Variational Denoising Network

Deyu Meng

Xi’an Jiaotong University

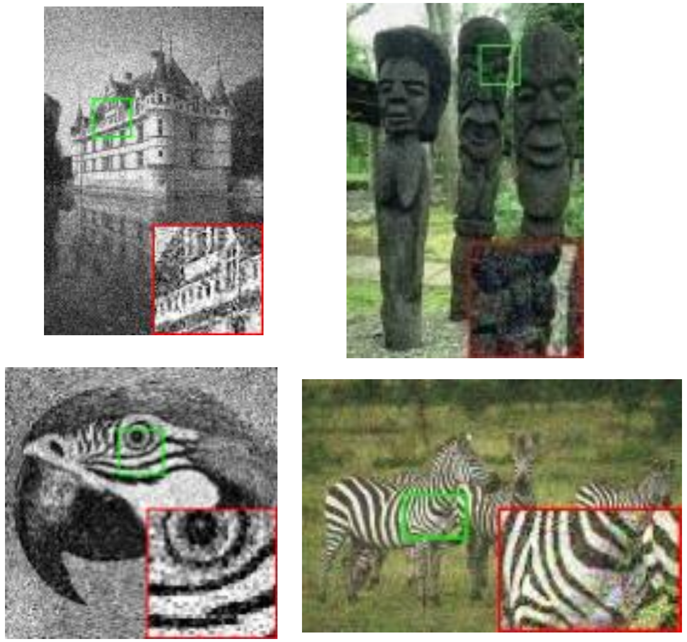
dymeng@mail.xjtu.edu.cn

http://gr.xjtu.edu.cn/web/dymeng

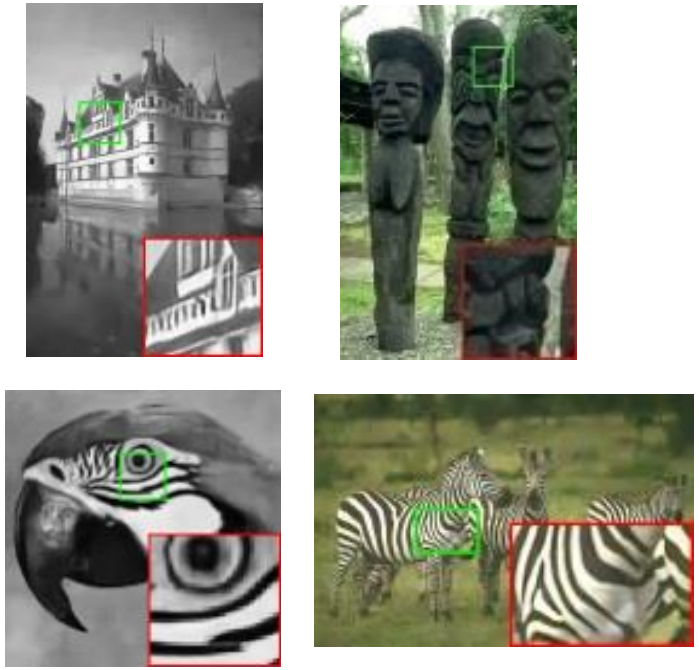
Denoising Problem

Assumption: $Y = Z + E$

Observation Y



Recovery Z*



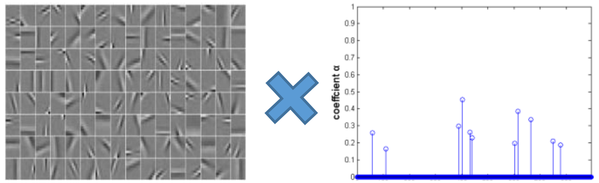
Model-driven Methodology

2010-2012

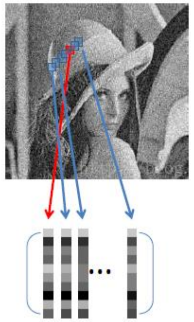
$$\arg\min_{Z} \ ||Y - Z||_2 + R(Z)$$

Low rank

Sparsity



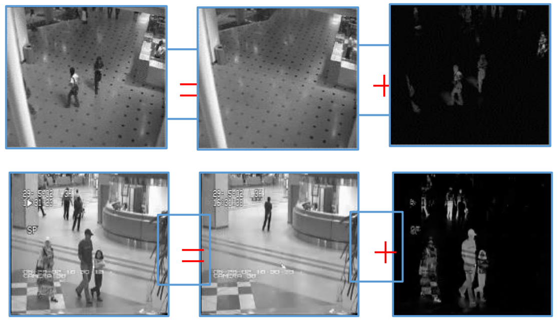
Non-local Similarity



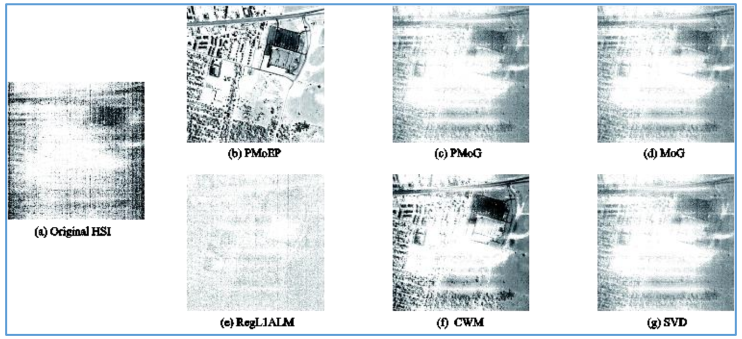
Gu, Xie, Meng, Zuo, Feng, Zhang, IJCV, 2017.

Model-driven Methodology: Noise Modeling

$$\arg\min_{Z,\theta} L_\theta(Y - Z) + R(Z) + R(\theta)$$

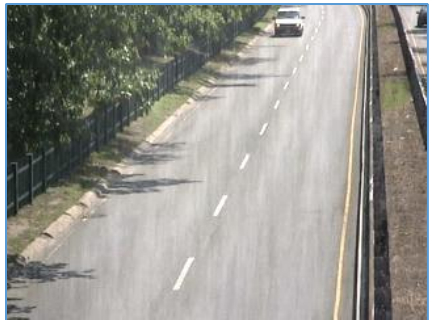


$$e \sim \sum_k \pi_k N(e|0, {\sigma_k}^2)$$



$$e \sim \sum_k \pi_k EP_{p_k}(e|0, \eta_k)$$

$$\boldsymbol{e} \sim \sum_k \pi_k N(\boldsymbol{e}|0, \Sigma_k)$$



DY Meng, D Fernando, ICCV 2013
Q, Zhao, DY Meng, et al., ICML, 2014

XY Cao, Q Zhao, DY Meng, et al., ICCV 2015

W Wei, LX Yi, DY Meng, et al., ICCV 2017

Model-driven Methodology

$$\arg\min_{Z,\theta} L_\theta(Y - Z) + R(Z) + R(\theta)$$

$Y \Rightarrow Z* = \text{Algorithm}(Y)$

Model-driven Methodology: Generative Understanding

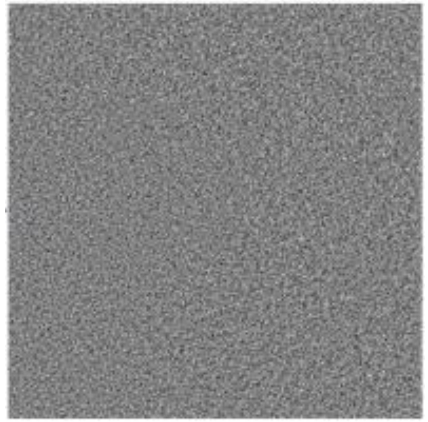
$$\arg\min_{Z,\theta} L_\theta(Y - Z) + R(Z) + R(\theta)$$

