Convexity-Aware Noise Calibration: A Self-Supervised Framework for Noise-Level-Unknown Image Denoising

Supplementary Material

7. Proof for Mismatched Additional Noise

While this proof can be found in Noisier2Noise [25], we provide a restatement here for completeness.

Lemma 1 *Let $N \sim \mathcal{A}$ and $M \sim \mathcal{B}$, where $\mathcal{A}$ and $\mathcal{B}$ are zero-mean Gaussian distributions with $\sigma_B = \alpha\sigma_A$. Further, let $X \sim \mathcal{X}$, $Y \triangleq X+N$, and $Z \triangleq X+N+M$. Then, $\mathbb{E}[M \mid Z] = \alpha^2\mathbb{E}[N \mid Z]$.*

We first show that $\mathbb{E}[M \mid Z, X] = \alpha^2\mathbb{E}[N \mid Z, X]$.

The conditional distribution of N given $Z = z$ and $X = x$ is proportional to:

$$\mathbb{P}(N = n \mid Z = z, X = x) \propto \mathbb{P}(N = n)\cdot\mathbb{P}(M = z-x-n).$$

Substituting the Gaussian densities:

$$\propto \exp\left(-\frac{n^2}{2\sigma_A^2}\right) \cdot \exp\left(-\frac{(z-x-n)^2}{2\alpha^2\sigma_A^2}\right).$$

Combining the exponents:

$$\propto \exp\left(-\frac{\alpha^2n^2 + (z-x-n)^2}{2\alpha^2\sigma_A^2}\right).$$

This distribution is Gaussian and symmetric, with mean $\frac{1}{1+\alpha^2}(z-x)$. Similarly, the conditional distribution of M given $Z = z$ and $X = x$ has mean $\frac{\alpha^2}{1+\alpha^2}(z-x)$. Therefore,

$$\mathbb{E}[M \mid Z = z, X = x] = \alpha^2\mathbb{E}[N \mid Z = z, X = x]. \quad (16)$$

Since this equality holds for all x and z, we conclude:

$$\mathbb{E}[M \mid Z] = \alpha^2\mathbb{E}[N \mid Z]. \quad (17)$$

Since the Noisier2Noise [25] denoising formula relies on the assumption that $\mathbb{E}[M \mid Z] = \mathbb{E}[N \mid Z]$, the final denoising result differs when this condition is not met.

8. Derivation of Conditional Expectation using Tweedie's formula

In Section 4 of the main text, we established the formulation for training a denoising network when the noise level is unknown. A critical component of this formulation is the estimation of the conditional expectation $\mathbb{E}[N \mid Z']$. Here, we provide a rigorous step-by-step derivation of this expectation using Tweedie's formula.

Step 1: Distributional Formulation of the Variables

Recall from the main text that the noisy image is defined as $Y = X + N$. Under the assumption that the clean image X is deterministic and the original noise follows an Additive White Gaussian Noise (AWGN) distribution $N \sim \mathcal{N}(0, \sigma_N^2)$, the distribution of Y is strictly Gaussian:

$$Y \sim \mathcal{N}(X, \sigma_N^2) \quad (18)$$

During training, we augment Y with additional synthetic noise $M \sim \mathcal{N}(0, \sigma_M^2)$, generating the noisier observation $Z' = Y + M$. Because the synthetic noise M is independent of the original noise N (and thus independent of Y), the conditional distribution of Z' given Y is also Gaussian, parameterized by the variance of M:

$$Z' \mid Y \sim \mathcal{N}(Y, \sigma_M^2) \quad (19)$$

Step 2: Marginal Distribution and Score Function

To apply Tweedie's formula, we first need to determine the marginal probability density function $p(Z')$ and its corresponding score function. Since $Y \sim \mathcal{N}(X, \sigma_N^2)$ and $M \sim \mathcal{N}(0, \sigma_M^2)$ are independent, the marginal distribution of their sum $Z' = Y + M$ is:

$$Z' \sim \mathcal{N}(X, \sigma_N^2 + \sigma_M^2) \quad (20)$$

The log-density function of this marginal distribution is expressed as:

$$\log p(Z') = -\frac{1}{2}\log\left(2\pi(\sigma_N^2+\sigma_M^2)\right) - \frac{(Z'-X)^2}{2(\sigma_N^2+\sigma_M^2)} \quad (21)$$

Differentiating this log-density with respect to Z' yields the score function of the marginal distribution:

$$\nabla_{Z'}\log p(Z') = \frac{d}{dZ'}\log p(Z') = -\frac{Z'-X}{\sigma_N^2+\sigma_M^2} \quad (22)$$

Step 3: Applying Tweedie's Formula

Tweedie's formula provides an elegant way to compute the posterior expectation of a latent variable given an observation corrupted by Gaussian noise. For our observation model $Z' \mid Y \sim \mathcal{N}(Y, \sigma_M^2)$, Tweedie's formula dictates that the posterior mean of the latent intermediate image Y is:

$$\mathbb{E}[Y \mid Z'] = Z' + \sigma_M^2\nabla_{Z'}\log p(Z') \quad (23)$$

Substituting the score function derived in Step 2 into this equation, we obtain:

$$\mathbb{E}[Y \mid Z'] = Z' - \sigma_M^2\left(\frac{Z'-X}{\sigma_N^2+\sigma_M^2}\right) \quad (24)$$

