Blind2Sound: Self-Supervised Image Denoising without Residual Noise

Jiazheng Liu[1,†] Zejin Wang[1,†] Bohao Chen[2,1] Hua Han[1,3,*]

[1]State Key Laboratory of Brain Cognition and Brain-inspired Intelligence Technology, Institute of Automation,
Chinese Academy of Sciences, Beijing, China

[2]School of Advanced Interdisciplinary Sciences, University of Chinese Academy of Sciences, Beijing, China

[3]School of Future Technology, University of Chinese Academy of Sciences, Beijing, China

{liujiazheng2018,wangzejin2018,chenbohao2024,hua.han}@ia.ac.cn

A. Calculation Details on Adaptive Re-Visible Loss

A.1. Model Formation

Given the noise-corrupted observation $\boldsymbol{y} \in \mathbb{R}^{n\times c}$, where $\boldsymbol{n}$ denotes the number of pixels per channel and c is the number of channels, we aim to learn the latent clean image $\boldsymbol{x} \in \mathbb{R}^{n\times c}$ directly from a single noisy image. Here, we consider the Poisson-Gaussian noise [2] inherent to real-world imaging sensors, then the corruption process becomes:

$$\boldsymbol{y} = \alpha P + N, \tag{1}$$

where $P \sim \text{Poisson}(\boldsymbol{x}/\alpha)$ is the signal-dependent Poisson noise due to the photon counting, $N \sim \mathcal{N}(0, \sigma^2)$ is the Gaussian noise accounts for the signal-independent errors such as electric and thermal noise. Note that α is a scaling factor that depends on the quantum efficiency and analog gain. For simplicity, the Poisson noise is approximated as signal-dependent Gaussian noise [3], and the final corruption can be reformulated as:

$$\boldsymbol{y} = \boldsymbol{x} + \mathcal{N}(0, \alpha\boldsymbol{x} + \sigma^2), \tag{2}$$

We would like to extend the re-visible transition to sense noise and achieve personalized noise removal while retaining lossless denoising. Meanwhile, the auxiliary branch used for noise estimation should be removed during inference. To this aim, we reconsider the re-visible scheme from Bayesian reasoning. First, we model $p(\boldsymbol{z_1}|\boldsymbol{\Omega_y})$ as a multivariate Gaussian which represents that the latent clean image $\boldsymbol{z_1}$ is generated from the masked noisy volume $\boldsymbol{\Omega_y}$ as follows:

$$\boldsymbol{z_1} \sim \mathcal{N}(\boldsymbol{z_1}|\boldsymbol{\mu_m}, \boldsymbol{\Sigma_m}), \tag{3}$$

where $\mathcal{N}(\cdot|\boldsymbol{\mu_m}, \boldsymbol{\Sigma_m})$ denotes the multivariate Gaussian distribution with mean $\boldsymbol{\mu_m}$ and variance $\boldsymbol{\Sigma_m}$.

For the masked branch, Eq. (2) incorporates extra noise knowledge into the explicit corruption model, provided as the likelihood $p(\boldsymbol{y}|\boldsymbol{z_1})$ given a clean value. Therefore, the marginal likelihood of the noisy training data can be constructed via the distribution of unobserved clean data $\boldsymbol{z_1}$:

$$p(\boldsymbol{y_1}) = \int p(\boldsymbol{y}|\boldsymbol{z_1})p(\boldsymbol{z_1}|\boldsymbol{\Omega_y})d\boldsymbol{z_1}. \tag{4}$$

As illustrated in Eq. (4), when only noisy training data $\boldsymbol{y}$ are available, a known noise model are able to explicitly predict the masked prior $p(\boldsymbol{z_1}|\boldsymbol{\Omega_y})$. Specifically, for an approximate Gaussian noise model, the covariance of two mutually independent Gaussian convolutions is simply the sum of the components [1]. Hence, the marginal likelihood $p(\boldsymbol{y_1})$ is calculated in closed form, allowing to obtain the distribution of $\boldsymbol{z_1}$ by maximizing Eq. (4). According to Eqs. (2) and (3) as well as the above analysis, the mean and variance of $\boldsymbol{y_1}$ becomes:

$$\boldsymbol{y_1} \sim \mathcal{N}(\boldsymbol{y_1}|\boldsymbol{\mu_m}, \boldsymbol{\Sigma_m} + diag(\alpha_1\boldsymbol{\mu_m}) + \sigma_1\mathbf{I}). \tag{5}$$

