Noise Distribution Adaptive Self-Supervised Image Denoising using Tweedie Distribution and Score Matching: Supplementary Material

Kwanyoung Kim[1] Taesung Kwon[2] Jong Chul Ye[1,2]
[1]Kim Jaechul Graduate School of AI, [2] Department of Bio and Brain Engineering, KAIST
{cubeyoung, star.kwon, jong.ye}@kaist.ac.kr

A. Proof of Proposition 1

Proof. For a given exponential family of probability distributions:

$$p(y|\mu) = p_0(y)\exp\left(\mu^\top T(y) - A(\mu)\right) \ , \tag{1}$$

Tweedie's formula [2] shows that the posterior estimate of the canonical parameter $\hat{\mu}$ should satisfy the following equation:

$$\hat{\mu}^\top T'(y) \quad = \quad -l_0'(y) + l'(y) \tag{2}$$

where $l'(y) := \nabla_y \log p(y)$ and $l_0'(y) := \nabla_y \log p_0(y)$ are score functions, and $T'(y) = \nabla_y T(y)$ [3].

Now, our goal is to use this formula to the saddle point approximation of Tweedie distribution [1] given by:

$$p(y;\mu,\phi) = (2\pi\phi y^\rho)^{-\frac{1}{2}}\exp\left(\frac{-d(y,\mu)}{2\phi}\right) \tag{3}$$

where

$$d(y,\mu) = 2\left(\frac{y^{2-\rho}}{(1-\rho)(2-\rho)} - \frac{y\mu^{1-\rho}}{1-\rho} + \frac{\mu^{2-\rho}}{2-\rho}\right). \tag{4}$$

By inspection of (1) and (3), we have

$$\begin{aligned} p_0(y) &= (2\pi\phi y^\rho)^{-\frac{1}{2}} \\ \mu^\top T(y) - A(\mu) &= \frac{-d(y,\hat{\mu})}{2\phi} \end{aligned} \tag{5}$$

Furthermore, we have

$$\begin{aligned} \frac{\partial d(y,\mu)}{\partial y} &= \frac{2}{1-\rho}y^{1-\rho} - \frac{2}{1-\rho}\mu^{1-\rho} \\ \frac{\partial \log(2\pi\phi y^\rho)^{-\frac{1}{2}}}{\partial y} &= -\frac{\rho}{2y} \end{aligned}$$

Accordingly,

$$\begin{aligned} \hat{\mu}^\top T'(y) &= -\frac{1}{\phi(1-\rho)}(y^{1-\rho} - \hat{\mu}^{1-\rho}) \\ &= \frac{\rho}{2y} + l'(y) \end{aligned}$$

which leads to

$$\hat{\mu}^{1-\rho} = y^{1-\rho} + \phi(1-\rho)\left(\frac{\rho}{2y} + l'(y)\right). \tag{6}$$

Therefore, we have

$$\begin{aligned} \hat{\mu} &= \exp\left\{\frac{1}{1-\rho}\log\left(y^{1-\rho} + \phi(1-\rho)\left(\frac{\rho}{2y} + l'(y)\right)\right)\right\} \\ &= y\left(1 + (1-\rho)\alpha(y,\rho,\phi)\right)^{\frac{1}{1-\rho}} \end{aligned} \tag{7}$$

where

$$\alpha(y,\rho,\phi) = \phi y^{\rho-1}\left(\frac{\rho}{2y} + l'(y)\right). \tag{8}$$

This concludes the proof. □

B. Proof of Proposition 2

Proof:

Additive Gaussian noise. In this case, we have $\rho = 0, \phi = \sigma^2$ for Tweedie distribution. Accordingly, (8) can be simplified as

$$\alpha(y,0,\phi) = \sigma^2 y^{-1} l'(y) \tag{9}$$

Therefore, using (7), we have

$$\hat{\mu} = y(1 + \sigma^2 y^{-1} l'(y)) = y + \sigma^2 l'(y).$$

Poisson noise. In this case, we have $\rho = 1, \phi = \zeta$ for Tweedie distribution. In this case, we have

$$\begin{aligned} &\lim_{\rho\to 1}(1 + (1-\rho)\alpha(y,\rho,\phi))^{\frac{1}{1-\rho}} \\ &= \exp\left[\lim_{\rho\to 1}\frac{\log(1 + (1-\rho)\alpha(y,\rho,\phi))}{1-\rho}\right] \\ &= \exp\left[\lim_{\rho\to 1}\frac{\alpha(y,\rho,\phi)}{1 + (1-\rho)\alpha(y,\rho,\phi)}\right] \\ &= \exp[\alpha(y,1,\phi)]. \end{aligned} \tag{10}$$

where the second equality comes from the L'Hospital's rule, and

$$\alpha(y,1,\phi)=\zeta\left(\frac{1}{2y}+l'(y)\right) \tag{11}$$

Therefore, we have

$$\begin{aligned}\hat{\mu} &= y\exp\left[\zeta\left(\frac{1}{2y}+l'(y)\right)\right]\\ &= y\exp(\frac{\zeta}{2y})\exp(\zeta l'(y))\\ &\approx (y+\frac{\zeta}{2})\exp(\zeta l'(y)),\end{aligned}$$

where the last approximation comes from $\exp(x)\approx 1+x$ for a small x.

Gamma noise. In this case, we have $\rho=2,\phi=1/k$ for Tweedie distribution. Using (8), we have

$$\alpha(y,2,k)=\frac{1}{k}y\left(\frac{1}{y}+l'(y)\right)=\frac{1}{k}(1+yl'(y)) \tag{12}$$

Therefore, we have

$$\begin{aligned}\hat{\mu} &= y\left(1-\frac{1}{k}(1+yl'(y))\right)^{-1}\\ &= \frac{ky}{(k-1)-yl'(y))}\end{aligned}$$

This concludes the proof.

C. Proof of Proposition 3

Proof. Let $y_2=y_1+\epsilon u$ for $u\sim\mathcal{N}(0,1)$. Then, we have

$$\alpha(y_1,\rho,\phi) = \phi y_1^{\rho-1}\left(\frac{\rho}{2y_1}+l'(y_1)\right) \tag{13}$$

$$\alpha(y_2,\rho,\phi) = \phi y_2^{\rho-1}\left(\frac{\rho}{2y_2}+l'(y_2)\right) \tag{14}$$

For a sufficiently small perturbation ϵ, we can assume that

$$\alpha(y_1,\rho,\phi)=\alpha(y_1,\rho,\phi)$$

Accordingly, by dividing (14) by (13), we have

$$1 = \left(\frac{y_2}{y_1}\right)^{\rho-1}\left(\frac{\frac{\rho}{2y_2}+l'(y_2)}{\frac{\rho}{2y_1}+l'(y_1)}\right)$$

By taking logarithm on both sides, we have

$$(\rho-2)\log\left(\frac{y_2}{y_1}\right)+\log\left(\frac{\rho+2y_2l'(y_2)}{\rho+2y_1l'(y_1)}\right)=0 \tag{15}$$

Furthermore, by denoting $w:=2y_2l'(y_2)-2y_1l'(y_1)$, we have

$$\begin{aligned}\log\left(\frac{\rho+2y_2l'(y_2)}{\rho+2y_1l'(y_1)}\right) &= \log\left(1+\frac{w}{\rho+2y_1l'(y_1)}\right)\\ &\approx \frac{w}{p+2y_1l'(y_1)}\end{aligned} \tag{16}$$

when $\frac{w}{\rho+2y_1l'(y_1)}\to 0$. By plugging (16) into (15), we can obtain the quadratic equation for ρ:

$$0=(\rho-2)\log\left(\frac{y_2}{y_1}\right)+\frac{2y_2l'(y_2)-2y_1l'(y_1)}{\rho+2y_1l'(y_1)}$$

