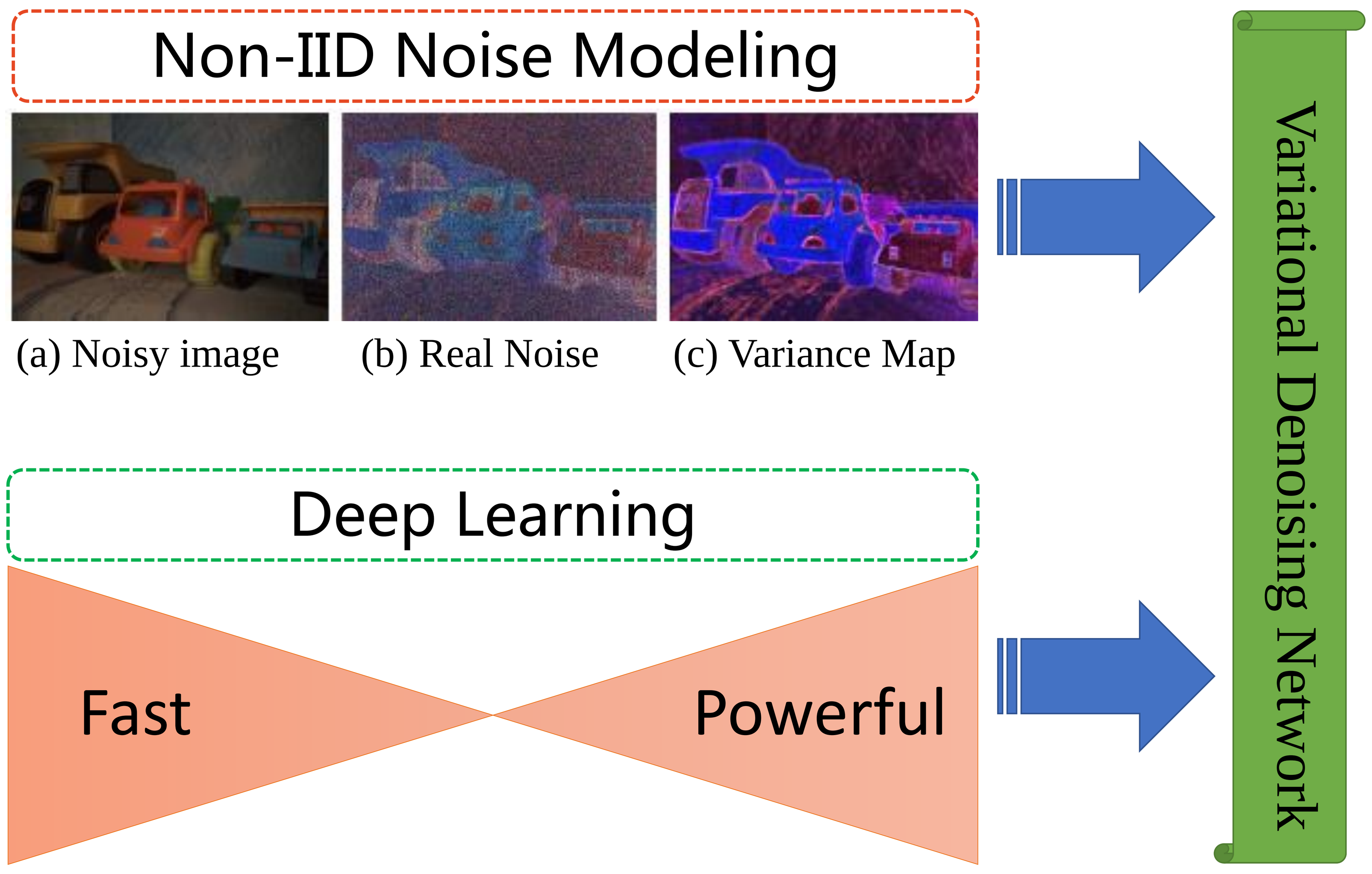
Variational Denoising Networks: Toward Blind Noise Modeling and Removal (NeurIPS 2019)

Zongsheng Yue[1]; Hongwei Yong[2,3]; Qian Zhao[1]; Lei Zhang[2,3]; Deyu Meng[1]

[1]Xi'an Jiaotong University, [2]The Hong Kong Polytechnic University

[3]DAMO Academy, Alibaba Group

Motivation



- Integrate both of the noise estimation and denoising tasks into a unique Bayesian Framework.
- Better generalization capability due to the generative model.
- Degenerate into MSE loss in special case.