[*]Corresponding author, † Equal contribution

For the visible branch, we construct $p(\boldsymbol{z_2}|\boldsymbol{y})$ as the generation of the latent clean image $\boldsymbol{z_2}$ from the raw noise image $\boldsymbol{y}$, which then becomes:

$$\boldsymbol{z_2} \sim \mathcal{N}(\boldsymbol{z_2}|\boldsymbol{\mu_v}, \boldsymbol{\Sigma_v}), \tag{6}$$

where the mean $\boldsymbol{\mu_v}$ and variance $\boldsymbol{\Sigma_v}$ are directly generated from the raw noise image $\boldsymbol{y}$ without gradient update. The marginal likelihood for the visible branch via the distribution of unobserved clean data $\boldsymbol{z_2}$ is then formulated as:

$$p(\boldsymbol{y_2}) = \int p(\boldsymbol{y}|\boldsymbol{z_2})p(\boldsymbol{z_2}|\boldsymbol{y})d\boldsymbol{z_2}. \tag{7}$$

Similar to Eq. (5), the mean and variance of $\boldsymbol{y_2}$ becomes:

$$\boldsymbol{y_2} \sim \mathcal{N}(\boldsymbol{y_2}|\boldsymbol{\mu_v}, \boldsymbol{\Sigma_v} + diag(\alpha_2\boldsymbol{\mu_v}) + \sigma_2\mathbf{I}). \tag{8}$$

We now have the marginal likelihood $p(\boldsymbol{y_1})$ for blind branch and $p(\boldsymbol{y_2})$ for visible branch. Moreover, the mean and variance of $\boldsymbol{z_2}$ are not involved in backpropagation because of identity mapping. Since only the blind distribution $p(\boldsymbol{z_1}|\boldsymbol{\Omega_y})$ via maximizing Eq. (4) has limited performance, we incorporate the errors of the visible branch $p(\boldsymbol{z_2}|\boldsymbol{y})$ into the mask gradient. Two-branch decorrelation enhances visible denoising without suppression from masked results. Hence, we model $\boldsymbol{y_1}$ and $\boldsymbol{y_2}$ as i.i.d., and apply Gaussian mixture to boost their representation. Combining Eqs. (5) and (8), the following enhanced target distribution $\boldsymbol{y}$ is obtained while retaining the independence of two branches:

$$\boldsymbol{y} \sim \sum_{i=1}^{2} \pi_i \cdot \mathcal{N}(\boldsymbol{y_i}|\boldsymbol{\mu_{y_i}}, \boldsymbol{\Sigma_{y_i}}), \tag{9}$$

where π_i is a hyper-parameter for the degree of re-visible. Besides, $0 \leq \pi_i \leq 1$ and $\pi_1 + \pi_2 = 1$. Set the blind factor π_1 to $1/(1+\lambda)$, the visible factor π_2 to $\lambda/(1+\lambda)$ and λ is a growing constant. Then, Eq. (9) is reformulated as:

$$\boldsymbol{y} \sim \mathcal{N}(\boldsymbol{y}|\frac{\boldsymbol{\mu_m} + \lambda\boldsymbol{\mu_v}}{1+\lambda}, \frac{\boldsymbol{\Sigma_{y_1}} + \lambda^2\boldsymbol{\Sigma_{y_2}}}{(1+\lambda)^2}). \tag{10}$$

We simply write $\boldsymbol{y} \sim \mathcal{N}(\boldsymbol{y}|\boldsymbol{\mu_y}, \boldsymbol{\Sigma_y})$ and denote the clean target image as $\boldsymbol{x}$. Since the mask mean $\boldsymbol{\mu_m}$ is just a lower bound of $\boldsymbol{x}$, the signal-dependent factor α_1 zooms this error. We replace the nosie model in Eq. (10) with a more accurate $p(\boldsymbol{y}|\boldsymbol{x})$ of zero mean and variance $diag(\alpha\boldsymbol{\mu_y}) + \pi_1^2\sigma_1\mathbf{I} + \pi_2^2\sigma_2\mathbf{I}$. The enhanced mixture marginal likelihood that bridges the blind and visible branches then becomes:

$$p(\boldsymbol{y}) = \int p(\boldsymbol{y}|\boldsymbol{x})p(\boldsymbol{x}|\boldsymbol{y}, \boldsymbol{\Omega_y})d\boldsymbol{x}. \tag{11}$$