By denoting $a=\log(\frac{y_2}{y_1}), b=2y_1l'(y_1)$, we have

$$\begin{aligned}0 &= a(\rho-2)+\frac{w}{\rho+b},\\ &= a(\rho-2)(\rho+b)+w.\end{aligned} \tag{17}$$

Therefore, the estimated noise model parameter $\hat{\rho}$ can be obtained as the solutions for the quadratic equation, which is given by:

$$\hat{\rho}=\frac{-a(b-2)\pm\sqrt{(a(b-2)^2-4a(-2ab+w)}}{2a}$$

□

D. Proof of Proposition 4

Proof. Let $y_2=y_1+\epsilon u$ for $u\sim\mathcal{N}(0,1)$, and the noise model parameter ρ is known. Suppose furthermore that ϵ is non-zero but sufficiently small that the following equality holds:

$$\mathbb{E}[\mu|y_1]=\mathbb{E}[\mu|y_2] \tag{18}$$

Now, we derive the formula for each distribution.

Additive Gaussian noise For the case of additive Gaussian noise, we have

$$\hat{x}_1=\mathbb{E}[\mu|y_1]=y_1+\sigma^2l'(y_1) \tag{19}$$

$$\hat{x}_2=\mathbb{E}[\mu|y_2]=y_2+\sigma^2l'(y_2) \tag{20}$$

By subtracting (19) from (20), we have

$$-\epsilon u=\sigma^2(l'(y_2)-l'(y_1)) \tag{21}$$

Thus, we have the following estimate:

$$\hat{\sigma}^2 = \frac{-\epsilon u}{l'(y_2)-l'(y_1)}$$

Poisson noise In this case, we have

$$\hat{x}_1 = \left(y_1 + \frac{\zeta}{2}\right) \exp(\zeta l'(y_1)) \tag{22}$$

$$\hat{x}_2 = \left(y_2 + \frac{\zeta}{2}\right) \exp(\zeta l'(y_2)), \tag{23}$$

By taking the logarithm of both equations and subtracting (22) from (23), we have

$$\begin{aligned} 0 &= \log\left(1 + \frac{\epsilon u}{y_1 + \zeta/2}\right) + \zeta(l'(y_2) - l'(y_1)) \\ &\approx \left(\frac{\epsilon u}{y_1 + \zeta/2}\right) + \zeta(l'(y_2) - l'(y_1)), \end{aligned}$$

where the last approximation comes from $x \approx \log(1+x)$ for sufficiently small x. This leads to the follwoing quadratic equation for ζ:

$$0 = \epsilon u + \zeta(y_1 + \zeta/2)(l'(y_2) - l'(y_1)). \tag{24}$$

Solving quadratic equation (24), we can obtain the following estimate:

$$\hat{\zeta} \quad = \quad y_1 + \sqrt{y_1^2 - 2c}$$

where $c = \epsilon u / \left(l'(y_2) - l'(y_1)\right)$.

Gamma noise For the case of Gamma noise,

$$\hat{x}_1 = \frac{k y_1}{k - 1 - y_1 l'(y_1)} \tag{25}$$

$$\hat{x}_2 = \frac{k y_2}{k - 1 - y_2 l'(y_2)} \tag{26}$$

By taking the inverse of both equations and subtracting (25) from (26), we have

$$\frac{1}{y_2} - \frac{1}{y_1} = \frac{1}{k}\left(\frac{1}{y_2} - \frac{1}{y_1} + l'(y_2) - l'(y_1)\right).$$

Then, we can obtain $\hat{k}$ by

$$\hat{k} \quad = \quad 1 + \frac{l'(y_2) - l'(y_1)}{\frac{1}{y_2} - \frac{1}{y_1}}$$

$\square$

E. Pseudocode Description

Algorithm 1 details the overall pipeline of the training procedure for the proposed method. First, the neural network R_Θ was trained by minimizing $\ell_{AR-DAE}(\Theta)$ to learn the estimation of the score function from the noisy input y. In the training, noisy images are sampled from an unknown noise model corrupted with various noise levels. This neural network training step is universally applied regardless of noise distribution. In particular, we annealed σ_a from σ_a^{max} to σ_a^{min} to stably train the network as suggested in [6]. Now let Θ' be an independent copy of the parameters and after the n^{th} training iteration, and we update the Θ' with the exponential moving average as indicated in Algorithm 1 as suggested in [6].