Y

=

Z

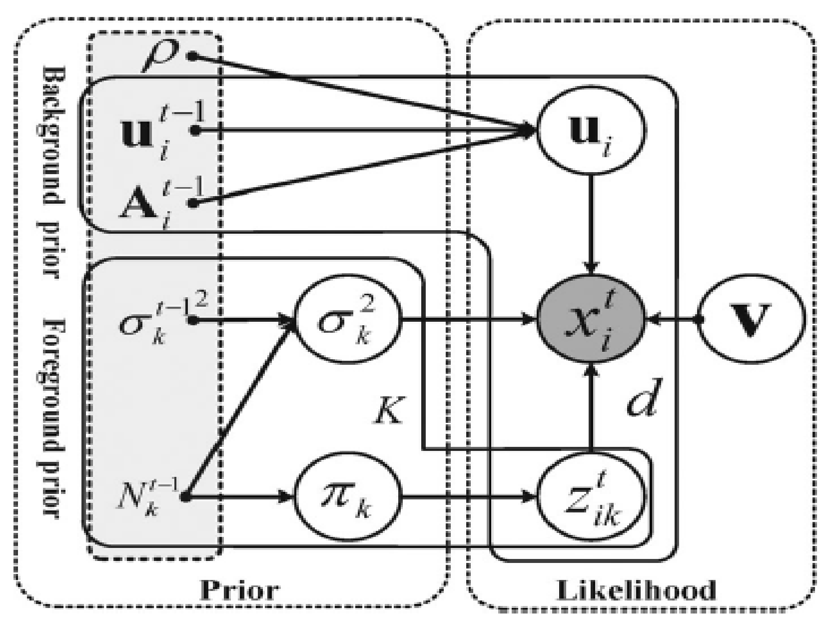
+



E

$$z \sim p(z);\ e \sim p(e;\theta) \quad \Rightarrow \quad p(z, e|y) \sim \text{likelihood}(y|z, e)\, p(z)\, p(e)$$

Model-driven Methodology: Generative Understanding



Yong, Meng, Zuo, Zhang, TPAMI, 2018

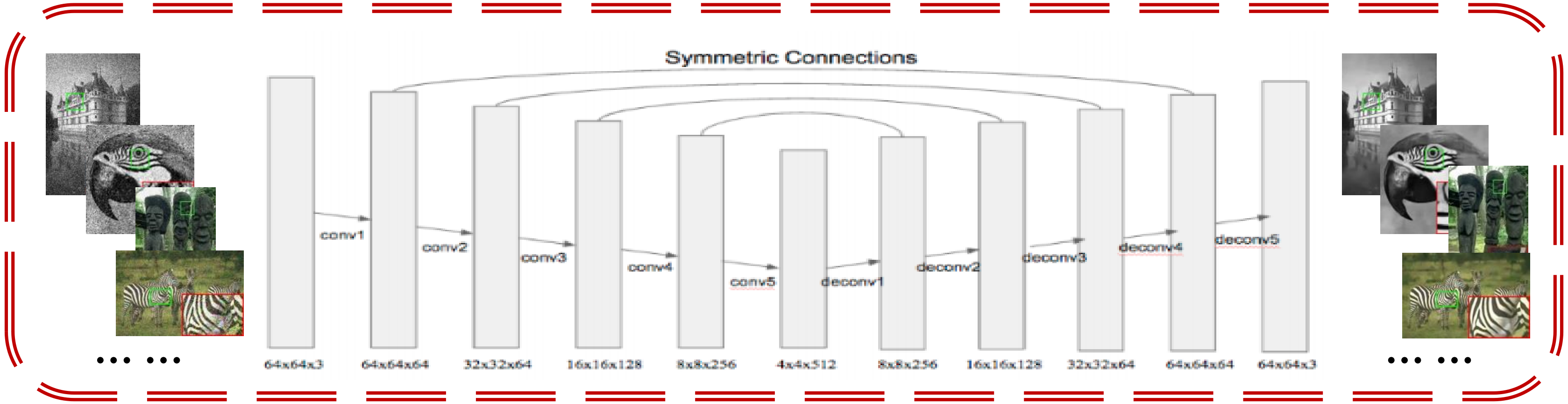
$$p(\Pi,\Sigma,\mathrm{v},\mathrm{U}|\mathrm{x}^t,\Theta^{t-1}) \propto p(\mathrm{x}^t|\Pi,\Sigma,\mathrm{v},\mathrm{U})p(\Sigma|\Theta^{t-1})p(\Pi|\Theta^{t-1})p(\mathrm{U}|\Theta^{t-1})p(\mathrm{v})$$

$$z \sim p(z);\, e \sim p(e;\theta) \quad \rightarrow \quad p(z,e|y) \sim \mathrm{likelihood}(y|z,e)p(z)p(e)$$