To fit the observed noisy training data, we minimize its negative log-likelihood loss in the training phase as follows:

$$\begin{aligned} \mathcal{L}_{arv} &= -\log p(\boldsymbol{y}) = -\log\left[\pi_1 p(\boldsymbol{y_1}) + \pi_2 p(\boldsymbol{y_2})\right] \\ &= \frac{1}{2}[(\boldsymbol{y} - \boldsymbol{\mu_y})^T\boldsymbol{\Sigma_y}^{-1}(\boldsymbol{y} - \boldsymbol{\mu_y})] \\ &+ \frac{1}{2}\log|\boldsymbol{\Sigma_y}| + const, \end{aligned} \tag{12}$$

where $const$ is an additive constant term that can be discarded, $\mathcal{L}_{arv}$ denotes the proposed adaptive re-visible loss. When the denoiser converges, the following is the optimal clean value $\tilde{\boldsymbol{x}}$ of Eq. (12):

$$\tilde{\boldsymbol{x}} = \frac{\boldsymbol{\mu_m} + \lambda\boldsymbol{\mu_v}}{1+\lambda}. \tag{13}$$

A.2. Gradient of the Intermediate Medium

With the analytic form of adaptive re-visible loss in Eq. (12), we further explore and validate the collaborative mechanism between gradient update medium $\boldsymbol{\mu_m}$ and constants $\boldsymbol{\mu_v}$. Let $\boldsymbol{n_y} = \boldsymbol{y} - \boldsymbol{\mu_y}$, the derivative of the medium $\boldsymbol{\mu_m}$ gives its gradient:

$$
\begin{aligned}
d(tr(\mathcal{L}_{brv})) &= d(tr(\frac{1}{2}\boldsymbol{n_y}^T\boldsymbol{\Sigma_y}^{-1}\boldsymbol{n_y} + \frac{1}{2}\log|\boldsymbol{\Sigma_y}|)) \\
&= \frac{1}{2}tr(d\boldsymbol{n_y}^T\boldsymbol{\Sigma_y}^{-1}\boldsymbol{n_y} \\
&\qquad\qquad + \boldsymbol{n_y}^T\boldsymbol{\Sigma_y}^{-1}d\boldsymbol{n_y} + \boldsymbol{n_y}^T d(\boldsymbol{\Sigma_y}^{-1})\boldsymbol{n_y} + d(\log|\boldsymbol{\Sigma_y}|)) \\
&= \frac{1}{2}tr(\boldsymbol{n_y}^T\boldsymbol{\Sigma_y}^{-1}d\boldsymbol{n_y} \\
&\qquad\qquad + \boldsymbol{n_y}^T\boldsymbol{\Sigma_y}^{-1}d\boldsymbol{n_y} + \boldsymbol{n_y}\boldsymbol{n_y}^T d(\boldsymbol{\Sigma_y}^{-1}) + |\boldsymbol{\Sigma_y}|^{-1}d(|\boldsymbol{\Sigma_y}|)) \\
&= \frac{1}{2}tr(2\boldsymbol{n_y}^T\boldsymbol{\Sigma_y}^{-1}d\boldsymbol{n_y} \\
&\qquad\qquad + \boldsymbol{n_y}\boldsymbol{n_y}^T(-\boldsymbol{\Sigma_y}^{-1}d\boldsymbol{\Sigma_y}\boldsymbol{\Sigma_y}^{-1}) + |\boldsymbol{\Sigma_y}|^{-1}|\boldsymbol{\Sigma_y}|\boldsymbol{\Sigma_y}^{-1}d\boldsymbol{\Sigma_y}) \\
&= \frac{1}{2}tr(2\boldsymbol{n_y}^T\boldsymbol{\Sigma_y}^{-1}d\boldsymbol{n_y} \\
&\qquad\qquad + (\boldsymbol{\Sigma_y}^{-1} - \boldsymbol{\Sigma_y}^{-1}\boldsymbol{n_y}\boldsymbol{n_y}^T\boldsymbol{\Sigma_y}^{-1})d(diag(\alpha\boldsymbol{\mu_x}))) \\
&= \frac{1}{2}tr(2\boldsymbol{n_y}^T\boldsymbol{\Sigma_y}^{-1}d\boldsymbol{n_y} \\
&\qquad\qquad + (\boldsymbol{\Sigma_y}^{-1} - \boldsymbol{\Sigma_y}^{-1}\boldsymbol{n_y}\boldsymbol{n_y}^T\boldsymbol{\Sigma_y}^{-1})d(\alpha\mathbf{I}\odot\sqrt{\boldsymbol{\mu_x}\boldsymbol{\mu_x}^T})) \\
&= \frac{1}{2}tr(2\boldsymbol{n_y}^T\boldsymbol{\Sigma_y}^{-1}d\boldsymbol{n_y} \\
&\qquad\qquad + (\boldsymbol{\Sigma_y}^{-1} - \boldsymbol{\Sigma_y}^{-1}\boldsymbol{n_y}\boldsymbol{n_y}^T\boldsymbol{\Sigma_y}^{-1})(\alpha\mathbf{I}\odot d\sqrt{\boldsymbol{\mu_x}\boldsymbol{\mu_x}^T})) \\
&= \frac{1}{2}tr(2\boldsymbol{n_y}^T\boldsymbol{\Sigma_y}^{-1}d\boldsymbol{n_y} \\
&\qquad\qquad + ((\boldsymbol{\Sigma_y}^{-1} - \boldsymbol{\Sigma_y}^{-1}\boldsymbol{n_y}\boldsymbol{n_y}^T\boldsymbol{\Sigma_y}^{-1})\odot\alpha\mathbf{I})^T d\sqrt{\boldsymbol{\mu_x}\boldsymbol{\mu_x}^T})) \\
&= \frac{1}{2}tr(2\boldsymbol{n_y}^T\boldsymbol{\Sigma_y}^{-1}d\boldsymbol{n_y} \\
&\quad + ((\boldsymbol{\Sigma_y}^{-1} - \boldsymbol{\Sigma_y}^{-1}\boldsymbol{n_y}\boldsymbol{n_y}^T\boldsymbol{\Sigma_y}^{-1})\odot\alpha\mathbf{I})^T((\sqrt{\boldsymbol{\mu_x}\boldsymbol{\mu_x}^T})'\odot d(\boldsymbol{\mu_x}\boldsymbol{\mu_x}^T))) \\
&= \frac{1}{2}tr(2\boldsymbol{n_y}^T\boldsymbol{\Sigma_y}^{-1}d\boldsymbol{n_y} \\
&\quad + ((\boldsymbol{\Sigma_y}^{-1} - \boldsymbol{\Sigma_y}^{-1}\boldsymbol{n_y}\boldsymbol{n_y}^T\boldsymbol{\Sigma_y}^{-1})\odot\alpha\mathbf{I}\odot(\sqrt{\boldsymbol{\mu_x}\boldsymbol{\mu_x}^T})')^T d(\boldsymbol{\mu_x}\boldsymbol{\mu_x}^T)) \\
&= \frac{1}{2}tr(2\boldsymbol{n_y}^T\boldsymbol{\Sigma_y}^{-1}d\boldsymbol{n_y} \\
&\quad + 2\boldsymbol{\mu_x}^T((\boldsymbol{\Sigma_y}^{-1} - \boldsymbol{\Sigma_y}^{-1}\boldsymbol{n_y}\boldsymbol{n_y}^T\boldsymbol{\Sigma_y}^{-1})\odot\alpha\mathbf{I}\odot(\sqrt{\boldsymbol{\mu_x}\boldsymbol{\mu_x}^T})')^T d\boldsymbol{\mu_x}) \\
&= tr(-\frac{1}{(\lambda+1)}\boldsymbol{n_y}^T\boldsymbol{\Sigma_y}^{-1}d\boldsymbol{\mu_m} \\
&\quad + \frac{\alpha}{(1+\lambda)}\boldsymbol{\mu_x}^T((\boldsymbol{\Sigma_y}^{-1} - \boldsymbol{\Sigma_y}^{-1}\boldsymbol{n_y}\boldsymbol{n_y}^T\boldsymbol{\Sigma_y}^{-1})\odot\mathbf{I}\odot(\sqrt{\boldsymbol{\mu_x}\boldsymbol{\mu_x}^T})')^T d\boldsymbol{\mu_m}).
\end{aligned}
\tag{14}
$$

