SeNM-VAE: Semi-Supervised Noise Modeling with Hierarchical Variational Autoencoder

Supplementary Material

1. Detailed derivation

The derivation of Equation 4 in the main paper is elucidated in detail herein. By introducing an inference model $q\left(\mathbf{z}, \mathbf{z_n} | \mathbf{x}, \mathbf{y}\right)$, we decompose $\log p\left(\mathbf{y}|\mathbf{x}\right)$ into the following two terms:

$$\begin{aligned} \log p\left(\mathbf{y}|\mathbf{x}\right) = &\mathbb{E}_{q\left(\mathbf{z},\mathbf{z_n}|\mathbf{x},\mathbf{y}\right)} \log \frac{p\left(\mathbf{y}, \mathbf{z}, \mathbf{z_n}|\mathbf{x}\right)}{q\left(\mathbf{z}, \mathbf{z_n}|\mathbf{x}, \mathbf{y}\right)} \\ &+ \left(\log p\left(\mathbf{y}|\mathbf{x}\right) - \mathbb{E}_{q\left(\mathbf{z},\mathbf{z_n}|\mathbf{x},\mathbf{y}\right)} \log \frac{p\left(\mathbf{y}, \mathbf{z}, \mathbf{z_n}|\mathbf{x}\right)}{q\left(\mathbf{z}, \mathbf{z_n}|\mathbf{x}, \mathbf{y}\right)}\right), \end{aligned} \tag{1}$$

where the first term represents the cELBO. The second term can be expressed as follows:

$$\begin{aligned} &\log p\left(\mathbf{y}|\mathbf{x}\right) - \mathbb{E}_{q\left(\mathbf{z},\mathbf{z_n}|\mathbf{x},\mathbf{y}\right)} \log \frac{p\left(\mathbf{y}, \mathbf{z}, \mathbf{z_n}|\mathbf{x}\right)}{q\left(\mathbf{z}, \mathbf{z_n}|\mathbf{x}, \mathbf{y}\right)} \\ =&\mathbb{E}_{q\left(\mathbf{z},\mathbf{z_n}|\mathbf{x},\mathbf{y}\right)} \left[\log p\left(\mathbf{y}|\mathbf{x}\right) - \log \frac{p\left(\mathbf{y}|\mathbf{x}\right) p\left(\mathbf{z}, \mathbf{z_n}|\mathbf{x}, \mathbf{y}\right)}{q\left(\mathbf{z}, \mathbf{z_n}|\mathbf{x}, \mathbf{y}\right)}\right] \\ =&\mathbb{E}_{q\left(\mathbf{z},\mathbf{z_n}|\mathbf{x},\mathbf{y}\right)} \log \frac{q\left(\mathbf{z}, \mathbf{z_n}|\mathbf{x}, \mathbf{y}\right)}{p\left(\mathbf{z}, \mathbf{z_n}|\mathbf{x}, \mathbf{y}\right)} \\ =&D_{\mathrm{KL}}\left(q\left(\mathbf{z}, \mathbf{z_n}|\mathbf{x}, \mathbf{y}\right) || p\left(\mathbf{z}, \mathbf{z_n}|\mathbf{x}, \mathbf{y}\right)\right). \end{aligned} \tag{2}$$

According to the proposed graphical model (as depicted in Figure 1a in the main paper), we have

$$\begin{aligned} p\left(\mathbf{y}, \mathbf{z}, \mathbf{z_n}|\mathbf{x}\right) =&p\left(\mathbf{z}|\mathbf{x}\right) p\left(\mathbf{z_n}|\mathbf{x}, \mathbf{z}\right) p\left(\mathbf{y}|\mathbf{x}, \mathbf{z}, \mathbf{z_n}\right) \\ =&p\left(\mathbf{z}|\mathbf{x}\right) p\left(\mathbf{z_n}|\mathbf{z}\right) p\left(\mathbf{y}|\mathbf{z}, \mathbf{z_n}\right), \\ p\left(\mathbf{z}, \mathbf{z_n}|\mathbf{x}, \mathbf{y}\right) =&p\left(\mathbf{z}|\mathbf{x}, \mathbf{y}\right) p\left(\mathbf{z_n}|\mathbf{x}, \mathbf{y}, \mathbf{z}\right) \\ =&p\left(\mathbf{z}|\mathbf{x}, \mathbf{y}\right) p\left(\mathbf{z_n}|\mathbf{y}, \mathbf{z}\right). \end{aligned} \tag{3}$$

To maintain consistency with the decomposition of $p\left(\mathbf{z}, \mathbf{z_n}|\mathbf{x}, \mathbf{y}\right)$, we choose

$$q(\mathbf{z}, \mathbf{z_n}|\mathbf{x}, \mathbf{y}) = q(\mathbf{z}|\mathbf{x}, \mathbf{y}) q(\mathbf{z_n}|\mathbf{y}, \mathbf{z}). \tag{4}$$

Consequently, the cELBO can be further factorized as

$$\begin{aligned} &\mathbb{E}_{q\left(\mathbf{z},\mathbf{z_n}|\mathbf{x},\mathbf{y}\right)} \log \frac{p\left(\mathbf{y}, \mathbf{z}, \mathbf{z_n}|\mathbf{x}\right)}{q\left(\mathbf{z}, \mathbf{z_n}|\mathbf{x}, \mathbf{y}\right)} \\ =&\mathbb{E}_{q\left(\mathbf{z},|\mathbf{x},\mathbf{y}\right)q\left(\mathbf{z_n}|\mathbf{y},\mathbf{z}\right)} \log \frac{p\left(\mathbf{z}|\mathbf{x}\right) p\left(\mathbf{z_n}|\mathbf{z}\right) p\left(\mathbf{y}|\mathbf{z}, \mathbf{z_n}\right)}{q\left(\mathbf{z}|\mathbf{x}, \mathbf{y}\right) q\left(\mathbf{z_n}|\mathbf{y}, \mathbf{z}\right)} \\ =&\mathbb{E}_{q\left(\mathbf{z},|\mathbf{x},\mathbf{y}\right)q\left(\mathbf{z_n}|\mathbf{y},\mathbf{z}\right)} \log p\left(\mathbf{y}|\mathbf{z_n}\right) + \mathbb{E}_{q\left(\mathbf{z},|\mathbf{x},\mathbf{y}\right)} \log \frac{p\left(\mathbf{z}|\mathbf{x}\right)}{q\left(\mathbf{z}|\mathbf{x}, \mathbf{y}\right)} \\ &+ \mathbb{E}_{q\left(\mathbf{z},|\mathbf{x},\mathbf{y}\right)q\left(\mathbf{z_n}|\mathbf{y},\mathbf{z}\right)} \log \frac{p\left(\mathbf{z_n}|\mathbf{z}\right)}{q\left(\mathbf{z_n}|\mathbf{y}, \mathbf{z}\right)} \\ =&\mathbb{E}_{q\left(\mathbf{z},|\mathbf{x},\mathbf{y}\right)q\left(\mathbf{z_n}|\mathbf{y},\mathbf{z}\right)} \log p\left(\mathbf{y}|\mathbf{z_n}\right) - D_{\mathrm{KL}}\left(q\left(\mathbf{z}|\mathbf{x}, \mathbf{y}\right) || p\left(\mathbf{z}|\mathbf{x}\right)\right) \\ &- \mathbb{E}_{q\left(\mathbf{z}|\mathbf{x},\mathbf{y}\right)} D_{\mathrm{KL}}\left(q\left(\mathbf{z_n}|\mathbf{y}, \mathbf{z}\right) || p\left(\mathbf{z_n}|\mathbf{y}, \mathbf{z}\right)\right). \end{aligned} \tag{5}$$