Data-driven Methodology: Learn Clean Image

2010-2012

$$\arg\min_{W} \ ||Z - \mathrm{Network}_{W}(Y)||_2$$

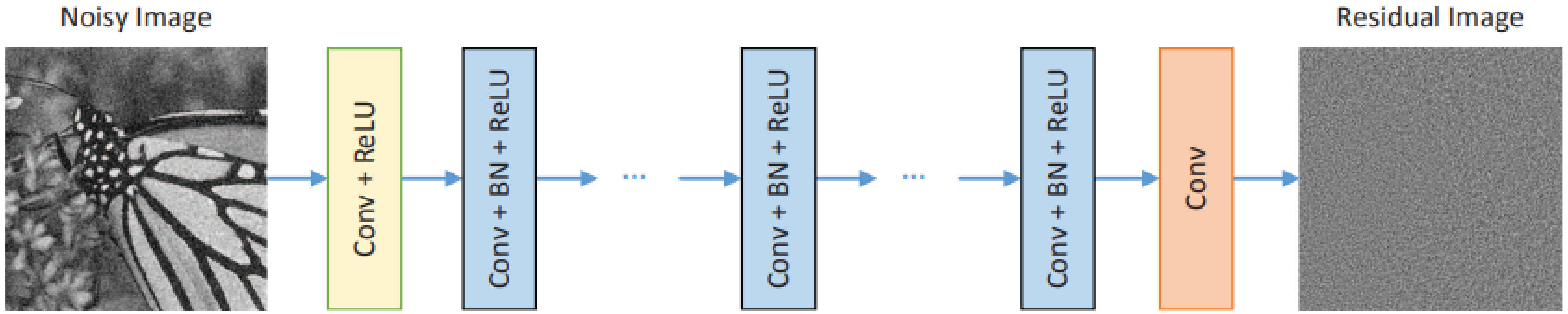


$Y \rightarrow Z^{*} = \mathrm{Network}_{W^{*}}(Y)$

Data-driven Methodology: Learn Noise

2010-2012

$$\arg\min_{W} \; ||E - \text{Network}_W(Y)||_2$$



Residual Network

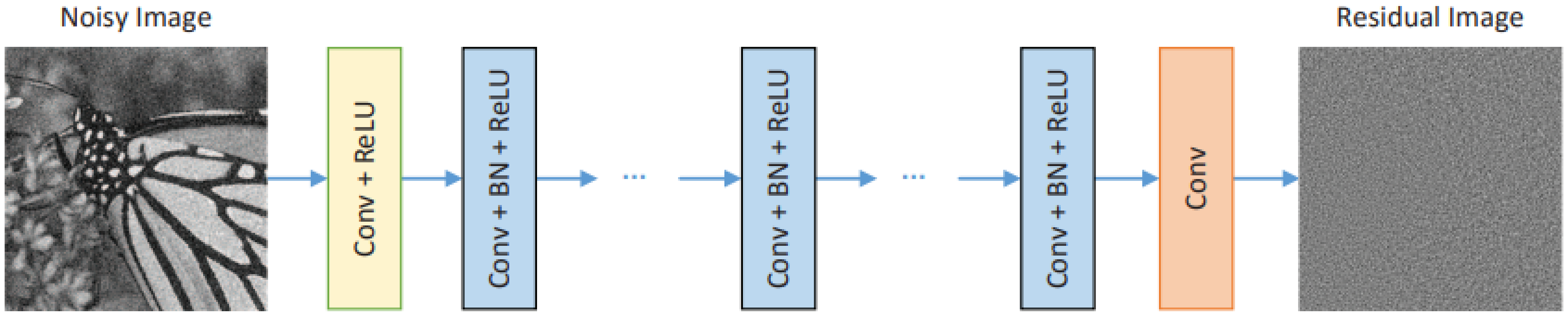
$$Y \longrightarrow E_* = \text{Network}_{W_*}(Y)$$

Zhang, Zuo, Chen, Meng, Zhang, TIP, 2017

Data-driven Methodology: Learn Noise

2010-2012

$$\arg\min_{W} \left\|E - \mathrm{Network}_{W}(Y)\right\|_{2}$$



Residual Network

Noise (distribution) should be more proper to be represented in stochastic manner instead of deterministic!

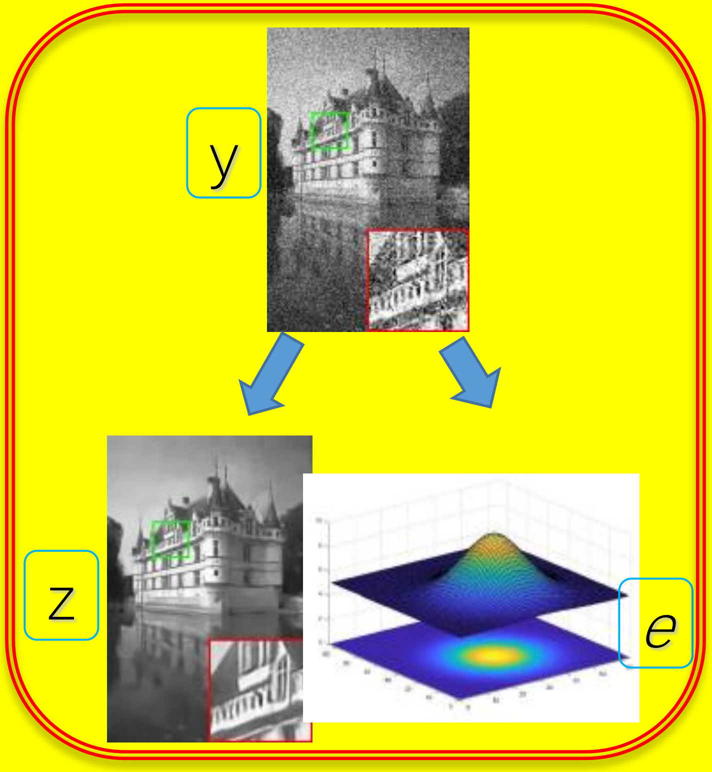
Motivation of This Work

- For Model-driven Methods:
  - ✓ Alleviate influence of assumptions on image and noise prior structures (better fit non-i.i.d. noises)
  - ✓ From parametric to more or less non-parametric
- For Data-driven methods:
  - ✓ Fit in Bayesian framework and make noises used more properly (stochastic end-to-end learning manner)
  - ✓ Alleviate the over-fitting issue to training data
- From noise estimation to noise inference for blind image denoising

Motivation of This Work

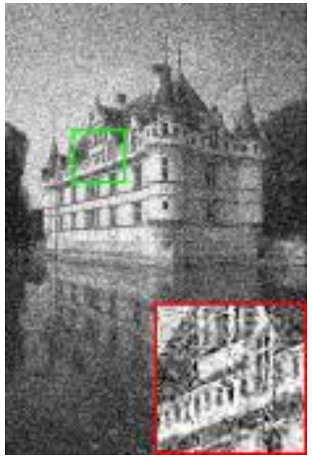
- For M
  - ✓ Al ... ctures (bett
  - ✓ Fr

$$q(z, \sigma^2 | y)$$



- For I
  - ✓ Fi ... (stochastic end-t
  - ✓ All
- From noise estimation to noise inference for blind image denoising

Problem Setting: Real Posterior



$\boldsymbol{y} = [y_1, \cdots, y_d]^T$ $\quad$ $\boldsymbol{x} = [x_1, \cdots, x_d]^T$

$$\boldsymbol{y} = \boldsymbol{z} + \boldsymbol{e},$$

$$y_i \sim N\left(y_i | z_i, \sigma_i^2\right)$$

Prior of z:

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

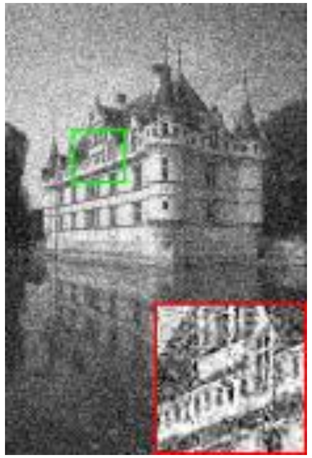
Prior of noise variance:

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

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

the filtering output of the variance map $(\hat{y} - \hat{x})^2$ by a Gaussian filter with $p \times p$ window