The inference of the proposed method is described in Algorithm 2. After we obtain the trained score models $R^*_{\Theta'}$, we firstly estimate the noise model parameter $\hat{\rho}$ with Equation (10) in the main paper using Proposition 3. Once the noise model is determined, we estimate the noise level parameter for the estimated noise model using Proposition 4. Then, the final clean image is reconstructed by Tweedie's formula as indicated in Table 2 in the main paper.

Algorithm 1: Training procedure of the proposed method

Given: learning rates γ, number of epochs N;
Input : noisy input y from training data set D_ϕ and noise level parameter $\phi \in (\sigma, \zeta, k)$, neural network R_Θ, independent copy of the parameter Θ', annealing sigma set S_{σ_a} with size of T, $S_{\sigma_a} = [\sigma_a^{min}, ..., \sigma_a^{max}]$, decay rate of exponential moving average m

for $n = 1$ *to* N **do**
  $u \sim \mathcal{N}(0, 1)$;
  $t \sim \mathcal{U}(0, T)$;
  $\sigma_a \to S^t_{\sigma_a}$
  $\ell_{AR-DAE}(\Theta) = \underset{\substack{y \sim P_Y \\ u \sim \mathcal{N}(0,I), \sigma_a \sim \mathcal{N}(0,\delta^2)}}{\mathbb{E}} \|u + \sigma_a R_\Theta(y + \sigma_a u)\|^2$;
  $\Theta \leftarrow \Theta - \gamma \nabla_\Theta \ell_{AR-DAE}(\Theta)$;
  $\Theta' \leftarrow m\Theta' - (1 - m)\Theta$;

Output: Trained the score model, $R^*_{\Theta'}(y) = \hat{l}'(y)$

F. Implementation Details

Training details To robustly train the proposed method, we randomly injected the perturbed noise into a noisy image instead of using linear scheduling as in [3]. In the case of the Gaussian and Gamma noise, σ_a^{max} and $\sigma_a^{\min}$ are set to [0.1,0.001], respectively. For the Poisson noise case, σ_a^{max} and $\sigma_a^{\min}$ was set to [0.1,0.02], respectively.

Noise model estimation In order to satisfy the assumption in (16), we only calculate the pixel values that satisfies

Algorithm 2: Inference procedure of the proposed method

Given: Trained score model $R^*_{\Theta'}$, the perturbed noise level ϵ;

Input : noisy input y_1 from training data set D_ϕ and noise level parameter $\phi \in (\sigma, \zeta, k)$, and generated perturbed noisy image $y_2 = y_1 + \epsilon\mu,\ \mu \sim \mathcal{N}(0, I)$;

Noise model estiamtion: $\hat{\rho}$ by Equation (10) in the main paper

Noise level estiamtion:

1 **if** $0 \leq \hat{\rho} < 0.9 \rightarrow y \in$ Gaussian noise **then**

$\hat{\sigma}^2 = \text{median}\left(\frac{-\epsilon u}{l'(y_2) - l'(y_1)}\right)$;

Output : $\hat{x} = y + \hat{\sigma}^2 l'(y_1)$

2 **else if** $0.9 \leq \hat{\rho} < 1.9 \rightarrow y \in$ Poisson noise **then**

$\hat{\zeta} = \text{median}\left(-y_1 + \sqrt{y_1^2 - 2c}\right)$;

Output : $\hat{x} = \left(y + \frac{\hat{\zeta}}{2}\right) \exp^{\hat{\zeta} l'(y)}$