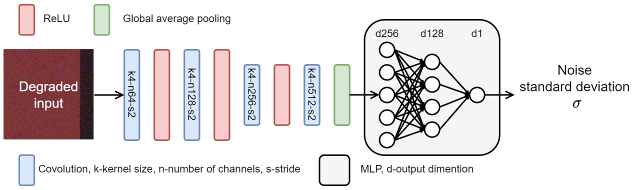


Figure 1. Architecture of degradation level prediction network.

2. Proof for Proposition 1

Proposition 1. *Let* $q(\mathbf{z}|\mathbf{x}, \mathbf{y})$ *be a mixture model of* $q(\mathbf{z}|\mathbf{x})$ *and* $q(\mathbf{z}|\mathbf{y})$*:*

$$q(\mathbf{z}|\mathbf{x}, \mathbf{y}) = p_1 q(\mathbf{z}|\mathbf{x}) + p_2 q(\mathbf{z}|\mathbf{y}), \tag{6}$$

then:

$$\begin{aligned} D_{\mathrm{KL}}(q(\mathbf{z}|\mathbf{x}, \mathbf{y}) \| p(\mathbf{z}|\mathbf{x})) \leq &p_1 D_{\mathrm{KL}}(q(\mathbf{z}|\mathbf{x}) \| p(\mathbf{z}|\mathbf{x})) \\ &+ p_2 D_{\mathrm{KL}}(q(\mathbf{z}|\mathbf{y}) \| p(\mathbf{z}|\mathbf{x})). \end{aligned} \tag{7}$$

Moreover, suppose that $q(\mathbf{z}|\mathbf{x}) = p(\mathbf{z}|\mathbf{x})$ *by sharing the same neural network. Then:*

$$D_{\mathrm{KL}}(q(\mathbf{z}|\mathbf{x}, \mathbf{y}) \| p(\mathbf{z}|\mathbf{x})) \leq p_2 D_{\mathrm{KL}}(q(\mathbf{z}|\mathbf{y}) \| q(\mathbf{z}|\mathbf{x})) \tag{8}$$

Proof. Using the log-sum inequality, we have:

$$\begin{aligned} &D_{\mathrm{KL}}(q(\mathbf{z}|\mathbf{x}, \mathbf{y}) \| p(\mathbf{z}|\mathbf{x})) \\ =&D_{\mathrm{KL}}(p_1 q(\mathbf{z}|\mathbf{x}) + p_2 q(\mathbf{z}|\mathbf{y}) \| p_1 p(\mathbf{z}|\mathbf{x}) + p_2 p(\mathbf{z}|\mathbf{x})) \\ =&\int (p_1 q(\mathbf{z}|\mathbf{x}) + p_2 q(\mathbf{z}|\mathbf{y})) \log \frac{p_1 q(\mathbf{z}|\mathbf{x}) + p_2 q(\mathbf{z}|\mathbf{y})}{p_1 p(\mathbf{z}|\mathbf{x}) + p_2 p(\mathbf{z}|\mathbf{x})} d\mathbf{z} \\ \leq&\int p_1 q(\mathbf{z}|\mathbf{x}) \log \frac{p_1 q(\mathbf{z}|\mathbf{x})}{p_1 p(\mathbf{z}|\mathbf{x})} + p_2 q(\mathbf{z}|\mathbf{y}) \log \frac{p_2 q(\mathbf{z}|\mathbf{y})}{p_2 p(\mathbf{z}|\mathbf{x})} d\mathbf{z} \\ =&p_1 D_{\mathrm{KL}}(q(\mathbf{z}|\mathbf{x}) \| p(\mathbf{z}|\mathbf{x})) + p_2 D_{\mathrm{KL}}(q(\mathbf{z}|\mathbf{y}) \| p(\mathbf{z}|\mathbf{x})), \end{aligned} \tag{9}$$

then (7) holds. Furthermore, since we can parameterize $q(\mathbf{z}|\mathbf{x})$ and $p(\mathbf{z}|\mathbf{x})$ with the same distribution, then $q(\mathbf{z}|\mathbf{x}) = p(\mathbf{z}|\mathbf{x})$, and (7) is reduced to (8). □

3. Architecture of the degradation level prediction network

We incorporate the standard deviation of the noise, along with the noisy image from the target domain, into our SeNM-VAE model to enable controlled generation of degradation levels. Specifically, we concatenate the degradation level with $\mathbf{b}_{\mathbf{n}}^{l}$ (see Equation 19 in the main paper) to enable conditional generation during both training and generation processes. Since the noisy image from the target domain lacks the corresponding clean image, its degradation level cannot be directly determined. Therefore, we introduce a degradation level prediction network trained on data from the paired domain and use it to predict the noise standard deviation for data from the target domain. The architecture of this network is illustrated in Figure 1. Our approach has been shown to successfully generate images with varying input noise levels, as demonstrated in Figure 2.

4. Experiment

4.1. Implementation details

Computation of KL divergence. We use KL divergence to evaluate the fidelity of generated noisy images. The KL divergence between two images can be calculated as follows:

$$D_{\mathrm{KL}}(\mathbf{I}_1, \mathbf{I}_2) = \sum_{i=0}^{255} p(\mathbf{I}_1 = i) \log \frac{p(\mathbf{I}_1 = i)}{p(\mathbf{I}_2 = i)}. \tag{10}$$

Training details of DnCNN. We train all DnCNN [20] models for 300k iterations using the Adam optimizer [7]. The initial learning rate is set to 10^{-4} and halved every 100k iterations. The batch size is 64, consisting of randomly cropped patches of size 40×40. Random flips and rotations are applied to augment the data. We evaluate the performance every 5k iterations on the SIDD validation dataset and select the model with the highest PSNR to evaluate on the benchmark set.