Problem Setting: Real Posterior



$\boldsymbol{y} = [y_1, \cdots, y_d]^T$

$\boldsymbol{x} = [x_1, \cdots, x_d]^T$

$$\mathrm{y_i} \sim \mathrm{N}\left(\mathrm{y_i} | \mathrm{z_i}, \sigma_\mathrm{i}^2\right)$$

Prior of z:

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

Prior of noise variance:

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

$$p(\boldsymbol{z}, \boldsymbol{\sigma}^2 | \boldsymbol{y}) \Longleftarrow \log p(\boldsymbol{y} | \boldsymbol{z}, \boldsymbol{\sigma}^2) p(\boldsymbol{z}) p(\boldsymbol{\sigma}^2)$$

Varational Posterior

$$p(z, \sigma^2 | y) \Rightarrow q(z, \sigma^2 | y) = q(z|y)q(\sigma^2|y)$$

$$q(z|y) = \prod_i^d \mathcal{N}(z_i | \mu_i(y; W_D), m_i^2(y; W_D))$$

D-Net

$$q(\sigma^2|y) = \prod_i^d \mathrm{IG}(\sigma_i^2 | \alpha_i(y; W_S), \beta_i(y; W_S))$$

S-Net

Network parameters W_D and W_S are shared by posteriors calculated on all training data

Objective: Minimizing KL Divergence

$$\min_{W_D, W_S} D_{KL}\left(q(z, \sigma^2 | y) || p(z, \sigma^2 | y)\right)$$

How?

Variational Inference!

How to Calculate KL? Variational Lower Bound

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

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

Min $D_{KL}\left(q(z,\sigma^2|y)||p(z,\sigma^2|y)\right)$ $\Leftrightarrow$ Max $\mathcal{L}(z,\sigma^2;y)$

- Widely used to design Bayesian inference algorithms:
  - ✓ Classical variational inference
  - ✓ EM
  - ✓ VAE

Objective Function of Our Method: All Closed-form

$$\text{Min } D_{KL}\left(q(z,\boldsymbol{\sigma}^2|y)||p(z,\boldsymbol{\sigma}^2|\boldsymbol{y})\right) \Longleftrightarrow \boxed{\text{Max } \mathcal{L}(z,\boldsymbol{\sigma}^2;y)}$$

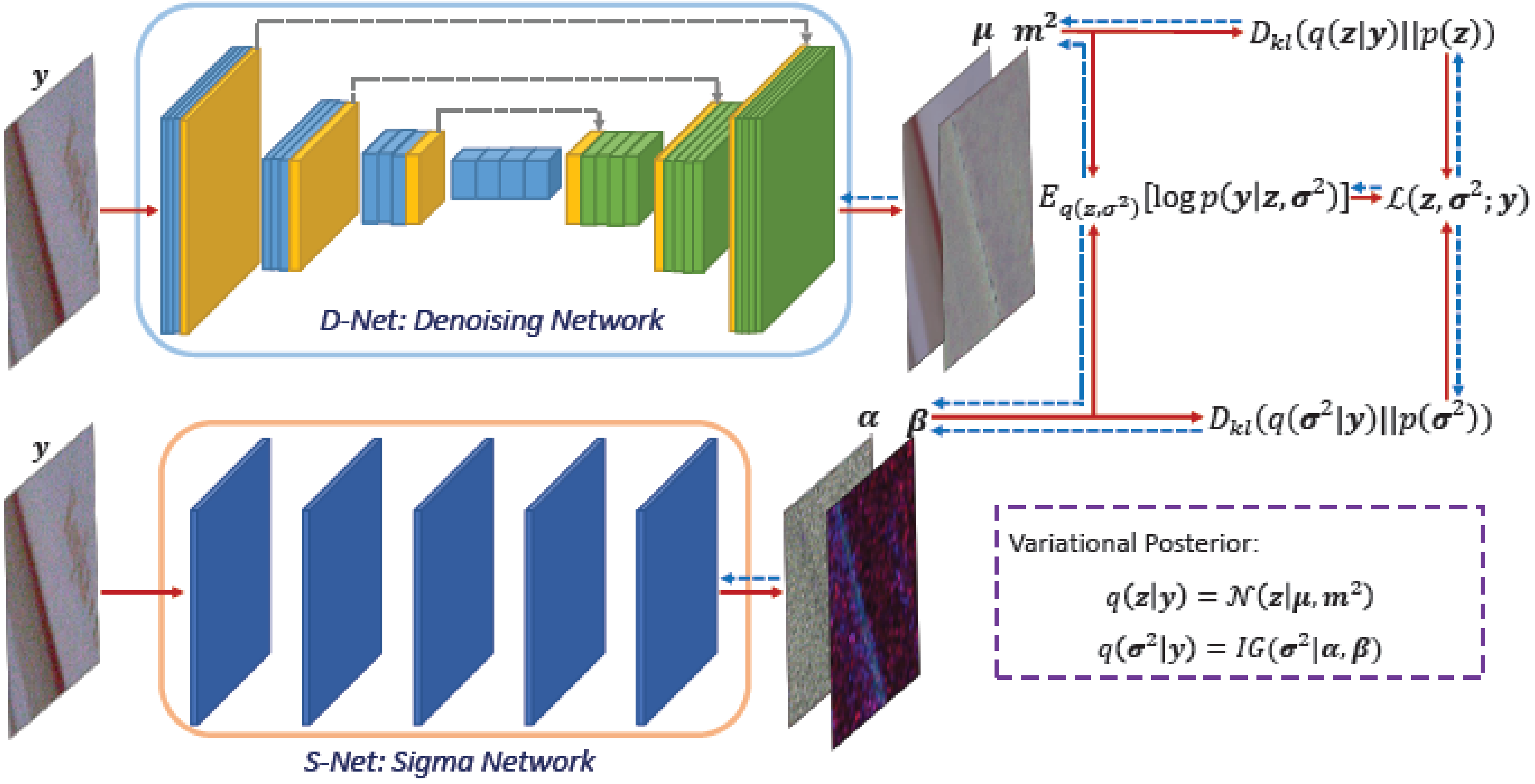
$$\mathcal{L}(z,\boldsymbol{\sigma}^2;y) = E_{q(\boldsymbol{z},\boldsymbol{\sigma}^2|\boldsymbol{y})}\left[\log p(y|z,\boldsymbol{\sigma}^2)\right] - D_{KL}\left(q(z|\boldsymbol{y})||p(z)\right) - D_{KL}\left(q(\boldsymbol{\sigma}^2|y)||p(\boldsymbol{\sigma}^2)\right)$$

$$E_{q(\boldsymbol{z},\boldsymbol{\sigma}^2|y)}\left[\log p(y|z,\boldsymbol{\sigma}^2)\right] = \sum_{i=1}^{d}\left\{-\frac{1}{2}\log 2\pi - \frac{1}{2}(\log\beta_i - \psi(\alpha_i)) - \frac{\alpha_i}{2\beta_i}\left[(y_i-\mu_i)^2 + m_i^2\right]\right\}$$