3 **else if** $1.9 \leq \hat{\rho} < 2.9 \rightarrow y \in$ Gamma noise **then**

$\hat{k} = \text{median}\left(1 + \frac{l'(y_2) - l'(y_1)}{\frac{1}{y_2} - \frac{1}{y_1}}\right)$;

Output : $\hat{x} = \frac{\hat{k} y}{(\hat{k} - 1) - y l'(y)}$

the following condition:

$$idx = -\epsilon < \frac{w}{\rho + b} < \epsilon \tag{27}$$

where ϵ was set to 1×10^{-5} for all of the cases. In the procedure of calculating (27), we can not access $\hat{\rho}$ value. Hence, based on assumption $\rho \in (0, 2)$, we empirically determine this value. In the case of additive Gaussian noise, this value is set to 2.5, otherwise to 2.2. We provide the implementation code based on Pytorch [5] as shown in Listing 1.

Listing 1. Source code of the proposed noise model esimation

```
def noise_model_estimation(y_1,score_model):
    # Inject noise into noisy images y_1
    epsilon = 1e-5
    n = torch.randn(y_1.shape)
    noise = epsilon * n
    y_2 = y_1 + noise
    # esimate the score functions
    l(y_1) = score_model(y_1)
    l(y_2) = score_model(y_2)
    # calculate each coefficient
    w = 2*(y_2*l(y_2) - y_1*l(y_1))
    a = torch.log(y_2/y_1)
    b = (2*y_1*l(y_1))
    # take only values under condition
    ww = w/(b+2.2)
    idx = (ww <= 1e-5) & (ww >= -1e-5)
    w = w[idx]
    b = b[idx]
    w = torch.nanmean(w)
    b = torch.nanmean(b)
    # Solve quadratic equation
    first = a*(b-2)
    second = 4*a*(- 2*a*b + w)
    sqrt = (first)**2 - second
    sqrt = torch.sqrt(sqrt)
    p1 = (-first + sqrt)/(2*a)
    p2 = (-first - sqrt)/(2*a)
    p1 = torch.nanmean(p1)
    p2 = torch.nanmean(p2)
    # take maximum of two values
    p = max(p1,p2)
    # take maximum of p and 0
    p = max(p,0)
    return p
```

Noise level estiamtion To use Proposition 4, we assume that the injected small noise is sufficiently small, the equality in (18) holds. To achieve this, we set ϵ to $1\times\ 10^{-5}$ for all noise cases.

G. Analysis for Noise Model Estimation

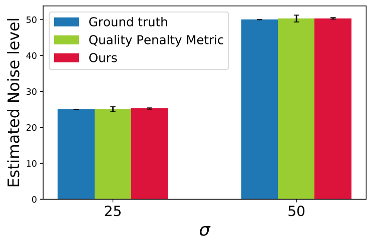
Table 1 shows the accuracy of the estimated noise model in the Kodak dataset. If the estimated noise distributions are equal to the truth noise distributions, we determine that the estimate is correct. We found that the proposed noise model estimation reach 100% accuracy for all cases. Thus, we concluded that we can successfully estimate the noise model with the proposed method.

Table 1. Accuracy of the proposed noise model estimation in the Kodak dataset.

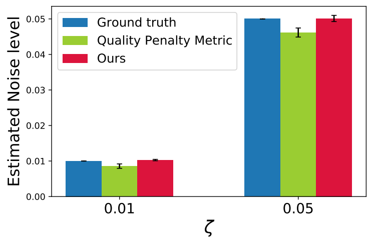
Noise type	Noise level	Accuracy(%)
Gaussian	$\sigma = 25$	100
	$\sigma = 50$	100
Poisson	$\zeta = 0.01$	100
	$\zeta = 0.05$	100
Gamma	$k = 100$	100
	$k = 50$	100

