NEURAL INFORMATION PROCESSING SYSTEMS

Diffusion Priors for Variational Likelihood Estimation and Image Denoising

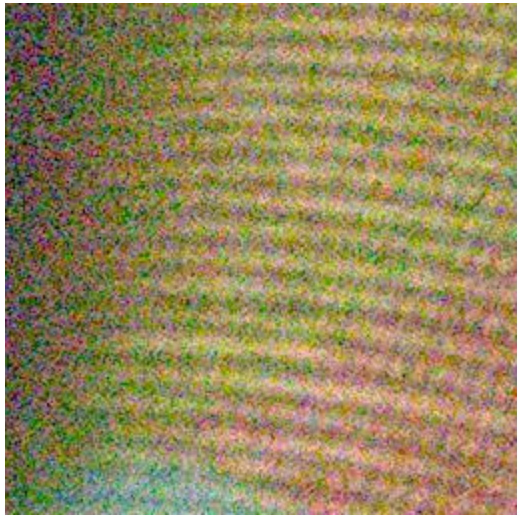
Jun Cheng, Shan Tan*
Huazhong University of Science and Technology, Wuhan, China

華中科技大學
HUAZHONG UNIVERSITY OF SCIENCE AND TECHNOLOGY

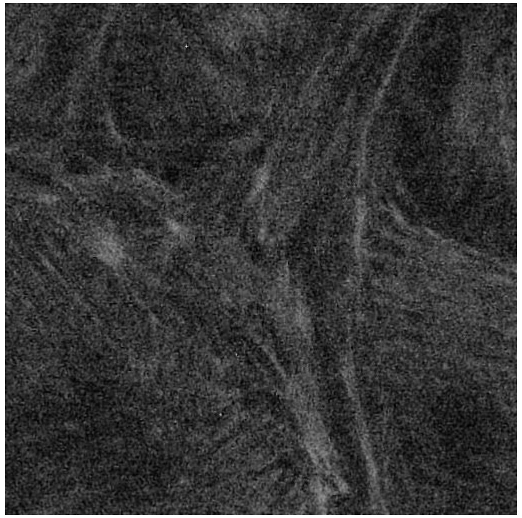
Introduction

Real-world Image Denoising:

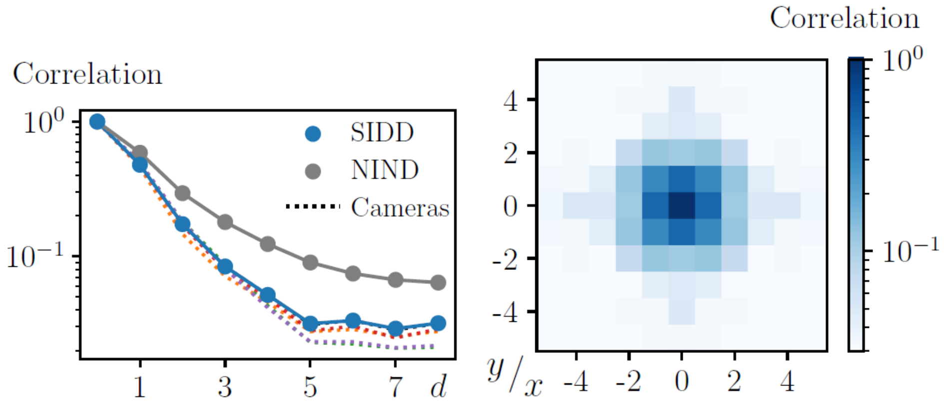
- Noise is complex: signal-dependent and spatially correlated.
- Supervised or self-supervised training require large amounts of (paired) images



Photography



Fluorescence microscopy



Noise Correlation [1]

Designing effective and data-efficient real-world denoising methods is important!

[1] Lee W, Son S, Lee K M. Ap-bsn: Self-supervised denoising for real-world images via asymmetric pd and blind-spot network. CVPR2022: 17725-17734.