We then have,

$$
\begin{aligned}
\nabla_{\boldsymbol{\mu_m}} = \frac{\partial\mathcal{L}_{brv}}{\partial\boldsymbol{\mu_m}} &= -\frac{1}{1+\lambda}\boldsymbol{\Sigma_y}^{-1}\boldsymbol{n_y} \\
&\qquad + \frac{\alpha}{(\lambda+1)}(\boldsymbol{\Sigma_y}^{-1} - \boldsymbol{\Sigma_y}^{-1}\boldsymbol{n_y}\boldsymbol{n_y}^T\boldsymbol{\Sigma_y}^{-1})\odot\mathbf{I}\odot(\sqrt{\boldsymbol{\mu_x}\boldsymbol{\mu_x}^T})'\boldsymbol{\mu_x}.
\end{aligned}
\tag{15}
$$

According to Eq. (15), we observe that including the gradient update term $\boldsymbol{\mu_m}$ in $\mathrm{diag}(\alpha\boldsymbol{\mu_y})$ results in severe instability during training because of a complicated second term in the gradient. Therefore, we consider disabling the gradient of $\mathrm{diag}(\alpha\boldsymbol{\mu_y})$, which stabilizes the training process and enhances the performance of the denoiser. Moreover, we can perform denoising directly from the raw noise image $\boldsymbol{y}$ in the inference. Combined with the above analysis, the final gradient form for the medium $\boldsymbol{\mu_m}$ becomes:

$$\nabla_{\mu_m} = \frac{\partial \mathcal{L}_{brv}}{\partial \boldsymbol{\mu_m}} = -\frac{1}{1+\lambda}\boldsymbol{\Sigma_y}^{-1}\boldsymbol{n_y}. \tag{16}$$

Next, we also analyze the collaborative association between another gradient medium $\boldsymbol{\Sigma_m}$ and other variables:

$$\begin{aligned}
d(tr(\mathcal{L}_{brv})) &= d(tr(\frac{1}{2}\boldsymbol{n_y}^T\boldsymbol{\Sigma_y}^{-1}\boldsymbol{n_y} + \frac{1}{2}\log|\boldsymbol{\Sigma_y}|)) \\
&= \frac{1}{2}tr(\boldsymbol{n_y}^T d(\boldsymbol{\Sigma_y}^{-1})\boldsymbol{n_y} + d(\log|\boldsymbol{\Sigma_y}|)) \\
&= \frac{1}{2}tr(\boldsymbol{n_y}\boldsymbol{n_y}^T d(\boldsymbol{\Sigma_y}^{-1}) + |\boldsymbol{\Sigma_y}|^{-1}d(|\boldsymbol{\Sigma_y}|)) \\
&= \frac{1}{2}tr(\boldsymbol{n_y}\boldsymbol{n_y}^T(-\boldsymbol{\Sigma_y}^{-1}d\boldsymbol{\Sigma_y}\boldsymbol{\Sigma_y}^{-1}) + |\boldsymbol{\Sigma_y}|^{-1}|\boldsymbol{\Sigma_y}|\boldsymbol{\Sigma_y}^{-1}d\boldsymbol{\Sigma_y})) \\
&= \frac{1}{2}tr(-\boldsymbol{\Sigma_y}^{-1}\boldsymbol{n_y}\boldsymbol{n_y}^T\boldsymbol{\Sigma_y}^{-1}d\boldsymbol{\Sigma_y} + \boldsymbol{\Sigma_y}^{-1}d\boldsymbol{\Sigma_y}) \\
&= tr(\frac{1}{2(1+\lambda)^2}(\boldsymbol{\Sigma_y}^{-1} - \boldsymbol{\Sigma_y}^{-1}\boldsymbol{n_y}\boldsymbol{n_y}^T\boldsymbol{\Sigma_y}^{-1})d\boldsymbol{\Sigma_m}).
\end{aligned} \tag{17}$$