Step 4: Recovering $\mathbb{E}[N \mid Z']$

As established in Eq. (7) of the main text, the conditional expectation of Y can be linearly decomposed due to the deterministic nature of X:

$$\mathbb{E}[Y \mid Z'] = \mathbb{E}[X + N \mid Z'] = X + \mathbb{E}[N \mid Z'] \tag{25}$$

Therefore, the expectation of the original noise $\mathbb{E}[N \mid Z']$ can be isolated by subtracting X from $\mathbb{E}[Y \mid Z']$:

$$\mathbb{E}[N \mid Z'] = \mathbb{E}[Y \mid Z'] - X \tag{26}$$

Substituting the result from Step 3 into this relation yields:

$$\mathbb{E}[N \mid Z'] = \left[Z' - \sigma_M^2 \left(\frac{Z' - X}{\sigma_N^2 + \sigma_M^2} \right) \right] - X \tag{27}$$

By regrouping the terms, we factor out $(Z' - X)$:

$$\begin{aligned} \mathbb{E}[N \mid Z'] &= (Z' - X) - \frac{\sigma_M^2}{\sigma_N^2 + \sigma_M^2}(Z' - X) \\ &= \left(1 - \frac{\sigma_M^2}{\sigma_N^2 + \sigma_M^2} \right) (Z' - X) \end{aligned} \tag{28}$$

Simplifying the scalar coefficient, we arrive at the exact formulation presented in Eq. (8) of the main text:

$$\mathbb{E}[N \mid Z'] = \frac{\sigma_N^2 + \sigma_M^2 - \sigma_M^2}{\sigma_N^2 + \sigma_M^2}(Z' - X) = \frac{\sigma_N^2}{\sigma_N^2 + \sigma_M^2}(Z' - X) \tag{29}$$

9. Variance Accuracy Analysis

The bias observed in variance estimation arises from the inherent stochasticity of deep learning training processes—including weight initialization, stochastic gradient descent (SGD) noise, and finite-precision fitting errors.

Specifically, as illustrated in Fig. 4, the loss landscape of our objective function becomes exceptionally flat in the vicinity of the ground-truth noise level. Given that the magnitude of $\mathrm{Var}(\hat{X}) - \mathrm{Var}(X)$ is on the order of 10^{-6} to 10^{-7} (as summarized in Tab. 7), the gradient signal in this region is extremely subtle and can easily be overwhelmed by the inherent stochasticity of the training process—including weight initialization, SGD-induced noise, finite-precision fitting errors, and other stochastic artifacts.

When the landscape is this flat, the optimization process no longer moves purely according to the local curvature; instead, it becomes sensitive to these random numerical fluctuations. This regime naturally induces a systematic statistical drift toward the center of the bounded interval $[0, 55]$ (i.e., 27.5), a phenomenon analogous to regression toward the mean. Consequently, this bias is an inevitable numerical artifact emerging from precision constraints within an extremely flat loss surface, rather than a fundamental deficiency in our derivation.

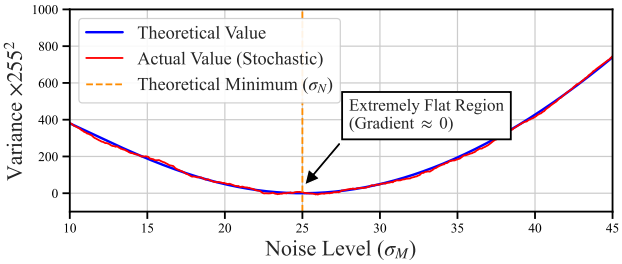

Figure 4. Inherent stochasticity on flat landscape.

10. Calibration Justification

We observe that the inherent stochasticity of the training process introduces a non-negligible bias in the estimation of variance. Values beyond the significant digits become random, introducing probabilistic variations in comparisons. For instance, if the true noise level of a dataset is $\sigma = 25$, precision limitations might cause $\mathrm{Var}(\hat{X})$ at $\sigma = 25.5$ to appear better than at $\sigma = 25$. Therefore, once the noise level is estimated, a calibration step becomes necessary to mitigate the numerical bias induced by the inherent stochasticity of the training process.

As shown in Tab. 8, with noise level search boundaries set to $[0, 55]$, values below $(0 + 55)/2 = 27.5$ tend to shift rightward, while values above 27.5 tend to shift leftward.

We employ multiple observations and least squares estimation to establish the relationship between noise level and deviation. Through least squares regression, we derive $y = 0.9622x + 1.2591$ to model the relationship between true and expected values. The calibration formula is consequently transformed to $\sigma' = (\sigma - 1.2591)/0.9622$, which serves as our final calibration function.