Introduction

NEURAL INFORMATION PROCESSING SYSTEMS

Diffusion Priors for Image Restoration

- Well developed for linear degradations combined with Gaussian noise
- Struggle to handle complex or unknown noise types. e.g., DDRM, DDNM, DPS

Real-world Noise Model

- Can be approximated with a structured multivariate Gaussian (MVG)
- However, MVG has expensive and unknown covariance

Motivation: Injecting *i.ni.d.* likelihood and Variational Bayes into reverse diffusion process

Main Contribution

- We propose adaptive likelihood estimation and MAP inference based on diffusion priors and variational Bayes to address real-world complex noise.
- We explore the local prior exhibited by diffusion models pre-trained with LR images.

Method

NEURAL INFORMATION PROCESSING SYSTEMS

General conditional generation

Unconditional Diffusion

$$p(x_{0:T}) = p(x_T)\prod_{t=1}^{T} p(x_{t-1}|x_t), p(x_{t-1}|x_t) = \mathcal{N}(\mu_\theta(x_t,t), \sigma_t^2 I)$$

General zero-mean noise

$$y_0 = x_0 + n_0(x_0) \qquad \mathrm{corr}(n_0^i, n_0^j) > 0$$

Posterior:
Injecting y_0 into generation process

$$p(x_{0:T}|y_0) \to p(x_T)\prod_{t=1}^{T} p(x_{t-1}|x_t, y_{t-1}) \propto p(x_T)\prod_{t=1}^{T} p(x_{t-1}|x_t) p(y_{t-1}|x_{t-1})$$

Key !

$y_t = \sqrt{\bar{a}_t} y_0 + \sqrt{1-\bar{a}_t}\epsilon$, and $p(x_T|y_T) \approx p(x_T)$

Naive Image Denoising

Using structured Gaussian to model real-world noise

$$p(y_0|x_0) = \mathcal{N}(x_0, \Sigma(x_0)) \Longrightarrow p(y_{t-1}|x_{t-1}) = \mathcal{N}(y_{t-1}; x_{t-1}, \Sigma(x_{t-1})) \text{ with } \Sigma(x_{t-1}) = \overline{\alpha}_{t-1}\Sigma(x_0)$$

Promising (again the Gaussian form), but

- unknown covariance $\Sigma(x_0)$
- computationally expensive with dense covariance

Need methods to handle them!

Method

Variational Denoising with Adaptive Likelihood Estimation

NEURAL INFORMATION PROCESSING SYSTEMS

$$p(y_{t-1}|x_{t-1},\phi_{t-1}) = \mathcal{N}(y_{t-1}; x_{t-1}, \mathrm{diag}(\phi_{t-1})^{-1}), \phi_{t-1} = \frac{\phi_0}{\overline{\alpha}_{t-1}}$$

Diagonal precision matrix
Reducing complexity!

$$p(\phi_{t-1}) = \prod_{i=1}^{N} \mathrm{Gamma}(\phi_{t-1}^i; \alpha_{t-1}, \beta_{t-1}), \text{with } \alpha_{t-1} = \alpha, \beta_{t-1} = \beta\overline{\alpha}_{t-1}$$

Precision priors
Estimating posterior!

New Posterior: both x_t and ϕ_t

$$p(x_{t-1},\phi_{t-1}|x_t,y_{t-1}) = \frac{p(y_{t-1}|x_{t-1},\phi_{t-1})^{\frac{1}{\gamma}} p(\phi_{t-1}) p(x_{t-1}|x_t)}{p(y_{t-1}|x_t)}$$

$\gamma \leq 1$ is the temperature parameter

Variational approximation

Trivial distribution

$$g(x_{t-1},\phi_{t-1}) = g(x_{t-1})g(\phi_{t-1})$$

Iteration between:

Update $g(x_{t-1}) = \mathcal{N}(\hat{\mu}_{t-1}, \hat{\sigma}_{t-1}^2)$

Update $g(\phi_{t-1}) = \prod_{i=1}^{N} \mathrm{Gamma}(\hat{\alpha}_{t-1}^i, \hat{\beta}_{t-1}^i)$