We then have,

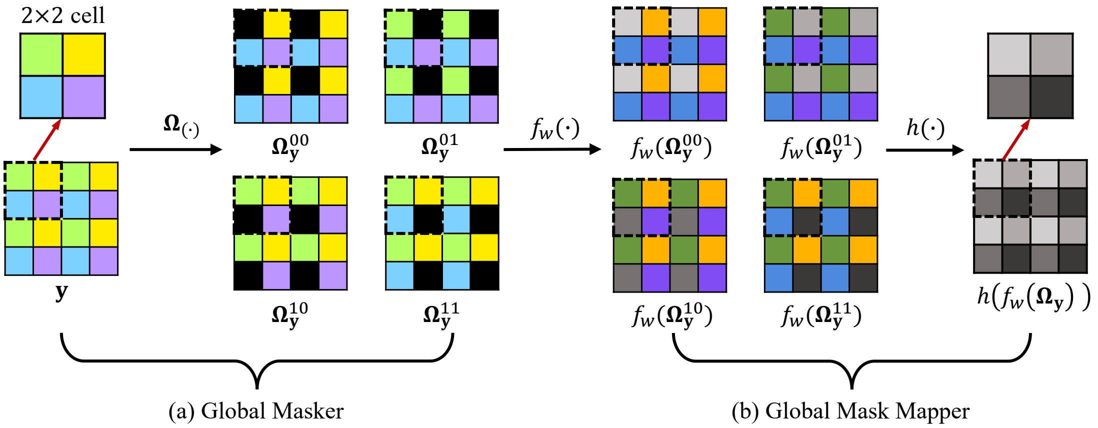
$$\nabla_{\boldsymbol{\Sigma}_m} = \frac{\partial \mathcal{L}_{brv}}{\partial \boldsymbol{\Sigma_m}} = \frac{1}{2(\lambda+1)^2}\left(\boldsymbol{\Sigma_y}^{-1} - \boldsymbol{\Sigma_y}^{-1}\boldsymbol{n_y}\boldsymbol{n_y}^{\mathrm{T}}\boldsymbol{\Sigma_y}^{-1}\right). \tag{18}$$

Obviously, the gradient of the intermediate medium $\boldsymbol{\Sigma_m}$ in Eq. (18) is conventional and there is no additional instability term as in Eq. (15).

B. Additional Implementation Details

B.1. Global Masker and Global Mask Mapper

As mentioned in the paper, we use the same global masker and mask mapper as Blind2Unblind [4]. Figure 1 shows the details of the global masker and mask mapper. Specifically, the global masker divides the raw noise image $\boldsymbol{y}$ into 2×2 cells and constructs a mask volume $\boldsymbol{\Omega_y}$ via occluding all pixels at the same location on each cell. The global mask mapper samples the pixels of the denoised mask volume where the blind spot is located and projects them to the same channel according to their position to form a denoised image. Code: https://anonymous.4open.science/r/Blind2Sound-1F33/.


2×2 cell
y
$\boldsymbol{\Omega}_{(\cdot)}$
$\boldsymbol{\Omega}_{\mathbf{y}}^{00}$
$\boldsymbol{\Omega}_{\mathbf{y}}^{01}$
$\boldsymbol{\Omega}_{\mathbf{y}}^{10}$
$\boldsymbol{\Omega}_{\mathbf{y}}^{11}$
$f_w(\cdot)$
$f_w(\boldsymbol{\Omega}_{\mathbf{y}}^{00})$
$f_w(\boldsymbol{\Omega}_{\mathbf{y}}^{01})$
$f_w(\boldsymbol{\Omega}_{\mathbf{y}}^{10})$
$f_w(\boldsymbol{\Omega}_{\mathbf{y}}^{11})$
$h(\cdot)$
$h(f_w(\boldsymbol{\Omega}_{\mathbf{y}}))$

(a) Global Masker

(b) Global Mask Mapper

Figure 1. Details of the global masker and mask mapper.

B.2. Training Settings

The covariance determines the output channels of the denoising network: two for grayscale images, nine for sRGB images, and fifteen for raw-RGB images. For training the denoising network, the noise estimator and the denoising network are jointly optimized simultaneously. The weights of the re-visible loss and Cramer Gaussian loss are set to 1 and 0.01, respectively.