Training details of DRUNet. All DRUNet [21] models are trained for 300k iterations using the Adam optimizer [7]. The initial learning rate is set to 10^{-4} and halved every 100k iterations. The batch size is 16, consisting of randomly cropped patches of size 128×128. We augment the data by applying random flips and rotations. We evaluate the performance every 5k iterations on the SIDD validation dataset and select the model with the highest PSNR to evaluate on the benchmark set.

Training details of NAFNet. We finetune the pre-trained NAFNet [1] on synthesized training set. The model is trained for 400k iterations with Adam optimizer [7]. The initial learning rate is set to 10^{-5}, and we use the cosine learning rate decay schedule. The batch size is 2, and the patch size is 256×256. We evaluate the denoising performance every 20k iterations on the SIDD validation dataset and select the model with the highest PSNR to evaluate on the benchmark set.

Training details of ESRGAN. We use the training code from Impressionism [6] and train the ESRGAN [14] model for 60k iterations. The initial learning rate is set to 10^{-4} and halved at 5k, 10k, 20k, 30k iterations. The batch size is 16, consisting of randomly cropped patches of size 128×128. Random flips and rotations are applied to augment the data. We use the model at 60k iterations to evaluate the final performance.

Method	# Paired Data	PSNR ↑	SSIM ↑
C2N [5]	0	33.95	0.878
DeFlow [17]		33.81	0.897
LUD-VAE [22]		34.82	0.926
SeNM-VAE	0.01% (10)	36.68	0.931
	0.1% (96)	36.89	0.928
	1% (960)	**37.24**	**0.938**
DANet [18]	100%	36.20	0.925
Flow-sRGB [8]		33.24	0.876
NeCA-W [4]		36.95	0.935
SeNM-VAE		**38.27**	**0.946**
Real noise	100%	38.31	0.946

Table 1. Comparison of denoising results on SIDD benchmark. DnCNN [20] is used as a downstream denoising model.

Method	# Paired Data	PSNR ↑	SSIM ↑
C2N [5]	0	36.08	0.903
DeFlow [17]		36.71	0.923
LUD-VAE [22]		37.60	0.933
SeNM-VAE	0.01% (10)	37.94	0.936
	0.1% (96)	38.21	0.942
	1% (960)	**38.44**	**0.943**
DANet [18]	100%	38.21	0.943
Flow-sRGB [8]		36.09	0.895
NeCA-W [4]		38.70	0.946
SeNM-VAE		**39.09**	**0.950**
Real noise	100%	38.83	0.949

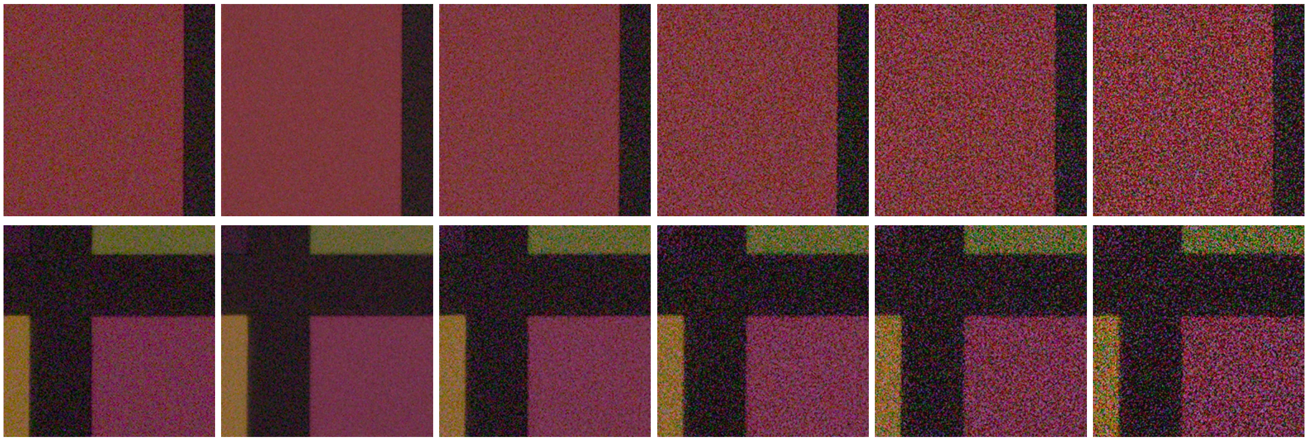
Table 2. Comparison of denoising results on DND benchmark. DnCNN [20] is used as a downstream denoising model.

4.2. Benchmark results

We replenish Table 1 in the main paper with the denoising results of DnCNN [20] on the SIDD and DND benchmarks. These results are shown in Table 1 and Table 2. Compared to the unpaired noise modeling methods, our method yields superior denoising results, even with 10 paired samples. Notably, as the number of paired samples increases, our method consistently exhibits the most effective denoising performance across both benchmarks. This further attests to the competitive advantage of our method in producing high-quality synthesized noisy images.

4.3. Model complexity

The proposed SeNM-VAE can effectively utilize a limited amount of paired data together with unpaired data to enhance the generation of high-quality training samples, without necessitating extensive computational resources. Specifically, the total number of parameters in our model amounts to 9.946M, with a total FLOPs of 617.36G required to generate a single $256 \times 256 \times 3$ image. Additionally, training can be completed within approximately 2 days on a single Nvidia 2080 Ti GPU on the SIDD dataset.



Real noisy | Gen noisy, $\sigma = 10$ | Gen noisy, $\sigma = 20$ | Gen noisy, $\sigma = 30$ | Gen noisy, $\sigma = 40$ | Gen noisy, $\sigma = 50$

Figure 2. Visual results of degradation level controllable generation on SIDD validation dataset, σ denotes the input degradation level. The model is trained with 10 paired data on the SIDD dataset.

During the inference stage, generating 1,280 images takes around 31 seconds.

4.4. Training stability

The overall training objective of SeNM-VAE consists of three parts. Firstly, it involves maximizing the conditional log-likelihood function, $\log p\left(\mathbf{y}|\mathbf{x}\right)$, through variational inference methods and the proposed mixture model, encompassing three key elements:

$$\begin{aligned} &\mathbb{E}_{q(\mathbf{z}|\mathbf{x},\mathbf{y})} D_{\mathrm{KL}}\left(q\left(\mathbf{z_n}|\mathbf{y},\mathbf{z}\right)||p\left(\mathbf{z_n}|\mathbf{z}\right)\right) \\ -&\mathbb{E}_{q(\mathbf{z}|\mathbf{x},\mathbf{y})q(\mathbf{z_n}|\mathbf{y},\mathbf{z})} \log p\left(\mathbf{y}|\mathbf{z},\mathbf{z_n}\right) \\ +&\lambda D_{\mathrm{KL}}\left(q\left(\mathbf{z}|\mathbf{y}\right)||q\left(\mathbf{z}|\mathbf{x}\right)\right). \end{aligned} \tag{11}$$

Another component comprises a regularization term, namely $\mathbb{E}_{q(\mathbf{z}|\mathbf{x},\mathbf{y})} \log p\left(\mathbf{x}|\mathbf{z}\right)$. This term plays a crucial role in enhancing the reconstruction capability of the source domain data, especially since the terms in (11) do not directly regulate the source domain data. To augment the generative capacity of the VAE model, we incorporate the LPIPS loss and GAN loss to complement the loss function for noisy image reconstruction, constituting the third part of the loss function. In our experiments, we train our model using the conventional ADAM optimizer [7] with its default settings. Employing standard training techniques in VDVAE [2], we observe stable convergence performance, as depicted in Figure 3.