Updated noise posterior $g(\phi_{t-1})$

Method

MAP estimation with updated likelihood

Given updated noise posterior $g(\phi_{t-1})$

Updated likelihood $p(y_{t-1}|x_{t-1}) = \mathrm{E}_{\phi_{t-1}\sim g(\phi_{t-1})} p(y_{t-1}|x_{t-1}, \phi_{t-1})$

$$
\begin{aligned}
x^*_{t-1} &= \text{argmax } \log p(y_{t-1}|x_{t-1}) + \log p(x_{t-1}|x_t) \\
&\approx \text{argmax } \mathrm{E}_{\phi_{t-1}} \log p(y_{t-1}|x_{t-1}, \phi_{t-1}) + \log p(x_{t-1}|x_t) \\
&= \text{argmax } -(x_{t-1} - y_{t-1})^2 \mathrm{E}(\phi_{t-1}) - \frac{(x_{t-1} - \mu_\theta(x_t, t))^2}{\sigma_t^2}
\end{aligned}
$$

Combination between observation and prior

$$
= \hat{\pi}_{t-1} y_{t-1} + (1 - \hat{\pi}_{t-1}) \mu_\theta(x_t, t), \text{ with } \hat{\pi}_{t-1} = \frac{\sigma_t^2}{\sigma_t^2 + 1/\mathrm{E}(\phi_{t-1})}
$$

MAP inference to get optimal x^*_{t-1} at each step

Rectification of $1/\mathrm{E}(\phi_{t-1})$

$$
\overline{1/\mathrm{E}(\phi_{t-1})} = \mathrm{Conv}(1/\mathrm{E}(\phi_{t-1}), G(l, s)), \hat{\pi}_{t-1} = \frac{\sigma_t^2}{\sigma_t^2 + \overline{1/\mathrm{E}(\phi_{t-1})}}
$$

NEURAL INFORMATION PROCESSING SYSTEMS

Method

NEURAL INFORMATION PROCESSING SYSTEMS

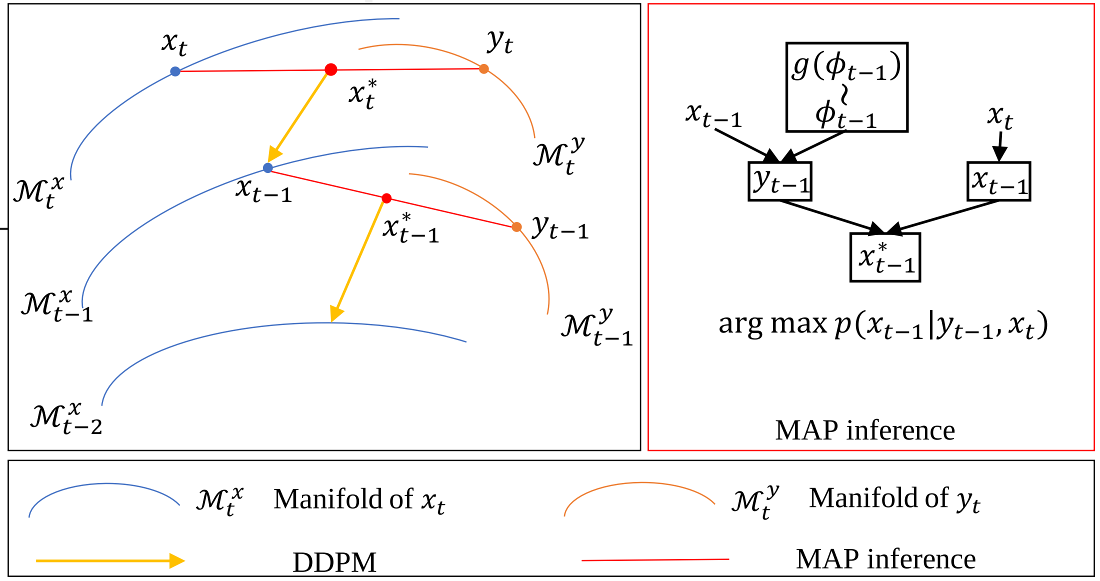
Whole procedure

Algorithm 1 Difusion priors-based variational image denoising

Input: Pre-trained diffusion model, noisy observation y_0, hyperparameters α, β, temperature γ