H. Analysis for Noise Level Estimation

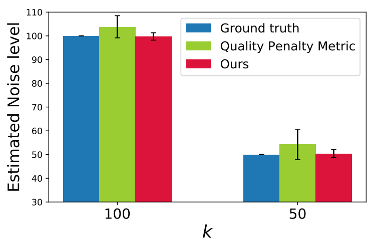
Fig. 1 shows the bar graph of estimated noise level parameters for each noise distribution in the Kodak dataset. Similar to the ablation study in the main paper, we carried out the analysis by fixing the noise model estimation and by only varying the estimation of noise level. From the ablation study in the main paper, we expect that the quality penalty metrics method in [3] estimates correctly in the case of additive Gaussian noise, but incorrectly for the case of other noise distributions. We can observe the similar findings in Figure 1. On the other hand, the proposed noise level



(a)



(b)



(c)

Figure 1. Estimated noise level parameters for each noise distribution in Kodak dataset. (a) Gaussian (σ =25, and 50). (b) Poisson (ζ = 0.01, and 0.05). (c) Gamma (k= 100, 50). The blue bars indicate the truth noise levels. The green bars and the red bars indicate that the average of estimated noise levels with the quality penalty metric in [3] and the proposed method, respectively.

estimation provides more accurate results compared to the quality penalty metrics in [3] with a small standard deviation. Thus, we can conclude that the proposed noise level estimation can successfully estimate the truth noise level in all noise distribution cases.

I. Ablation Study on Score Estimation

We demonstrate the effectiveness of each component in improving the score model. Table 2 compares the PSNR values of the results with and without each component on CBSD68 dataset (Poisson noise ζ = 0.01). EMA and GS denote Exponential Moving Average and Geometric Sequence, respectively. To fairly compare each case, we performed the ablation study using the same procedure. From the table, we observe the performance degradation when any component of the proposed method is absent. Therefore, we can conclude that EMA and GS in the proposed method are essential for improving the score model in the training procedure of the network.

Table 2. Ablation studies on score estimation using CBSD68 data (Poisson noise cases, $\zeta = 0.01$).

Component	Ours	Case1	Case2	Case3
EMA	✓	✓	✗	✗
GS	✓	✗	✓	✗
PSNR(dB)	**32.53**	32.41	32.23	32.03

J. Additional Experiment on Mixed Poisson-Gaussian Noise Case

To show the effectiveness of the proposed method for the case of mixed noise, we carried out the experiments on Set12 dataset in case of mixed Poisson-Gaussian noise as shown in Table 3. We confirm that our method outperforms the other methods in mixed noise case. We observed that when the Poisson componets are dominating, the input can be interpreted as Poisson noise whereas the slightly overestimated noisy level parameter ($\hat{\zeta}$ from 0.01 to 0.013) compensate for the Gaussian part.

Table 3. The quantitative results using the various methods in terms of PSNR(dB) for the case of mixed Poisson-Gaussian noise on the Set12 dataset.

Dataset	Noise type	N2V	N2S	Laine	Nei2Nei	Ours	Supervised
Set12	$(\zeta,\sigma) = (0.01,5)$	28.44	29.78	31.43	31.33	**31.46**	31.59
	$(\zeta,\sigma) = (0.01,15)$	28.12	29.26	30.53	30.35	**30.58**	30.98

K. Ablation Study on Hyperparameter of Noise Model Estimation

In this section, we analyzed the sensitivity with respect to the noise model estimation. In estimating the noise model, we determined the specific noise type when the noise model parameter, $\hat{\rho}$, belong to a certain range that is associated with the hyperparameter. For the analysis of hyperparameter of noise model estimation, we carried out experimentes in which the rules of the range that determine the noise model are varied in Kodak dataset. As shown in Table 4, narrower selection bins reduce the accuracy, and we thus chose the best range empirically.