$$D_{KL}\left(q(z|y)||p(z)\right) = \sum_{i=1}^{d}\left\{\frac{(\mu_i - x_i)^2}{2\varepsilon_0^2} + \frac{1}{2}\left[\frac{m_i^2}{\varepsilon_0^2} - \log\frac{m_i^2}{\varepsilon_0^2} - 1\right]\right\}$$

$$\begin{aligned} D_{KL}\left(q(\boldsymbol{\sigma}^2|y)||p(\boldsymbol{\sigma}^2)\right) = \sum_{i=1}^{d}\Bigg\{&\left(\alpha_i - \frac{p^2}{2} + 1\right)\psi(\alpha_i) + \left[\log\Gamma\left(\frac{p^2}{2} - 1\right) - \log\Gamma(\alpha_i)\right] \\ &+ \left(\frac{p^2}{2} - 1\right)\left(\log\beta_i - \log\frac{p^2\xi_i}{2}\right) + \alpha_i\left(\frac{p^2\xi_i}{2\beta_i} - 1\right)\Bigg\}\end{aligned}$$

Implementation Scheme



Variational Posterior:

$$q(\boldsymbol{z}|\boldsymbol{y}) = \mathcal{N}(\boldsymbol{z}|\boldsymbol{\mu},\boldsymbol{m}^2)$$

$$q(\boldsymbol{\sigma}^2|\boldsymbol{y}) = IG(\boldsymbol{\sigma}^2|\boldsymbol{\alpha},\boldsymbol{\beta})$$

More Explanations on Rationality of This Objective

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

$$E_{q(\boldsymbol{z},\boldsymbol{\sigma}^2|\boldsymbol{y})}\left[\log p(\boldsymbol{y}|\boldsymbol{z},\boldsymbol{\sigma}^2)\right] = \sum_{i=1}^{d}\left\{-\frac{1}{2}\log 2\pi - \frac{1}{2}(\log\beta_i - \psi(\alpha_i)) - \frac{\alpha_i}{2\beta_i}\left[(y_i-\mu_i)^2 + m_i^2\right]\right\}$$

$$D_{KL}\left(q(\boldsymbol{z}|\boldsymbol{y})||p(\boldsymbol{z})\right) = \sum_{i=1}^{d}\left\{\frac{(\mu_i - x_i)^2}{2\varepsilon_0^2} + \frac{1}{2}\left[\frac{m_i^2}{\varepsilon_0^2} - \log\frac{m_i^2}{\varepsilon_0^2} - 1\right]\right\}$$

$$D_{KL}\left(q(\boldsymbol{\sigma}^2|\boldsymbol{y})||p(\boldsymbol{\sigma}^2)\right) = \sum_{i=1}^{d}\left\{\left(\alpha_i - \frac{p^2}{2} + 1\right)\psi(\alpha_i) + \left[\log\Gamma\left(\frac{p^2}{2} - 1\right) - \log\Gamma(\alpha_i)\right] + \left(\frac{p^2}{2} - 1\right)\left(\log\beta_i - \log\frac{p^2\xi_i}{2}\right) + \alpha_i\left(\frac{p^2\xi_i}{2\beta_i} - 1\right)\right\}$$

- Weighted least square loss
- Robust learning scheme
- Consistent to our previous noise modeling methodology

Degeneration to Classical Denoising Network

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

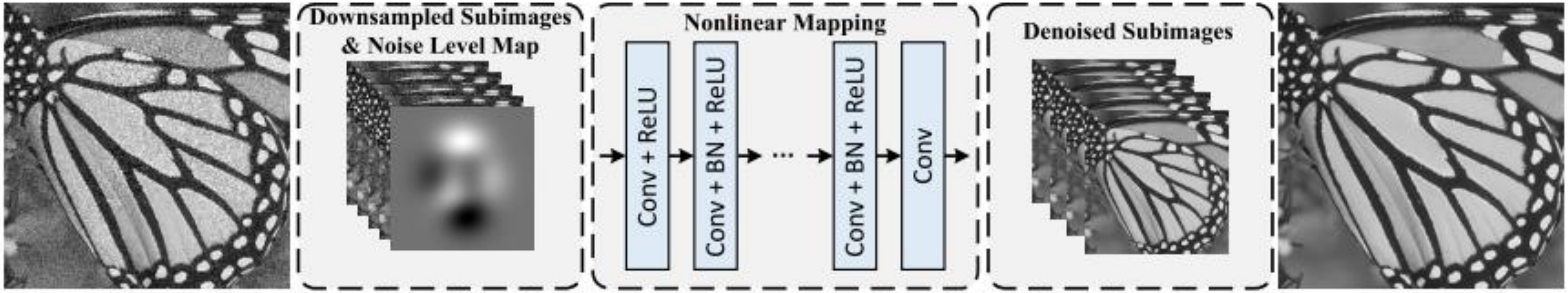
$$E_{q(\boldsymbol{z},\boldsymbol{\sigma}^2|\boldsymbol{y})}\left[\log p(\boldsymbol{y}|\boldsymbol{z},\boldsymbol{\sigma}^2)\right] = \sum_{i=1}^{d}\left\{-\frac{1}{2}\log 2\pi - \frac{1}{2}(\log\beta_i - \psi(\alpha_i)) - \frac{\alpha_i}{2\beta_i}\left[(y_i-\mu_i)^2 + m_i^2\right]\right\}$$

$$D_{KL}\left(q(\boldsymbol{z}|\boldsymbol{y})||p(\boldsymbol{z})\right) = \sum_{i=1}^{d}\left\{\frac{(\mu_i - x_i)^2}{2\varepsilon_0^2} + \frac{1}{2}\left[\frac{m_i^2}{\varepsilon_0^2} - \log\frac{m_i^2}{\varepsilon_0^2} - 1\right]\right\}$$

$$D_{KL}\left(q(\boldsymbol{\sigma}^2|\boldsymbol{y})||p(\boldsymbol{\sigma}^2)\right) = \sum_{i=1}^{d}\left\{\left(\alpha_i - \frac{p^2}{2} + 1\right)\psi(\alpha_i) + \left[\log\Gamma\left(\frac{p^2}{2} - 1\right) - \log\Gamma(\alpha_i)\right] + \left(\frac{p^2}{2} - 1\right)\left(\log\beta_i - \log\frac{p^2\xi_i}{2}\right) + \alpha_i\left(\frac{p^2\xi_i}{2\beta_i} - 1\right)\right\}$$

- Set epslo_0 to almost zero, the method will be degenerated to classical deep learning strategy
- The posterior inference process puts dominant emphasis on fitting priors imposed on the latent clean image, while almost neglects the effect of noise variations. This naturally leads to its sensitiveness to unseen complicated noises contained in test images.