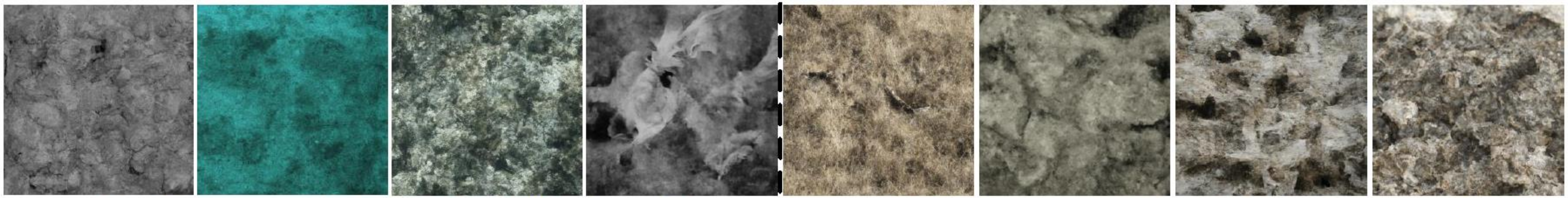
```
x_T ~ N(0, I), E(φ_T) = 1⃗
for t = T, ··· , 1 do
    Compute μ_θ(x_t, t) based on Eq. (4); Compute y_{t−1} based on Eq. (7)
    Set μ̂^{old}_{t−1} = 0⃗, μ̂_{t−1} = μ_θ(x_t, t)
    while ||μ̂^{old}_{t−1} − μ̂_{t−1}||_2^2 ≥ 1e^{−6} do
        Update g(x_{t−1}) = N(μ̂_{t−1}, σ̂^2_{t−1}) using Eq. (12)
        Update g(φ_{t−1}) = ∏_{i=1}^{N} Gamma(α̂^i_{t−1}, β̂^i_{t−1}) using Eq. (14)
    end while
    Solve optimal x_{t−1} using Eq. (15) or Eq. (17)
end for
return x_0
```

Visual undertanding



Method

Local Diffusion Priors



256×256 images sampled from 128×128 diffusion model   512×512 images sampled from 256×256 diffusion model

- Generated textures in HR images from LR diffusion model mainly focus on local areas
- local property of LR diffusion models is similar to traditional TV priors and Markov random fields

Pre-trained LR diffusion prior
For HR noisy images is directly feasible!