L. Ablation Study on Hyperparameter of Noise Level Estimation

We also analyzed the sensitivity of hyperparameter of noise level estimation, ϵ, in Propoisition 4. In the Proposition 4, we assume that the injected small noise, ϵ is sufficiently small, the equality in (18) holds. For the anlaysis of the acceptable noise level for ϵ, we carried out the experiments in which the indepdent noise level ϵ was varied

Table 4. The ablation studies of noise model parameter, ρ, by varing the rule of the range.

Noise type	Start point	End point of range			
Gaussian	$0 \sim$	0.9	0.7	0.5	0.3
	$\sigma = 25$	100%	95.83%	79.16%	66.66%
Poisson	$0.9 \sim$	1.9	1.7	1.5	1.3
	$\zeta = 0.01$	100%	100%	100%	75%
Gamma	$1.9 \sim$	2.9	2.7	2.5	2.3
	$k = 25$	100%	100%	79.16%	16.66%

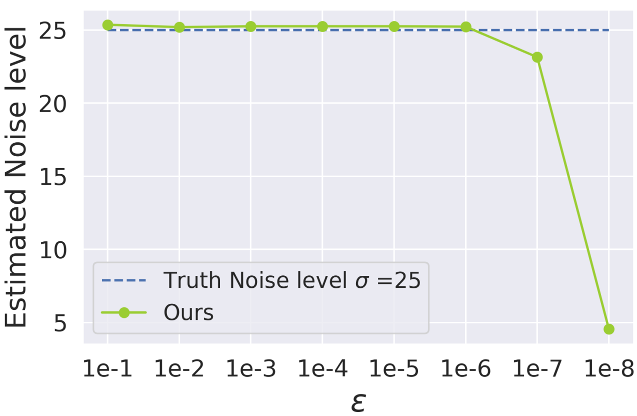


Figure 2. The analysis on hyperparameter for noise level estimation.

for the case of Gaussian ($\sigma = 25$) in CBSD68 dataset. We fixed the other experiment setting and only varied the noise level ϵ in the noise level estimation procedure. As shown in Fig. 2, we found that the range [1×10^{-1},1×10^{-6}] is acceptable.

M. Qualitative Results

We provide more examples of the denoising results by various methods on the AAPM CT dataset [4] as shown in Fig. 3. Similar to the results shown in the main paper, the improvements are consistent. In particular, other self-supervised approaches produce over-smooth denoised images, which produces the boundary structure of CT images in difference images. On the other hand, the proposed method provides similar results compared to target images and only generates noise components in the difference images.

N. Negative Societal Impacts

As a negative societal impact, the failure of image denoising methods can lead to side effects. For example, removing both the noise and the texture of the medical images could lead to misdiagnosis.

References

[1] Peter K Dunn and Gordon K Smyth. Tweedie family densities: methods of evaluation. In *Proceedings of the 16th international workshop on statistical modelling, Odense, Denmark*, pages 2–6. Citeseer, 2001. 1

[2] Bradley Efron. Tweedie's formula and selection bias. *Journal of the American Statistical Association*, 106(496):1602–1614, 2011. 1

[3] Kwanyoung Kim and Jong Chul Ye. Noise2Score: Tweedie's Approach to Self-Supervised Image Denoising without Clean Images. *arXiv preprint arXiv:2106.07009*, 2021. 1, 3, 4, 5

[4] Cynthia H McCollough, Adam C Bartley, Rickey E Carter, Baiyu Chen, Tammy A Drees, Phillip Edwards, David R Holmes III, Alice E Huang, Farhana Khan, Shuai Leng, et al. Low-dose CT for the detection and classification of metastatic liver lesions: results of the 2016 low dose CT grand challenge. *Medical physics*, 44(10):e339–e352, 2017. 6

[5] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017. 4

[6] Yang Song and Stefano Ermon. Improved techniques for training score-based generative models. *arXiv preprint arXiv:2006.09011*, 2020. 3