4.5. Degradation modeling in real-world SR

As a complementary investigation to the noise synthesis experiment presented in the main paper, we conduct analogous assessments to evaluate the quality of the generated training pairs in real-world SR tasks. The configuration for training the degradation modeling methods remains consistent with that outlined in the downstream SR experiment. Subsequently, we train the ESRGAN [14] model using paired data derived from the comparison methods. The resultant metrics, including PSNR, SSIM, and LPIPS, on both the AIM19 and NTIRE20 datasets, are detailed in Tables 3 and Table 4, respectively. These results demonstrate the effectiveness of our semi-supervised approach in learning the degradation model in real-world SR scenarios.

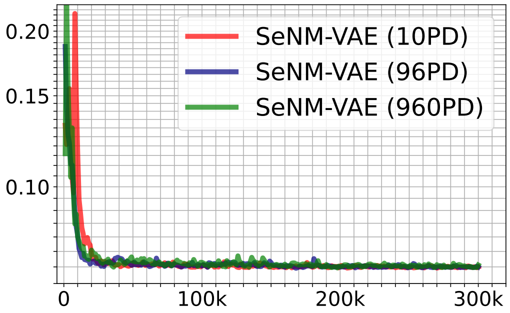


Figure 3. Loss curve of SeNM-VAE during training. Our model converges to the minimum steadily and uniformly, regardless of the quantity of paired samples utilized.

Method	PSNR ↑	SSIM ↑	LPIPS ↓
FSSR	20.97	0.5383	0.374
Impressionism	21.93	0.6128	0.426
DASR	21.05	0.5674	0.376
DeFlow	21.43	0.6003	0.349
LUD-VAE	22.25	0.6194	0.341
SeNM-VAE	**22.37**	**0.6307**	**0.335**

Table 3. Comparison of SR performance on AIM19. SeNM-VAE is trained with 10 paired data.

4.6. Effects of varying mixture weights

In our main paper, we define the inference model $q\left(\mathbf{z}|\mathbf{x},\mathbf{y}\right)$ as a linear combination of two mixture components $q\left(\mathbf{z}|\mathbf{x}\right)$

Method	PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$
FSSR	21.01	0.4229	0.435
Impressionism	25.24	0.6740	0.230
DASR	22.98	0.5093	0.379
DeFlow	24.95	0.6746	0.217
LUD-VAE	25.78	0.7196	0.220
SeNM-VAE	**25.91**	**0.7212**	**0.216**

Table 4. Comparison of SR performance on NTIRE20. SeNM-VAE is trained with 10 paired data.

and $q\left(\mathbf{z}|\mathbf{y}\right)$, expressed as:

$$q\left(\mathbf{z}|\mathbf{x}\right) = p_1 q\left(\mathbf{z}|\mathbf{x}\right) + p_2 q\left(\mathbf{z}|\mathbf{y}\right),$$

where p_1 and p_2 are mixture weights. In this experiment, we investigate the impact of different p_1 and p_2 values. Given that $p_2 = 1 - p_1$, we evaluate five cases for p_1 using the SIDD dataset, each with 10 paired samples. As shown in Table 5, the noisy data generated by SeNM-VAE achieves the minimum FID and KLD values when $p_1 = 0.5$, while the downstream denoising network (DnCNN [20]) exhibits its highest PSNR when $p_1 = 0.7$.

p_1	0.1	0.3	0.5	0.7	0.9
FID $\downarrow$	17.39	18.27	**17.25**	19.20	17.99
KLD $\downarrow$	0.037	0.044	**0.036**	0.039	0.044
PSNR $\uparrow$	36.48	36.28	36.73	**36.98**	36.72

Table 5. Comparison of noise quality on SIDD validation dataset. DnCNN [20] is used as a downstream denoising model.

5. Visual results

Owing to the space constraints within the main context, we exhibit additional visualizations of synthetic noise, real-world denoising results, and real-world SR results as a supplement.

5.1. Noise synthesis

We present synthesized noisy images generated by SeNM-VAE trained with varying numbers of paired data. Furthermore, we conduct a comparative analysis with fully-paired deep noise modeling methods, including DANet [18], Flow-sRGB [8], and NeCA-W [4]. The visual results on the SIDD validation dataset are depicted in Figure 4.

5.2. Real-world denoising

Downstream denoising. We employ DRUNet [21] as the downstream denoising model and train it on the paired domain alongside synthetic paired samples generated by SeNM-VAE. We compare our semi-supervised denoising method with direct training on the paired domain and several self-supervised denoising methods, namely CVF-SID [11], AP-BSN + R^3 [9], SCPGabNet [10], and SDAP(S)(E) [12]. Denoising results on the SIDD validation dataset are displayed in Figure 5, Figure 6, and Figure 7.

Finetune denoising. We perform fine-tuning on NAFNet [1], a pre-trained denoising model, using additional training samples generated by SeNM-VAE trained with full paired data from the SIDD training dataset. The finetuned NAFNet is compared against its original version as well as three alternative methods, namely Uformer [15], MAXIM [13], and Restormer [19]. Denoising results on the SIDD validation dataset are presented in Figure 8.

5.3. Real-world SR

SeNM-VAE is also employed to simulate the degradation process of real-world SR tasks. We leverage ESRGAN [14] as the downstream model. Our semi-supervised SR method is compared with a supervisedly trained ESRGAN, along with five unpaired degradation modeling methods, namely FSSR [3], Impressionism [6], DASR [16], DeFlow [17], and LUD-VAE [22]. Evaluation is conducted on the AIM19 and NTIRE20 validation datasets. Visualizations of the SR results are provided in Figure 9, Figure 10, Figure 11, and Figure 12.