NEURAL INFORMATION PROCESSING SYSTEMS

Experiments

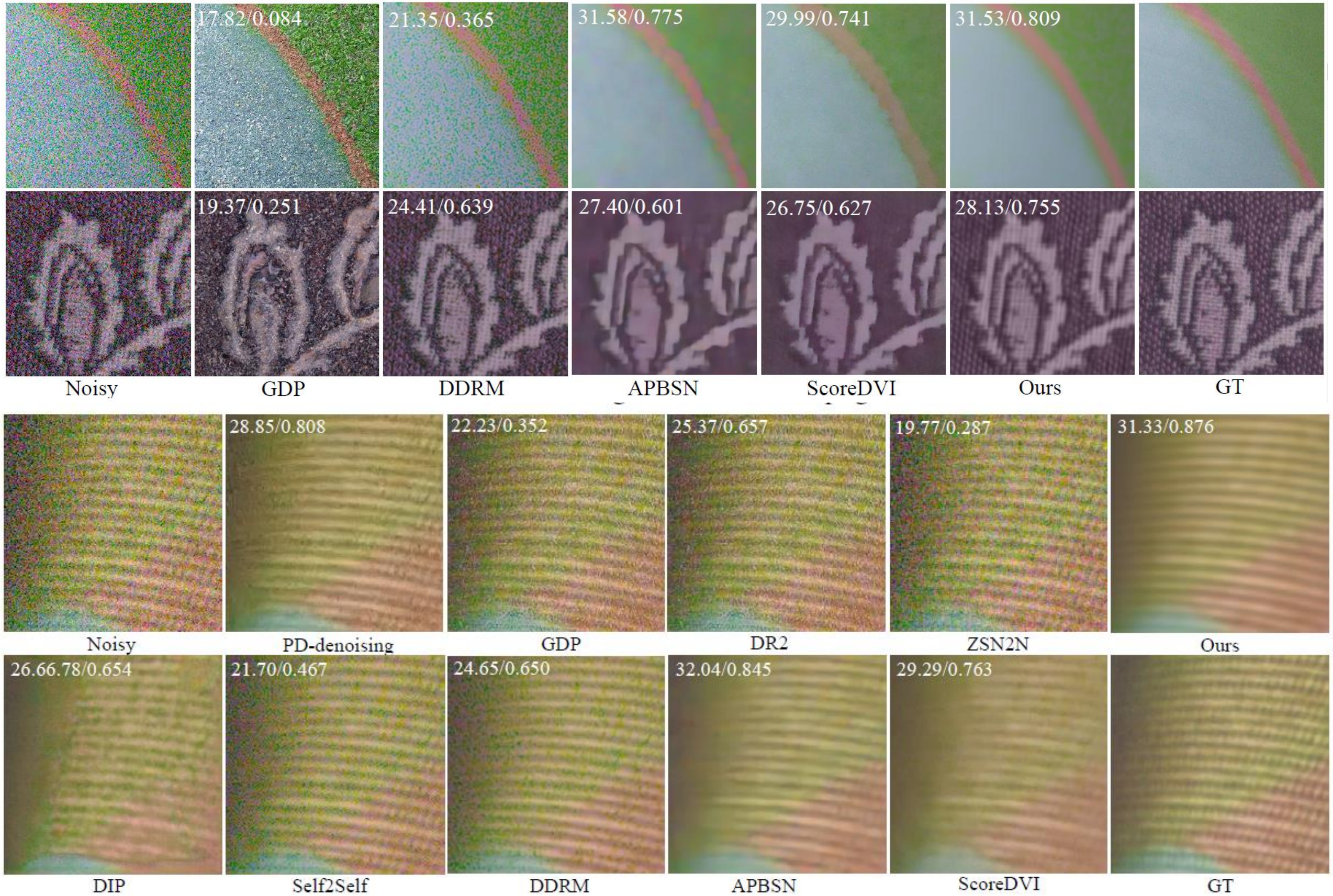
Real-world denoising

Table 2: Quantitative comparisons (PSNR(dB)/SSIM) of different methods on diverse real-world image datasets. The best and second-best PSNR/SSIM results are marked in **bold** and underlined.

Methods	SIDD Validation [1]	FMDD [51]	PolyU [47]	CC [30]	Average
DIP [42]	32.11/0.740	32.90/0.854	37.17/0.912	35.61/0.912	34.45/0.855
Self2Self [36]	29.46/0.595	30.76/0.695	38.33/0.962	37.45/0.948	34.00/0.800
PD-denoising [52]	33.97/0.820	33.01/0.856	37.04/0.940	35.85/0.923	34.97/0.885
ZS-N2N [27]	25.58/0.433	31.61/0.767	36.05/0.916	33.58/0.854	31.71/0.743
ScoreDVI [7]	34.75/0.856	33.10/**0.865**	37.77/0.959	37.09/0.945	35.68/0.906
GDP [12]	27.65/0.615	27.68/0.698	32.30/0.905	31.45/0.916	29.77/0.784
DR2 [45]	32.02/0.728	30.52/0.813	34.37/0.925	32.30/0.876	32.30/0.836
DDRM [19]	33.14/0.796	32.54/0.837	33.14/0.767	36.04/0.923	33.72/0.831
Ours	34.76/**0.887**	**33.14**/0.860	**38.71/0.970**	**38.01/0.959**	**36.16/0.919**
APBSN [23]	**36.80**/0.874	31.99/0.836	37.03/0.951	34.88/0.925	35.18/0.897

- Our method performs best among zero-shot methods
- Our approach effectively removes severe noise while preserving image details and textures.