Some Current Blind Denosing Methods

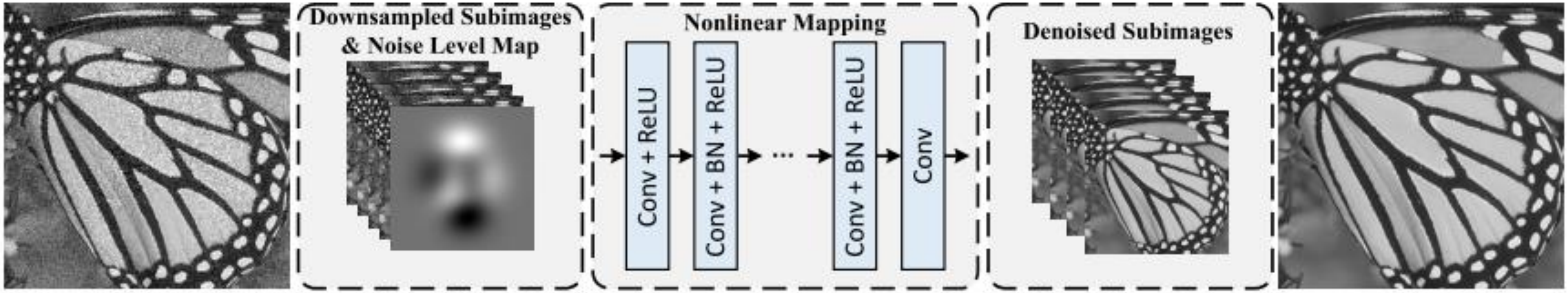


A supplemental stage to estimate the noise level, and then input this knowledge into network together with noisy image

Zhang Zuo, Zhang,TIP, 2018.

Guo, Yan, Zhang, Zuo, Zhang. arXiv:1807.04686, 2018

Difference Between Current Blind Denosing Method



From noise estimation to noise inference
Alleviate workload in testing stage

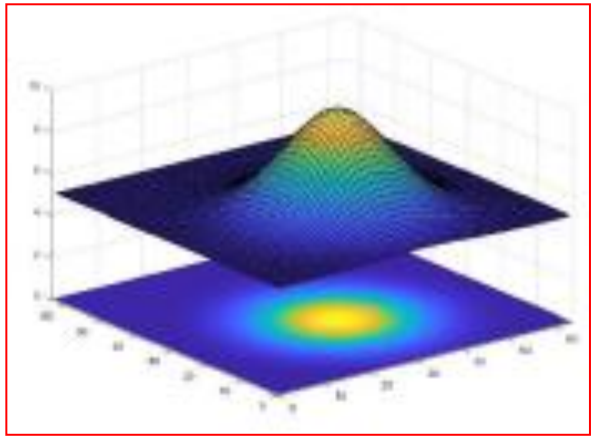
Synthetic Experiments

Training images:

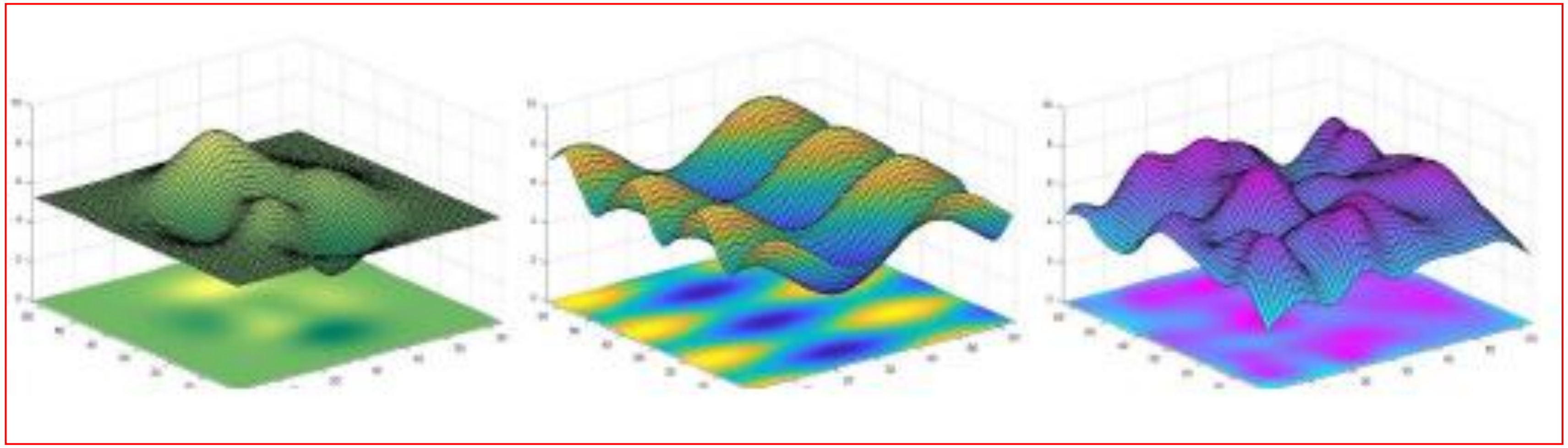
- ✓ 432 images from BSD
- ✓ 400 images from ImageNet
- ✓ 4744 images from Waterloo

Test images:

- ✓ Set5
- ✓ LIVE1
- ✓ BSD68



Training Noise



Test Noise

Synthetic Experiments

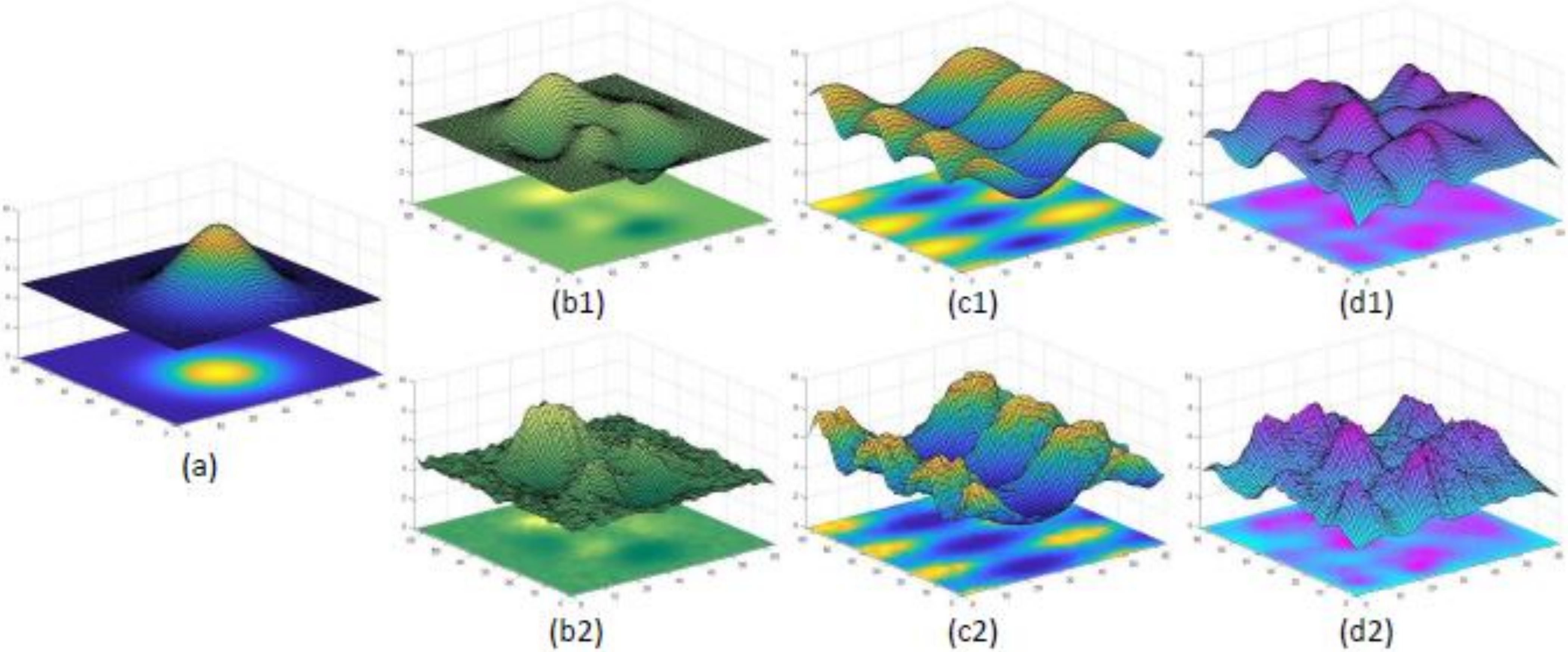
Table 1: The PSNR(dB) results of all competing methods on the three groups of test datasets. The best and second best results are highlighted in bold and Italic, respectively.

Cases	Datasets	Methods									
		CBM3D	WNNM	NCSR	MLP	DnCNN-B	FFDNet	FFDNet$_v$	FFDNet$_e$	UDNet	VDN
Case 1	Set5	27.76	26.53	26.62	27.26	29.87	*30.16*	30.15	27.90	28.13	**30.39**
	LIVE1	26.58	25.27	24.96	25.71	28.81	*28.99*	28.96	27.02	27.19	**29.22**
	BSD68	26.51	25.13	24.96	25.58	28.72	*28.78*	28.77	26.89	27.13	**29.02**
Case 2	Set5	26.34	24.61	25.76	25.73	29.05	*29.60*	29.56	25.87	26.01	**29.80**
	LIVE1	25.18	23.52	24.08	24.31	28.18	*28.58*	28.56	24.85	25.25	**28.82**
	BSD68	25.28	23.52	24.27	24.30	28.14	*28.43*	28.42	24.81	25.13	**28.67**
Case 3	Set5	27.88	26.07	26.84	26.88	29.17	*29.54*	29.49	27.60	27.54	**29.74**
	LIVE1	26.50	24.67	24.96	25.26	28.15	*28.39*	28.38	26.44	26.48	**28.65**
	BSD68	26.44	24.60	24.95	25.10	28.10	*28.22*	28.20	26.34	26.44	**28.46**

(a) (b) (c) (d) (e) (f)

Figure 3: Image denoising results of a typical test image in Case 2. (a) Noisy image, (b) Groundtruth, (c) CBM3D (24.63dB), (d) DnCNN-B (27.83dB), (e) FFDNet (28.06), (f) VDN (28.32).

Synthetic Experiments



Synthetic Experiments

Table 2: The PSNR(dB) results of all competing methods on AWGN noise cases of three test datasets.

Sigma	Datasets	Methods								
		CBM3D	WNNM	NCSR	MLP	DnCNN-B	FFDNet	FFDNet_e	UDNet	VDN
$\sigma = 15$	Set5	33.42	32.92	32.57	-	34.04	34.30	*34.31*	34.19	**34.34**
	LIVE1	32.85	31.70	31.46	-	33.72	**33.96**	**33.96**	33.74	33.94
	BSD68	32.67	31.27	30.84	-	*33.87*	33.85	33.68	33.76	**33.90**
$\sigma = 25$	Set5	30.92	30.61	30.33	30.55	31.88	*32.10*	32.09	31.82	**32.24**
	LIVE1	30.05	29.15	29.05	29.16	31.23	*31.37*	*31.37*	31.09	**31.50**
	BSD68	29.83	28.62	28.35	28.93	*31.22*	31.21	31.20	31.02	**31.35**
$\sigma = 50$	Set5	28.16	27.58	27.20	27.59	28.95	*29.25*	*29.25*	28.87	**29.47**
	LIVE1	26.98	26.07	26.06	26.12	27.95	*28.10*	*28.10*	27.82	**28.36**
	BSD68	26.81	25.86	25.75	26.01	27.91	*27.95*	*27.95*	27.76	**28.19**