Full Bayesian Model

Training set $D = \{\boldsymbol{y}_j, \boldsymbol{x}_j\}_{j=1}^{n}$, $\boldsymbol{y}_j$: Noisy image, $\boldsymbol{x}_j$: Simulated clean image.

For any $\{\boldsymbol{y}, \boldsymbol{x}\} \in D$, we assumed the following generation process:

$$y_i \sim N(y_i | z_i, \sigma_i^2), i = 1,2,\cdots,d$$

Conjugate prior for $\boldsymbol{z}$:

$$z_i \sim N(z_i | x_i, \varepsilon_0^2), i = 1,2,\cdots,d$$

Conjugate prior for $\boldsymbol{\sigma}$:

$$\sigma_i^2 \sim IG\left(\sigma_i^2 \middle| \frac{p^2}{2} - 1, \frac{p^2 \xi_i}{2}\right), i = 1,2,\cdots,d$$

Where $\xi = \mathcal{G}\left((\boldsymbol{y} - \boldsymbol{x})^2; p\right)$.

Goal (Posterior Inference):

$$p(\boldsymbol{z}, \boldsymbol{\sigma}^2 | \boldsymbol{y}) \xleftarrow{\text{approximate}} q(z, \sigma^2 | y)$$

Likelihood decomposition:

$$\log p(\boldsymbol{y}; \boldsymbol{z}, \boldsymbol{\sigma}^2) = \underbrace{\mathcal{L}(\boldsymbol{z}, \boldsymbol{\sigma}^2; \boldsymbol{y})}_{\textit{Lower Bound}} + \underbrace{D_{KL}\left(q(\boldsymbol{z}, \boldsymbol{\sigma}^2 | \boldsymbol{y}) || p(\boldsymbol{z}, \boldsymbol{\sigma}^2 | \boldsymbol{y})\right)}_{\textit{Non-Negative}}$$

Objective Function:

$$\boxed{\max \mathcal{L}(\boldsymbol{z}, \boldsymbol{\sigma}^2; \boldsymbol{y})}$$

Conditional independence assumption:

$$q(\boldsymbol{z}, \boldsymbol{\sigma}^2 | \boldsymbol{y}) = q(\boldsymbol{z} | \boldsymbol{y}) q(\boldsymbol{\sigma}^2 | \boldsymbol{y})$$

Variational posterior form:

$$q(z|y) = \prod_i N(z_i | \mu_i, m_i^2), q(\sigma^2 | y) = \prod_i IG(\sigma_i^2 | \alpha_i, \beta_i)$$

Analytical lower bound:

$$\mathcal{L}(z, \sigma^2; y) = E_{q(z,\sigma^2|y)}\left[\log p(y | z, \sigma^2)\right] - D_{KL}\left(q(z|y) || p(z)\right) - D_{KL}\left(q(\sigma^2|y) || p(\sigma^2)\right)$$