C. More Experimental Results

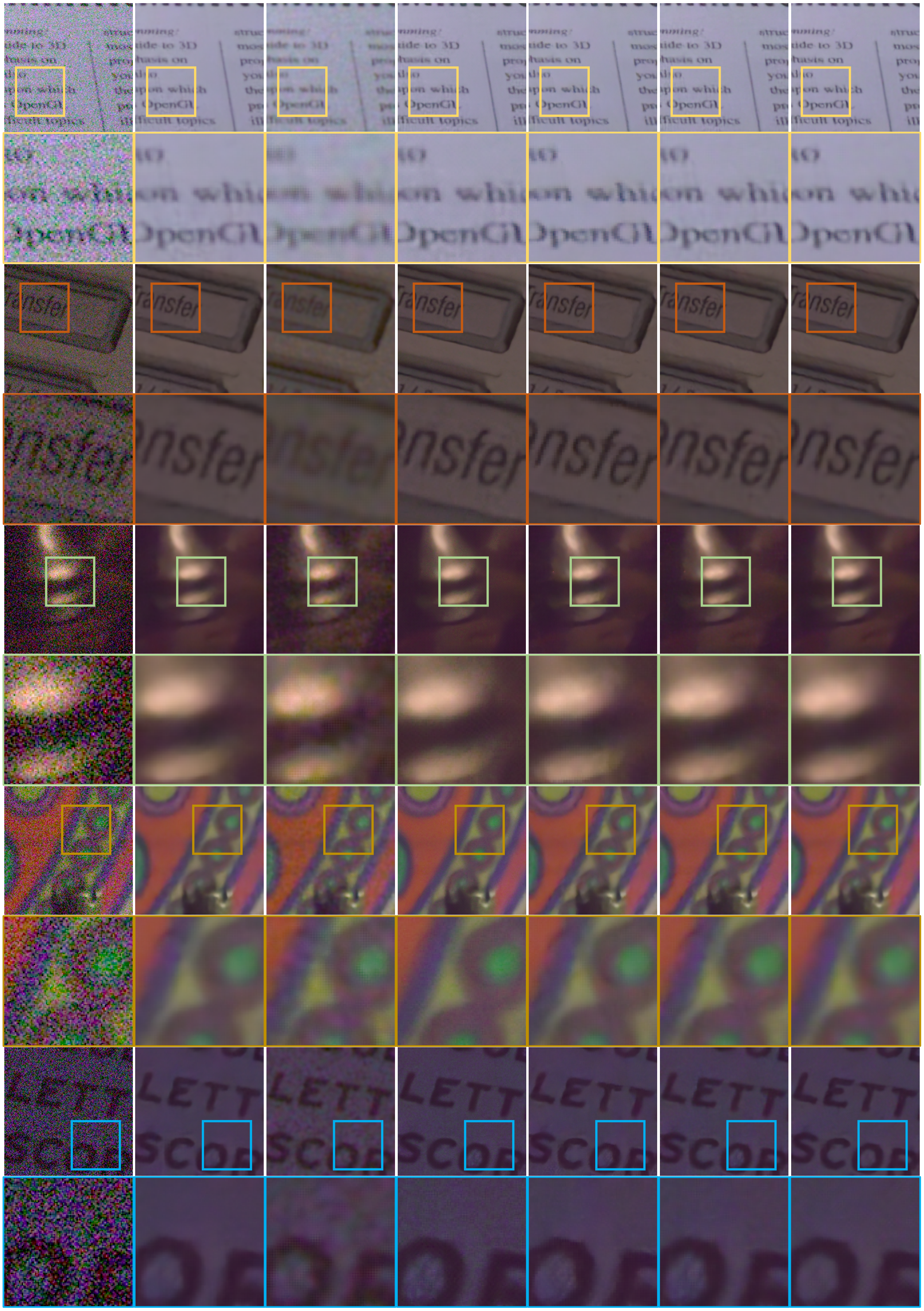


Figure 2. Visualization results on real-world SIDD Benchmark.

References

[1] Paul A Bromiley. Products and convolutions of gaussian distributions. *Medical School, Univ. Manchester, Manchester, UK, Tech. Rep*, 3:2003, 2003. 1

[2] Alessandro Foi, Mejdi Trimeche, Vladimir Katkovnik, and Karen Egiazarian. Practical poissonian-gaussian noise modeling and fitting for single-image raw-data. *TIP*, 17(10):1737–1754, 2008. 1

[3] Samuel W Hasinoff. Photon, poisson noise. *Computer Vision: A Reference Guide*, pages 608–610, 2014. 1

[4] Zejin Wang, Jiazheng Liu, Guoqing Li, and Hua Han. Blind2unblind: Self-supervised image denoising with visible blind spots. In *CVPR*, 2022. 4