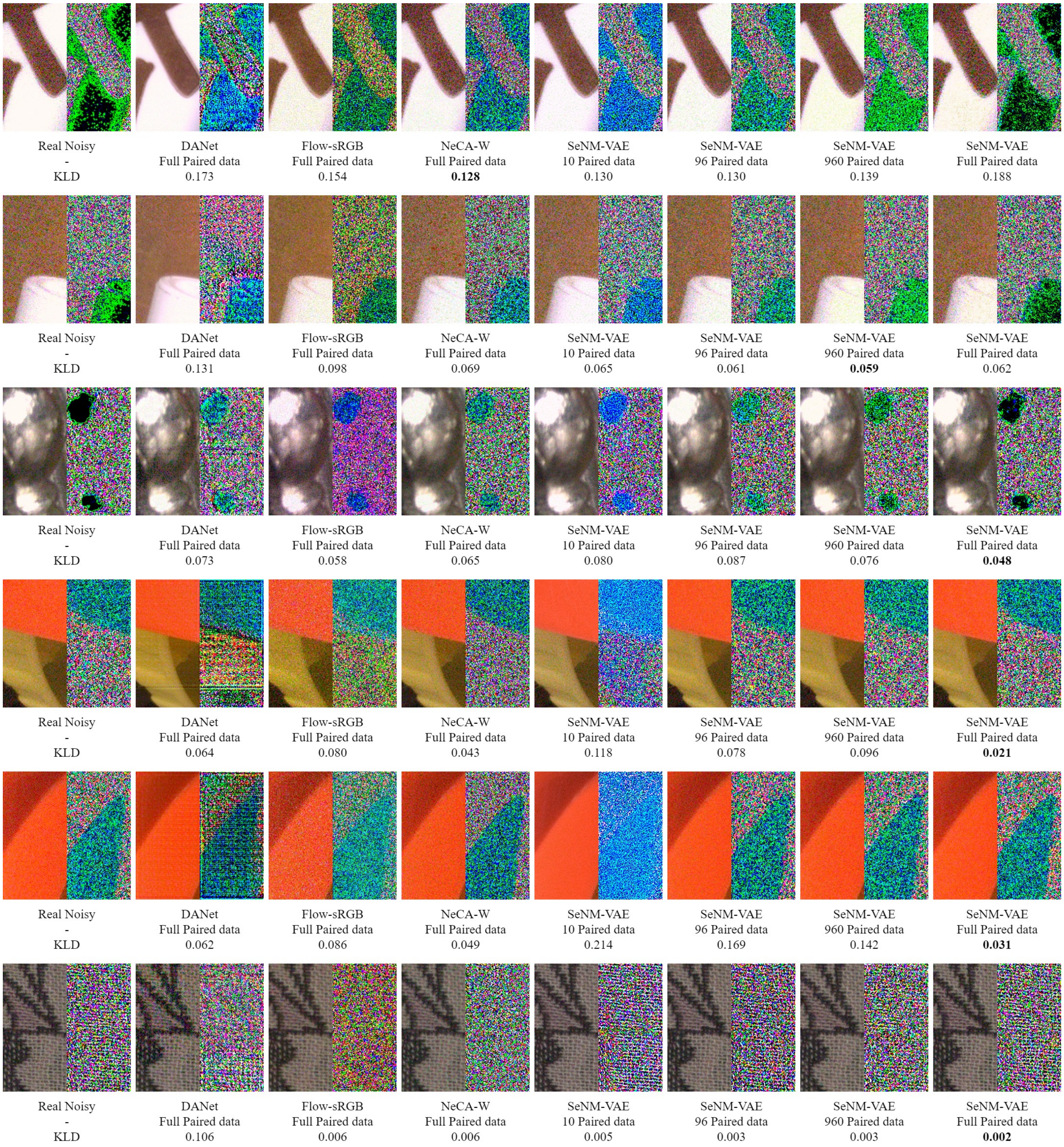


Figure 4. Visual comparisons of noise generation on the SIDD validation set. KLD value is reported as the performance metric.