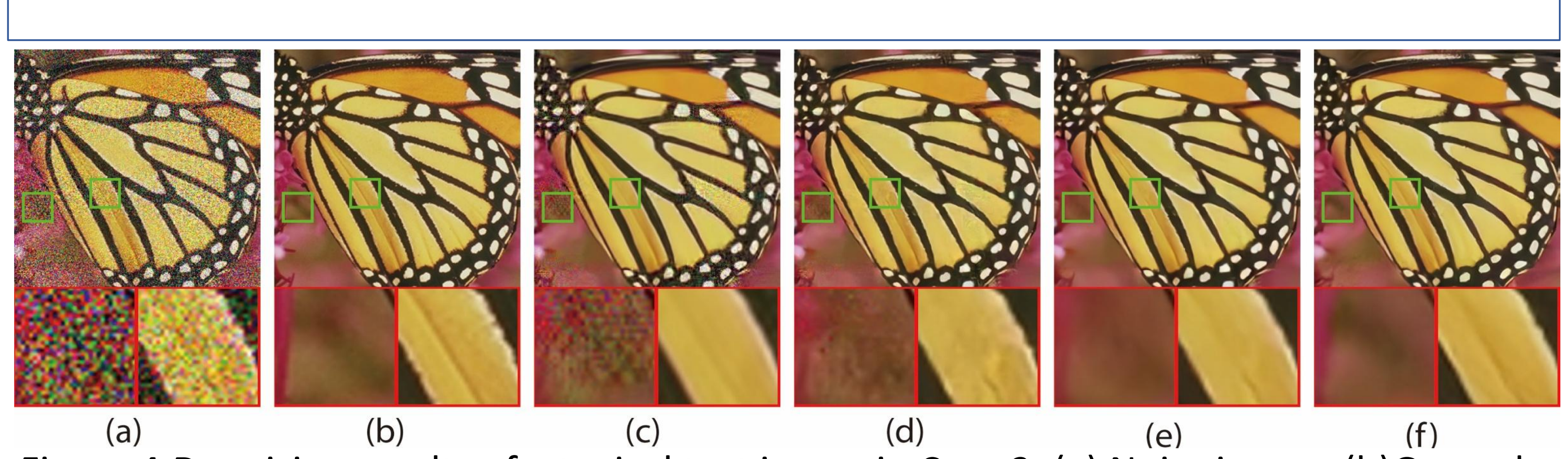


Figure 4:Denoising results of a typical test image in Case 2. (a) Noisy image, (b)Ground-truth, (c) CBM3D , (d) DnCNN (27.83 dB), (e) FFDNet (28.06 dB), (f) VDN (28.32 dB).

