Guoqing Li, and Hua Han. Blind2Unblind: Self-supervised image denoising with visible blind spots. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2027–2036, 2022. 1, 6

[23] Xiaohe Wu, Ming Liu, Yue Cao, Dongwei Ren, and Wangmeng Zuo. Unpaired learning of deep image denoising. In *European Conference on Computer Vision (ECCV)*, pages 352–368, 2020. 1

[24] Rajeev Yasarla, Jeya Maria Jose Valanarasu, Vishwanath Sindagi, and Vishal M. Patel. Self-supervised denoising transformer with Gaussian process. In *IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1474–1484, 2024. 1

[25] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Ming–Hsuan Yang. Restormer: Efficient transformer for high-resolution image restoration. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5718–5729, 2022. 3

[26] Jiale Zhang, Yulun Zhang, Jinjin Gu, Jiahua Dong, Linghe Kong, and Xiaokang Yang. Xformer: Hybrid X-shaped transformer for image denoising. In *International Conference on Learning Representations (ICLR)*, 2024. 3

[27] Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang. Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising. *IEEE Transactions on Image Processing*, 26(7):3142–3155, 2017. 3

[28] Kai Zhang, Wangmeng Zuo, and Lei Zhang. FFDNet: Toward a fast and flexible solution for CNN-based image denoising. *IEEE Transactions on Image Processing*, 27(9):4608–4622, 2018. 3, 4

[29] Kai Zhang, Yawei Li, Wangmeng Zuo, Lei Zhang, Luc Van Gool, and Radu Timofte. Plug-and-Play image restoration with deep denoiser prior. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(10):6360–6376, 2022. 1, 3, 4, 5

[30] Yide Zhang, Yinhao Zhu, Evan Nichols, Qingfei Wang, Siyuan Zhang, Cody Smith, and Scott Howard. A Poisson-Gaussian denoising dataset with real fluorescence microscopy images. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11710–11718, 2019. 2, 3, 6

[31] Ruofan Zhou, Majed El Helou, Daniel Sage, Thierry Laroche, Arne Seitz, and Sabine Süsstrunk. W2S: Microscopy data with joint denoising and super-resolution for widefield to SIM mapping. In *European Conference on Computer Vision Workshops (ECCVW)*, 2020. 3, 6

[32] Yunhao Zou, Chenggang Yan, and Ying Fu. Iterative denoiser and noise estimator for self-supervised image denoising. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 13265–13274, 2023. 1