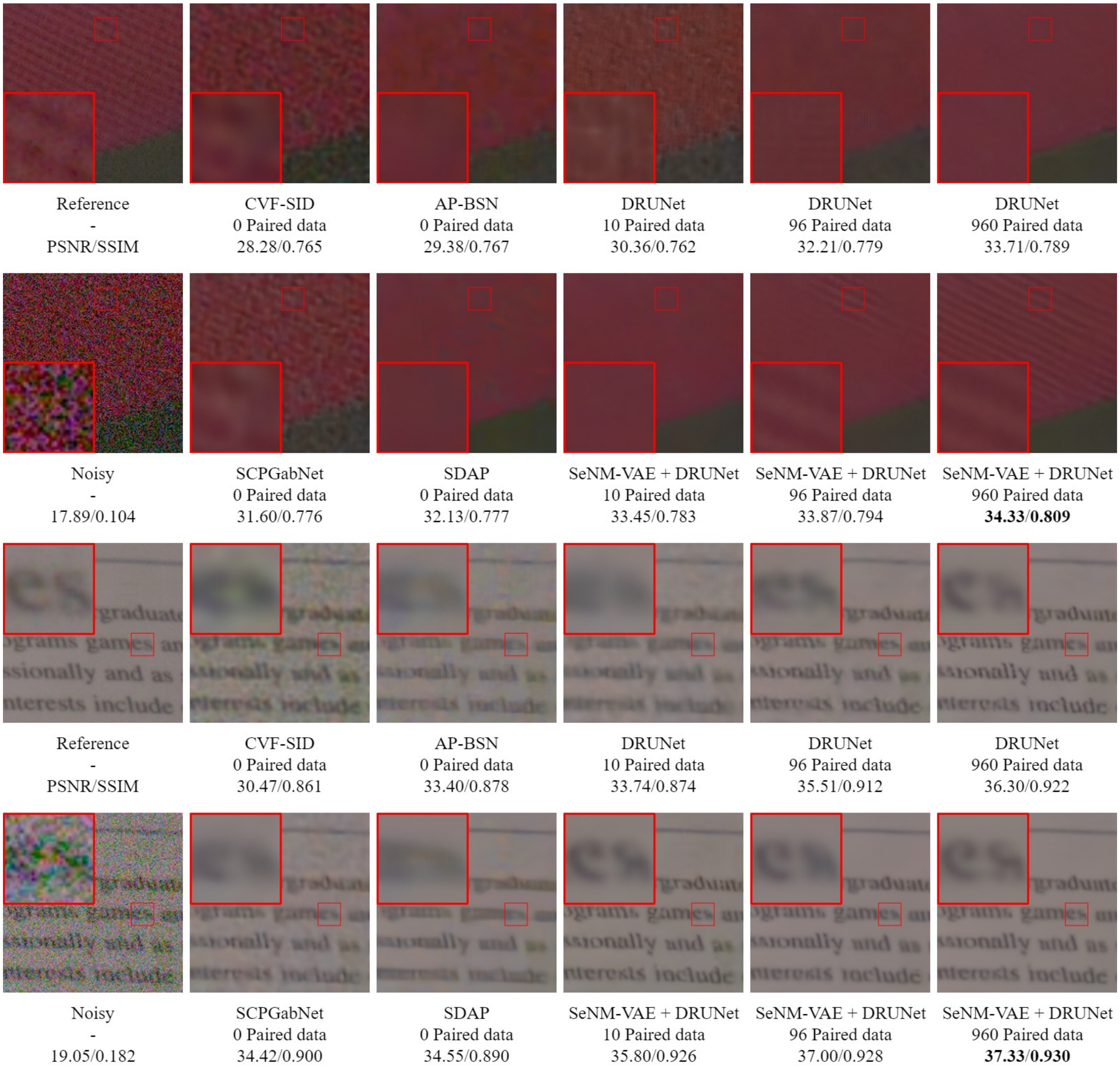
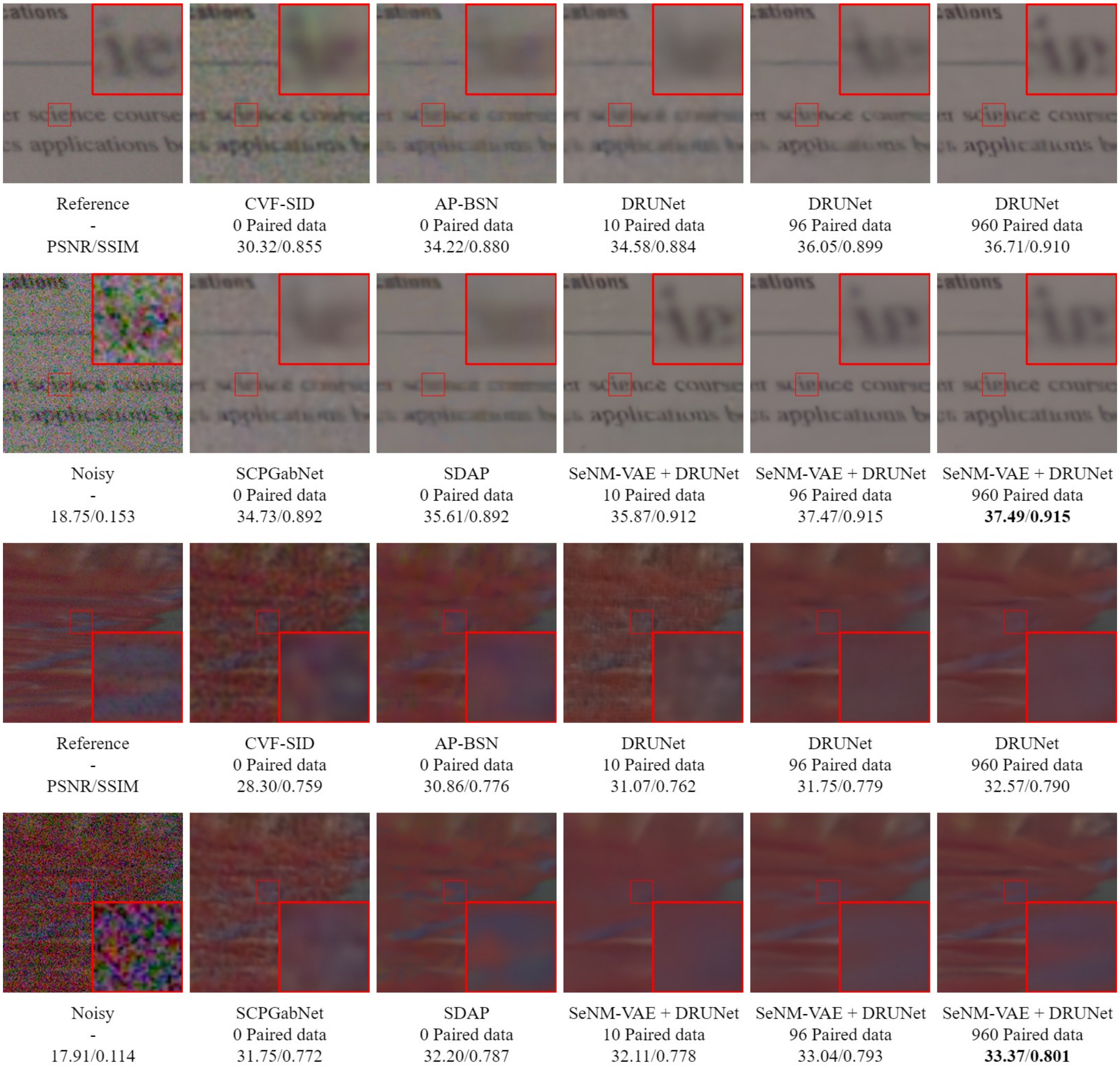


Figure 5. Visual comparisons of downstream denoising results on the SIDD validation set. Performance metrics, including PSNR and SSIM values, are reported for evaluation.