Functions of The Objective Function

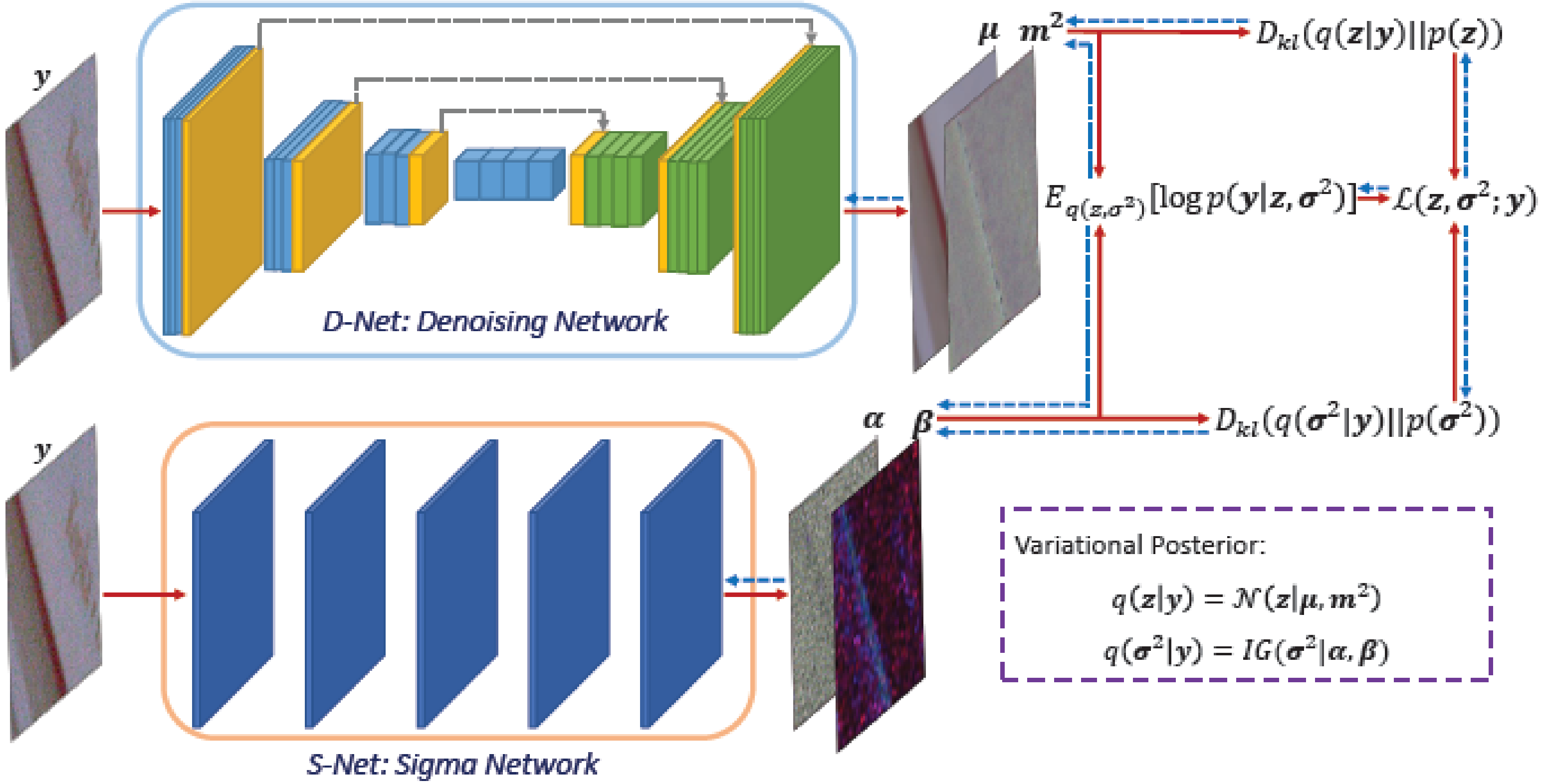


Table 4: PSNR results of different architecture combinations on Renoir Dataset.

Combinations	D-0	D-U	D-D	U-0	U-D	U-U
PSNR	38.51	**38.80**	38.68	39.11	**39.45**	39.35

Real Experiments

SIDD medium dataset:

- ✓ 320 real noisy images
- ✓ captured by 5 cameras
- ✓ under 10 scenes

Renoir dataset:

- ✓ 117 noisy and relatively low-noise image pairs under different scenes

Training data

SIDD validation set

DND dataset:

- ✓ 50 high-resolution images
- ✓ from 50 scenes
- ✓ taken by 4 consumer cameras

Test data

Real Experiments

Table 3: The PSNR (dB) results of all compared methods on SIDD Benchmark Dataset.

CBM3D	WNNM	MLP	DnCNN-B	CBDNet	VDN
25.65	25.78	24.71	23.66	33.28	39.02

Table 4: The PSNR (dB) results of all compared methods on SIDD validation set.

DnCNN-B	CBDNet	VDN
38.65	38.68	39.04

Table 5: The PSNR (dB) results of all competing methods on DND Benchmark Dataset.

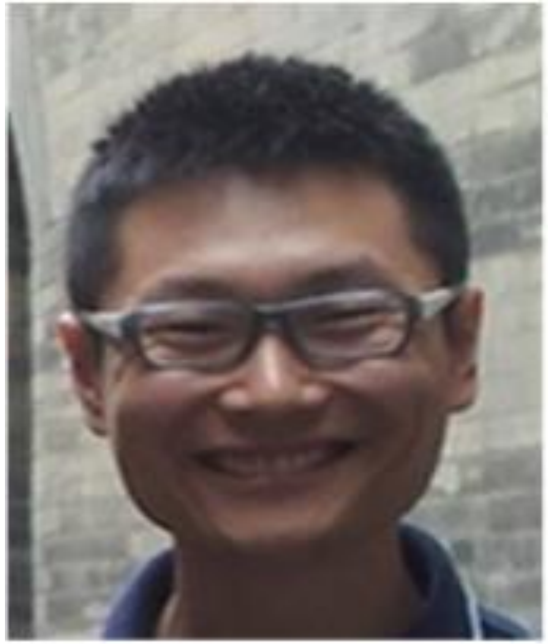
CBM3D	WNNM	NCSR	MLP	DnCNN-B	FFDNet	CBDNet	VDN
34.51	34.67	34.05	34.23	37.90	37.61	38.06	38.35

(a) (b) (c) (d) (e) (f)

Figure 4: Denoising results on one typical image in the validation set of SIDD. (a) Noisy image, (b) Simulated "clean" image, (c) WNNM(21.80dB), (d) DnCNN (34.48dB), (e) CBDNet (34.84dB), (d) VDN (35.50dB).

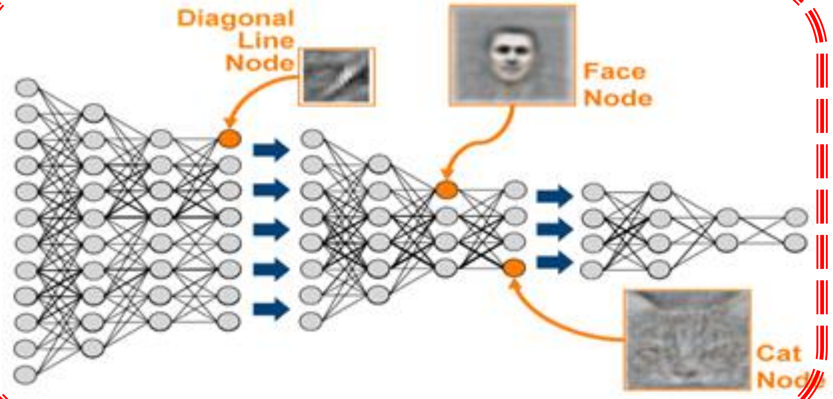
Summary

- **A new variational inference algorithm for blind image denoising**
  - Learn an approximate posterior to the true posterior with the latent variables (including clean image and noise variances) conditioned on the input noisy image
  - both tasks of blind image denoising and noise estimation can be naturally attained in a unique Bayesian framework
- **Open a new direction for noise modeling (noise inference)**
- **Extension to other low-level tasks: super-resolution, deblurring**

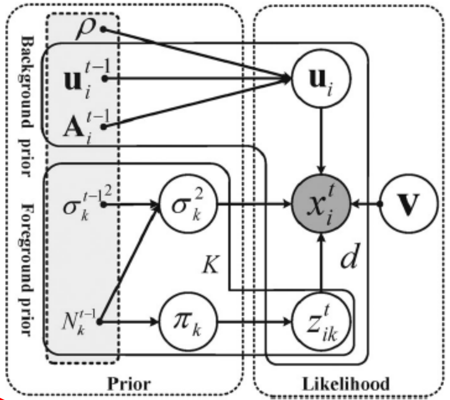


Data

Network



Model



$$\mathcal{L}(f(\mathbf{W}), \mathbf{X})$$

$$+$$

$$R(\mathbf{W})$$