Network Learning

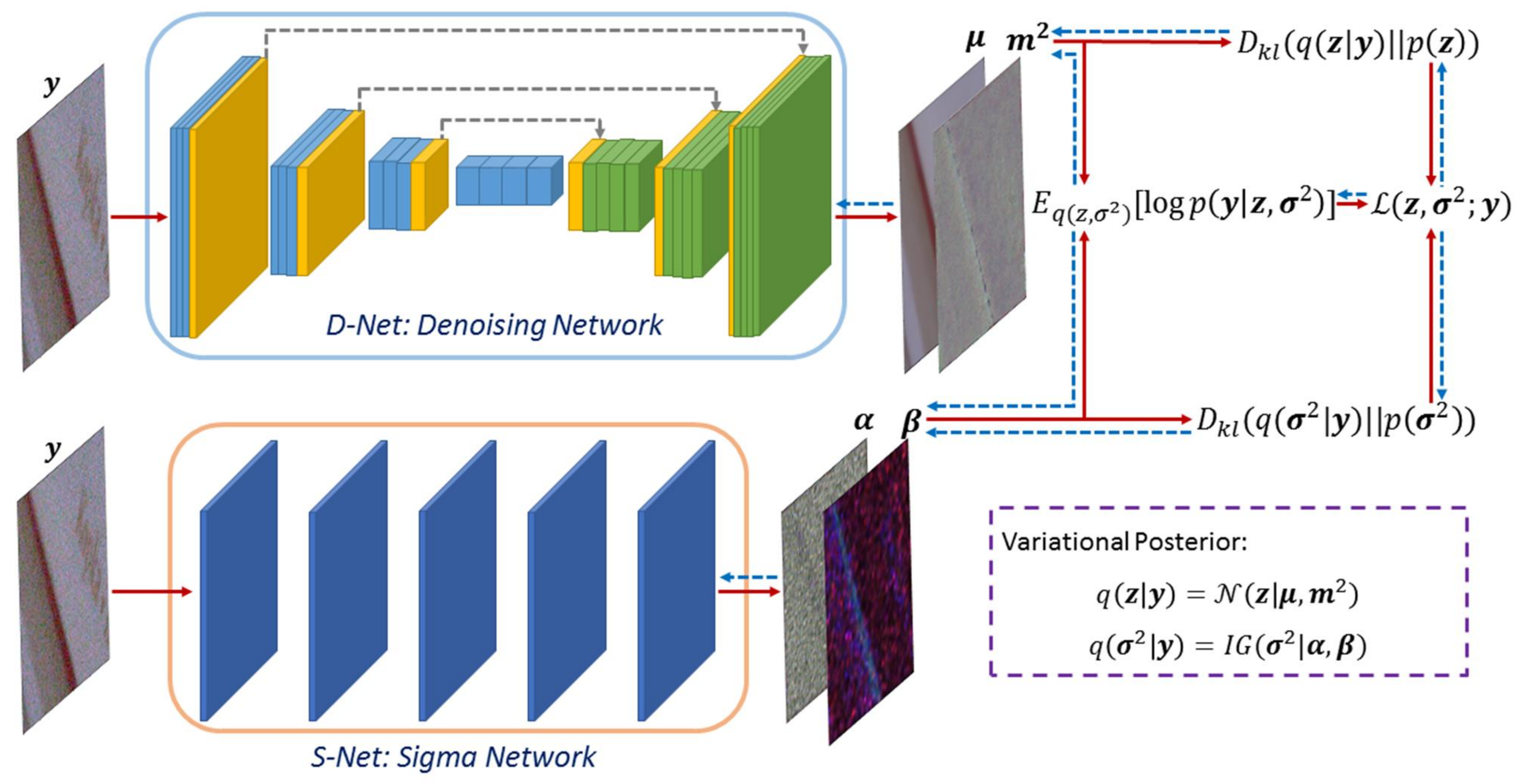


Figure2: The proposed deep variational inference network. The red solid lines denote the forward process, and the blue dotted lines mark the gradient flow .

It should be noted that our method is a general framework, most of the commonly used network architecture in image restoration can also be easily substituted.

Non-IID Gaussian Denoising

Noise generation: $\boldsymbol{n} = \boldsymbol{n}^1 \odot \boldsymbol{M}, n_{ij}^1 \sim N(n_{ij}^1 | 0, 1)$