Reference
-
PSNR/SSIM

CVF-SID
0 Paired data
30.32/0.855

AP-BSN
0 Paired data
34.22/0.880

DRUNet
10 Paired data
34.58/0.884

DRUNet
96 Paired data
36.05/0.899

DRUNet
960 Paired data
36.71/0.910

Noisy
-
18.75/0.153

SCPGabNet
0 Paired data
34.73/0.892

SDAP
0 Paired data
35.61/0.892

SeNM-VAE + DRUNet
10 Paired data
35.87/0.912

SeNM-VAE + DRUNet
96 Paired data
37.47/0.915

SeNM-VAE + DRUNet
960 Paired data
37.49/**0.915**

Reference
-
PSNR/SSIM

CVF-SID
0 Paired data
28.30/0.759

AP-BSN
0 Paired data
30.86/0.776

DRUNet
10 Paired data
31.07/0.762

DRUNet
96 Paired data
31.75/0.779

DRUNet
960 Paired data
32.57/0.790

Noisy
-
17.91/0.114

SCPGabNet
0 Paired data
31.75/0.772

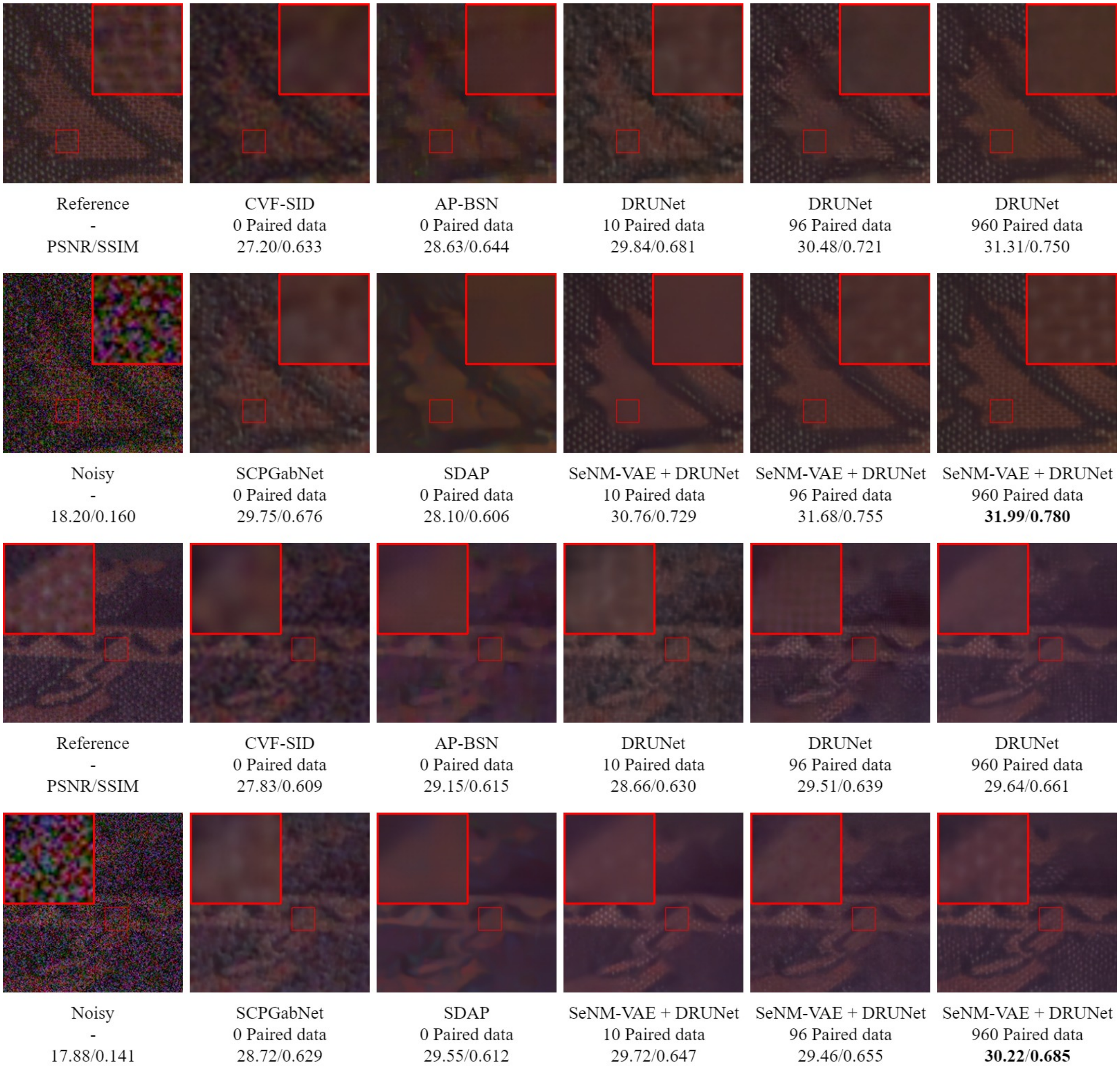
SDAP
0 Paired data
32.20/0.787

SeNM-VAE + DRUNet
10 Paired data
32.11/0.778

SeNM-VAE + DRUNet
96 Paired data
33.04/0.793

SeNM-VAE + DRUNet
960 Paired data
33.37/**0.801**

Figure 6. Visual comparisons of downstream denoising results on the SIDD validation set. Performance metrics, including PSNR and SSIM values, are reported for evaluation.



Reference
-
PSNR/SSIM

CVF-SID
0 Paired data
27.20/0.633

AP-BSN
0 Paired data
28.63/0.644

DRUNet
10 Paired data
29.84/0.681

DRUNet
96 Paired data
30.48/0.721

DRUNet
960 Paired data
31.31/0.750

Noisy
-
18.20/0.160

SCPGabNet
0 Paired data
29.75/0.676

SDAP
0 Paired data
28.10/0.606

SeNM-VAE + DRUNet
10 Paired data
30.76/0.729

SeNM-VAE + DRUNet
96 Paired data
31.68/0.755

SeNM-VAE + DRUNet
960 Paired data
31.99/**0.780**

Reference
-
PSNR/SSIM

CVF-SID
0 Paired data
27.83/0.609

AP-BSN
0 Paired data
29.15/0.615

DRUNet
10 Paired data
28.66/0.630

DRUNet
96 Paired data
29.51/0.639

DRUNet
960 Paired data
29.64/0.661

Noisy
-
17.88/0.141

SCPGabNet
0 Paired data
28.72/0.629

SDAP
0 Paired data
29.55/0.612

SeNM-VAE + DRUNet
10 Paired data
29.72/0.647

SeNM-VAE + DRUNet
96 Paired data
29.46/0.655

SeNM-VAE + DRUNet
960 Paired data
30.22/**0.685**

Figure 7. Visual comparisons of downstream denoising results on the SIDD validation set. Performance metrics, including PSNR and SSIM values, are reported for evaluation.