Regarding cross-domain generalization, it is noteworthy that our calibration parameters were derived solely from grayscale data; yet, they were applied directly to sRGB (e.g., DFWB) and real-world (e.g., SIDD, FMD) datasets in our experiments. The observation that parameters learned from synthetic grayscale images successfully generalize to color and real-world domains serves as strong empirical evidence that the calibration is an intrinsic property of the estimator itself, fundamentally independent of image semantics or color distribution.

11. Detailed Experimental Settings

In the global evaluation experiments, to maintain comparable data volume with DnCNN [48] training while following Restormer [45]'s protocol, we randomly selected 200 images from the DFWB dataset for noise estimation and training. The DFWB dataset consists of RGB images.

For the step-by-step derivation experiments, we adopted DnCNN [48]'s evaluation protocol, including identical datasets (grayscale images) and equivalent hyperparameters (learning rate, optimizer, scheduler, batch size). However,

Table 5. Comparison of theoretical and actual variance $\text{Var}(\hat{X})$ for different $\sigma_M \times 255$ values on the SET12 dataset.

$\sigma_M \times 255$	5	10	15	20	25	30	35	40	45	50	55	60
Theoretical $\text{Var}(\hat{X})$	0.0517	0.0490	0.0461	0.0439	0.0432	0.0439	0.0462	0.0497	0.0546	0.0604	0.0674	0.0754
Actual $\text{Var}(\hat{X})$	0.0505	0.0482	0.0454	0.0435	0.0428	0.0433	0.0452	0.0484	0.0524	0.0570	0.0627	0.0686

Table 6. Comparison of theoretical and actual variance $\text{Var}(\hat{X})$ for different $\sigma_M \times 255$ values on the BSD68 dataset.

$\sigma_M \times 255$	5	10	15	20	25	30	35	40	45	50	55	60
Theoretical $\text{Var}(\hat{X})$	0.0548	0.0521	0.0492	0.0470	0.0463	0.0471	0.0493	0.0529	0.0577	0.0636	0.0706	0.0785
Actual $\text{Var}(\hat{X})$	0.0523	0.0500	0.0475	0.0456	0.0447	0.0448	0.0467	0.0494	0.0530	0.0575	0.0615	0.0685

Table 7. Different noise levels correspond to distinct values of theoretical $\text{Var}(\hat{X}) - \text{Var}(X)$.

$\sigma_M \times 255$	24.5	24.6	24.7	24.8	24.9	25.0	25.1	25.2	25.3	25.4	25.5
Theoretical value	0.00000769	0.00000492	0.00000277	0.00000123	0.00000031	0.00000000	0.00000031	0.00000123	0.00000277	0.00000492	0.00000769

Table 8. True estimated values on SET12, BSD68, and Train400 datasets. All values are multiplied by 255.

	15	25	50
SET12	15.48	25.23	49.32
BSD68	15.60	25.53	48.85
Train400	15.61	25.72	49.78

to ensure stabilization of $\text{Var}(\hat{X})$, we extended training by 10 additional epochs. Both PSNR and SSIM metrics were evaluated on RGB or single-channel grayscale images.

The search boundaries for variance estimation were set to [0, 55], with a precision of 0.01. Each variance computation required at least three consecutive denoising results with sequential comparisons. Additional experimental details are available in our code repository.

12. Discussion on Limitations

12.1. Real-world applicability

While our method assumes AWGN (stated in Lines 119-131 and 585-590), it remains applicable to real-world scenarios. Complex real-world noise can be converted to approximate AWGN via Pixel-Shuffle Downsampling (PD) [51] or Variance Stabilizing Transformation (VST) [24], extending our method's scope. We validated this on SIDD (smartphone) and FMD (microscopy) datasets. We randomly sampled 200 images from the SIDD dataset and utilized the entire Confocal BPAE subset from FMD, conducting experiments in accordance with the established original settings. As shown in Tab. 9, combining our method with PD_2 to decorrelate spatial noise yields significant improvements over baselines ignoring correlation.

12.2. Computational cost

Although our method achieves accurate noise level estimation, it introduces a higher computational complexity compared to simpler approaches. Let $\mathcal{O}(T)$ be the baseline complexity. With k iterations of ternary search (each requiring two trisection evaluations), the total complexity becomes $\mathcal{O}_{\text{ours}} = 2k \cdot \mathcal{O}(T)$. Given the search interval reduction of $2/3$ per step, the error decays exponentially as $|\sigma_M^{2*} - \sigma_N^2| \leq l \cdot (2/3)^k$. Empirically, $k = 10$ yields high precision ($\epsilon \approx 0.01/255$) for an interval $l = 0.5 = 127.5/255$. For better accuracy, each variance comparison may incorporate multiple training results; Details will be released in our code.

Table 9. Comparison of our method on real-world datasets.

Method	SIDD		FMD	
	PSNR↑	SSIM↑	PSNR↑	SSIM↑
Noise2Void	24.25	0.369	25.41	0.377
Noise2Self	26.66	0.509	27.26	0.486
Neighbor2Neighbor	27.68	0.552	26.88	0.469
Proprosed	27.71	0.591	27.41	0.521
Proposed w/ PD	**31.22**	**0.716**	**30.18**	**0.706**