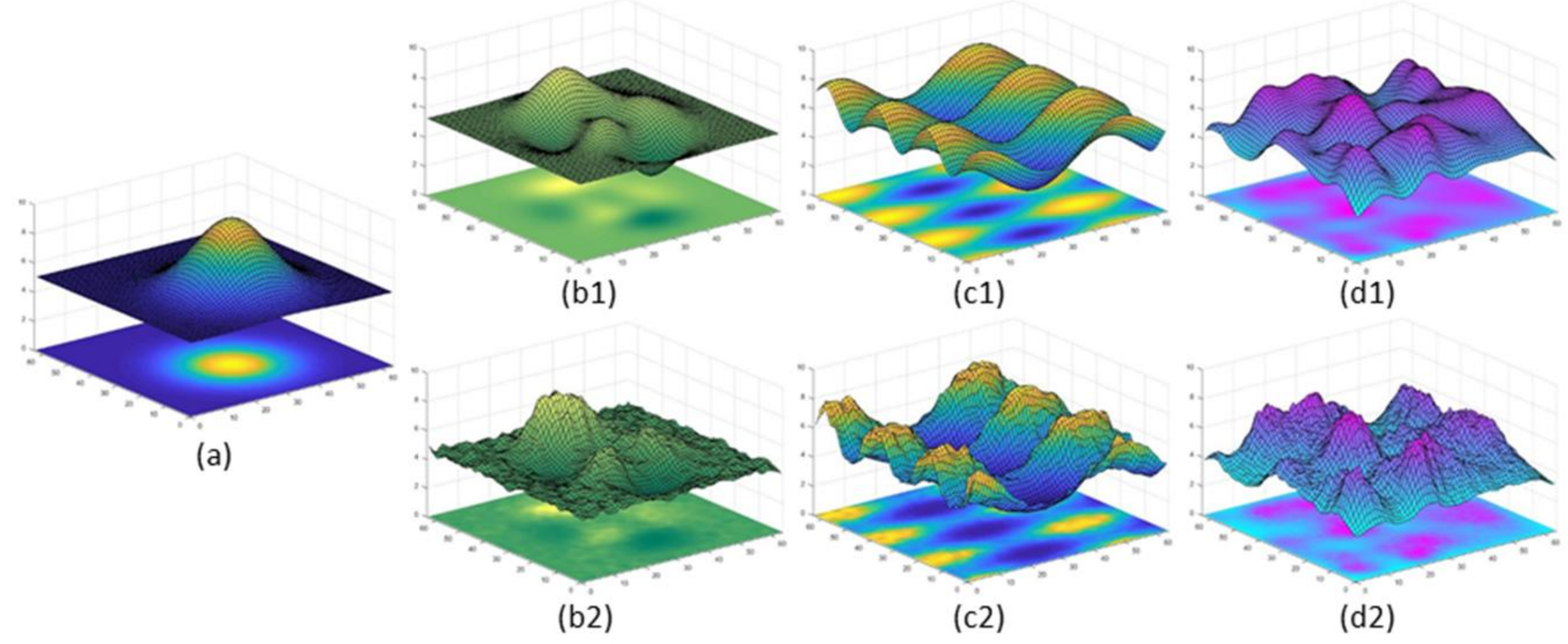


Figure 3: (a) The variance map M for noise generation in training data. (b1)-(d1): Three different Ms on the testing data in Case 1-3. (b2)-(d2): Corresponding predicted Ms by our method on the testing data.

Table 1: The PSNR results of all competing methods on the three groups of test datasets.

Cases	Datasets	Methods								
		CBM3D	WNNM	NCSR	MLP	DnCNN	FFDNet	$FFDNet_v$	UDNet	VDN
Case 1	Set5	27.76	26.53	26.62	27.26	29.87	30.16	30.15	28.13	**30.39**
	LIVE1	26.58	25.27	24.96	25.71	28.81	28.99	28.96	27.19	**29.22**
	BSD68	26.51	25.13	24.96	25.58	28.72	28.78	28.77	27.13	**29.02**
Case 2	Set5	26.34	24.61	25.76	25.73	29.05	29.60	29.56	26.01	**29.80**
	LIVE1	25.18	23.52	24.08	24.31	28.18	28.58	28.56	25.25	**28.82**
	BSD68	25.28	23.52	24.27	24.30	28.14	28.43	28.42	25.13	**28.67**
Case 3	Set5	27.88	26.07	26.84	26.88	29.17	29.54	29.49	27.54	**29.74**
	LIVE1	26.50	24.67	24.96	25.26	28.15	28.39	28.38	26.48	**28.65**
	BSD68	26.44	24.60	24.95	25.10	28.10	28.22	28.20	26.44	**28.46**

Real Benchmark Performance

Table 2: The comparison results of different methods on SIDD Benchmark and Validation dataset.

Datasets	SIDD Benchmark						SIDD Validation		
Methods	CBM3D	WNNM	MLP	DnCNN	CBDNet	VDN	DnCNN	CBDNet	VDN
PSNR	25.65	25.78	24.71	23.66	33.28	**39.23**	38.41	38.68	**39.28**
SSIM	0.685	0.809	0.641	0.583	0.868	**0.971**	**0.909**	0.901	**0.909**

Table 3: The comparison results of different methods DND Benchmark.

Methods	CBM3D	WNNM	NCSR	MLP	DnCNN	FFDNet	CBDNet	VDN
PSNR	34.51	34.67	34.05	34.23	37.90	37.61	38.06	**39.38**
SSIM	0.8507	0.8646	0.8351	0.8331	0.9430	0.9415	0.9421	**0.9518**

Table 4: Hyper-parameters ε_0^2 analysis on the SIDD validation dataset.

ε_0^2	1e-4	1e-5	1e-6	1e-7	1e-8	MSE
PSNR	38.89	39.20	**39.28**	39.05	39.03	39.01
SSIM	0.9046	0.9079	**0.9086**	0.9064	0.9063	0.9061

Table 5: Hyper-parameters p analysis on the SIDD validation dataset.

p	5	7	11	15	19
PSNR	39.26	**39.28**	39.26	39.24	39.24
SSIM	**0.9089**	0.9086	0.9086	0.9079	0.9079

Paper

Source Code