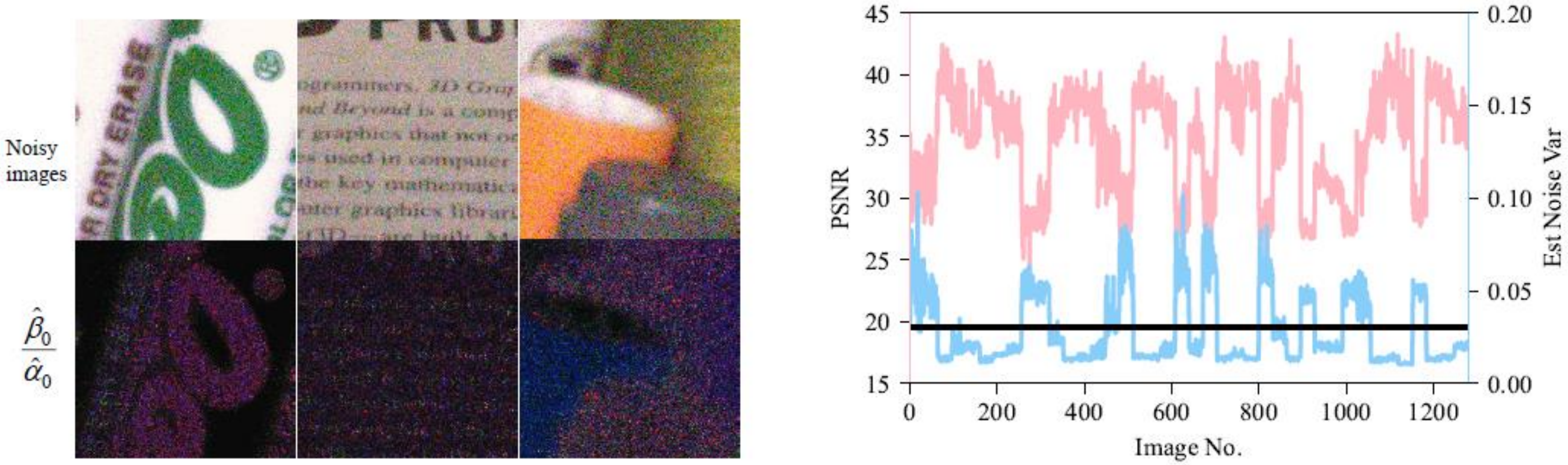
Experiments



Visual results

Experiments

Ablations on adaptive likelihood estimation and local Gaussian convolution



(a) Visual results of $\hat{\beta}_0/\hat{\alpha}_0$

(b) PSNR vs average $\hat{\beta}_0/\hat{\alpha}_0$.

Figure 4: The estimated noise variance $1/\mathrm{E}(\phi_0) = \hat{\beta}_0/\hat{\alpha}_0$ on SIDD dataset

Noise is adaptively estimated

with ALE	with Gaussian Conv	SIDD	FMDD	PolyU	CC
✗	✗	32.12/0.741	27.07/0.530	35.40/0.895	33.10/0.830
✓	✗	34.63/0.870	33.11/**0.865**	38.70/0.969	37.82/0.956
✓	✓	**34.76/0.887**	**33.14**/0.860	**38.71/0.970**	**38.01/0.959**

Experiments

NEURAL INFORMATION PROCESSING SYSTEMS

Ablations on Local diffusion priors

Res.: Train → Test	SIDD	Res.: Train → Test	CC	PolyU	FMDD
128 → 256	**34.80**/0.836	256 → 512	**38.01/0.959**	**38.71/0.970**	**33.14/0.860**
256 → 256	34.76/**0.887**	512 → 512	37.01/0.950	38.33/0.966	33.02/0.859

LR diffusion performs best generally

Application to other non-Gaussian noises

CBSD68 [28]	Poisson ($\lambda = 30$)	Bernoulli ($p = 0.2$)	KodaK [13]	Poisson ($\lambda = 30$)	Bernoulli ($p = 0.2$)
ZS-N2N	27.55/0.781	20.20/**0.828**	ZS-N2N	28.09/0.750	19.98/**0.820**
Ours	**29.24/0.833**	**26.11**/0.784	Ours	**30.56/0.839**	**27.17**/0.799

Application to demosaicing

Table 8: Results of image demosaicing

Dataset	Set14	CBSD68
DDRM	24.68/0.714	24.52/0.705
Ours	**26.02/0.756**	**25.43/0.732**

Thanks!