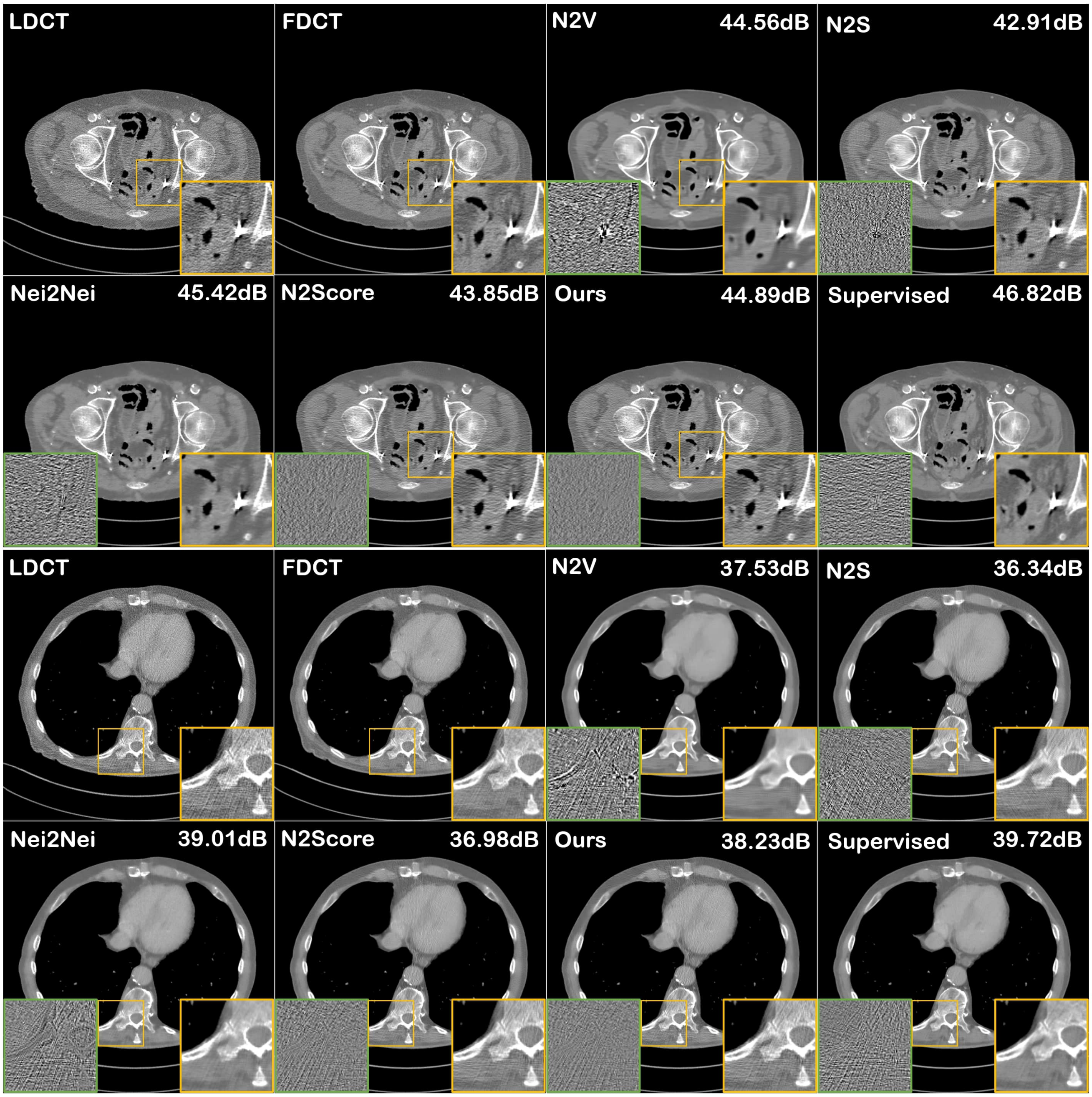


Figure 3. Denoising results of AAPM data using various methods. The yellow box and green box show the enlarged view of image and difference image between network input and output, respectively. The intensity window of CT image is (-500,500)[HU] and the intensity window of difference is (-200,200) [HU].