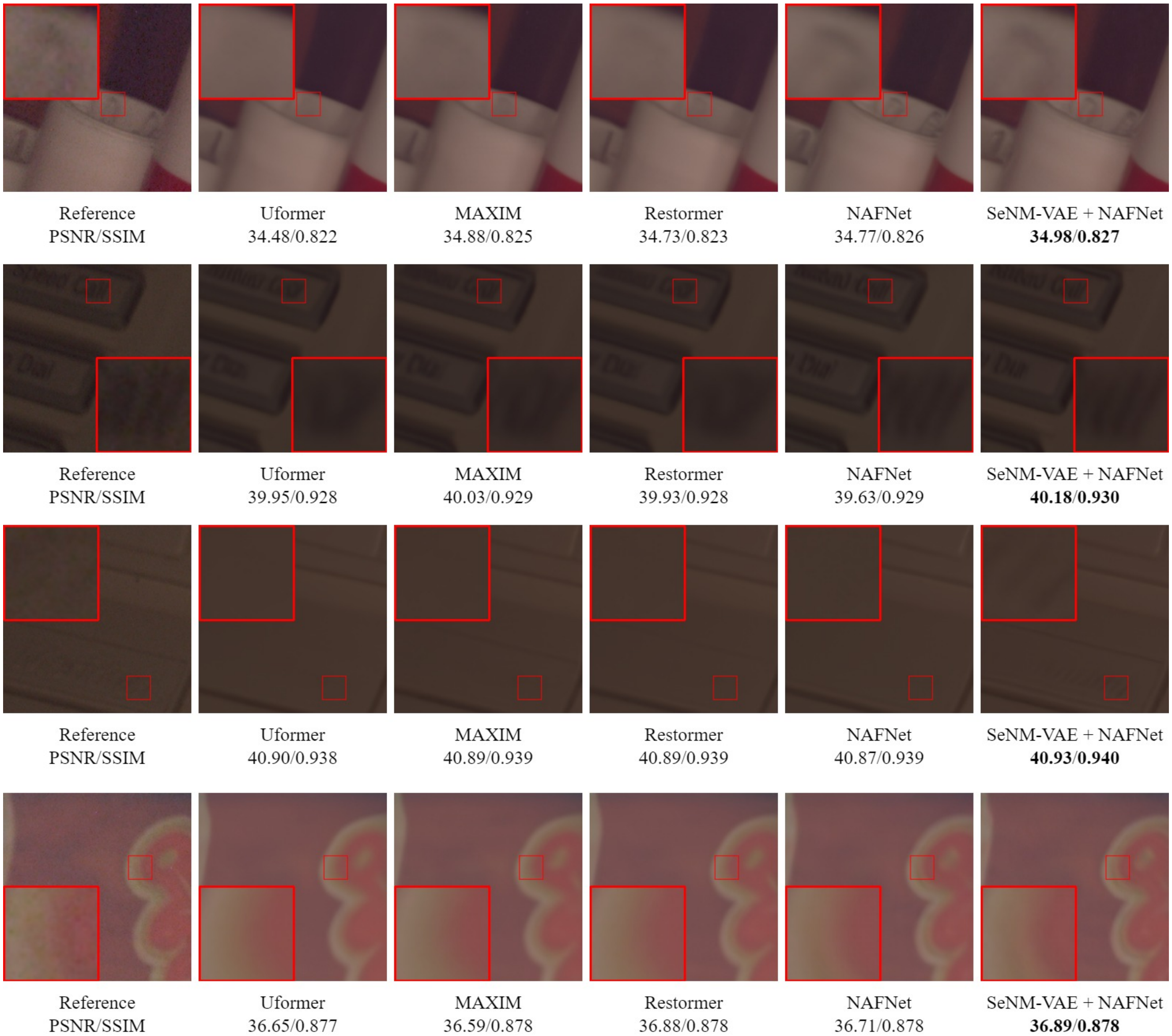
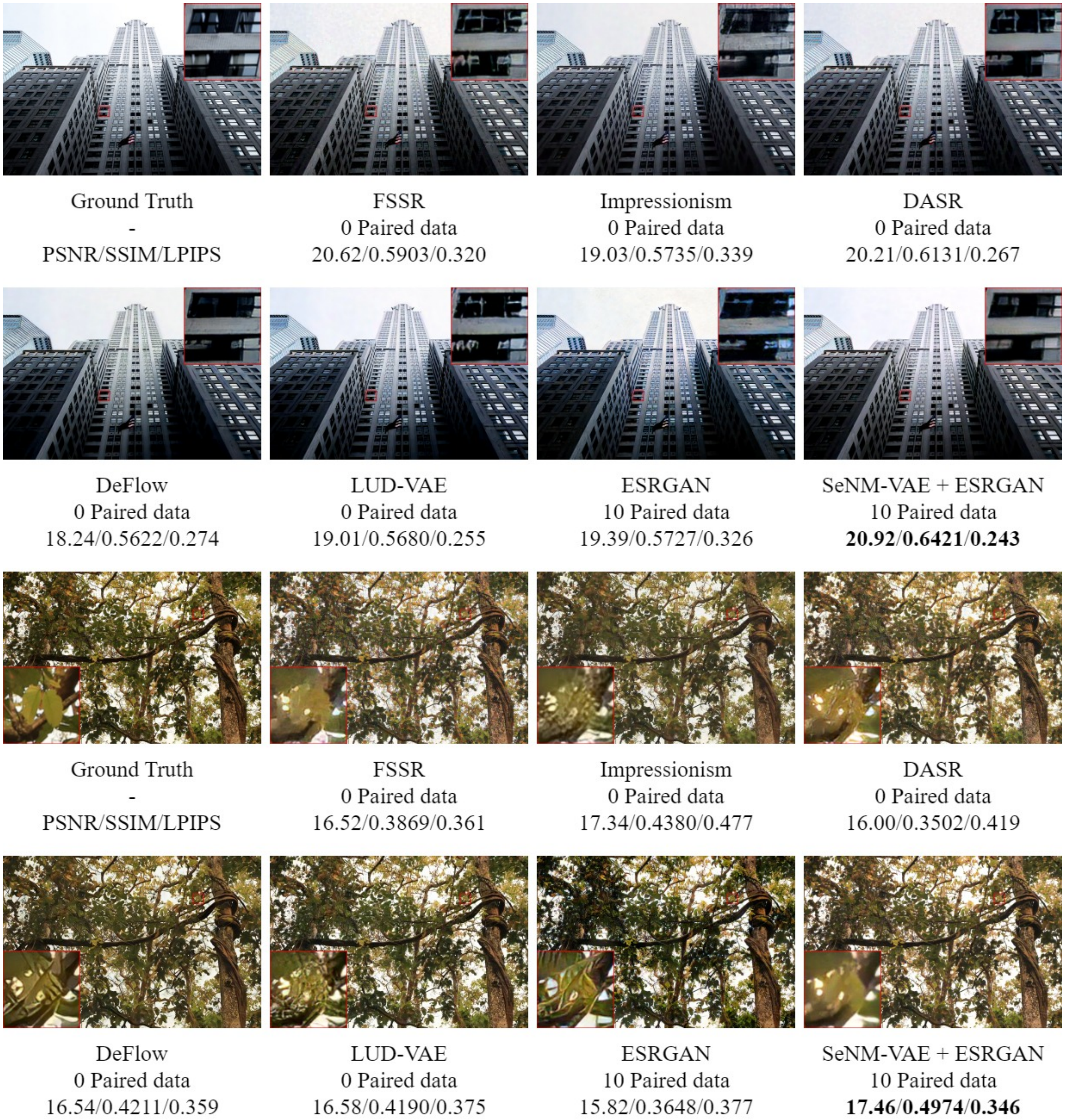


Figure 8. Visual comparisons of fine-tuned denoising results on the SIDD validation set. Performance metrics, including PSNR and SSIM values, are reported for evaluation.



Ground Truth
-
PSNR/SSIM/LPIPS

FSSR
0 Paired data
20.62/0.5903/0.320

Impressionism
0 Paired data
19.03/0.5735/0.339

DASR
0 Paired data
20.21/0.6131/0.267

DeFlow
0 Paired data
18.24/0.5622/0.274

LUD-VAE
0 Paired data
19.01/0.5680/0.255

ESRGAN
10 Paired data
19.39/0.5727/0.326

SeNM-VAE + ESRGAN
10 Paired data
20.92/**0.6421**/**0.243**

Ground Truth
-
PSNR/SSIM/LPIPS

FSSR
0 Paired data
16.52/0.3869/0.361

Impressionism
0 Paired data
17.34/0.4380/0.477

DASR
0 Paired data
16.00/0.3502/0.419

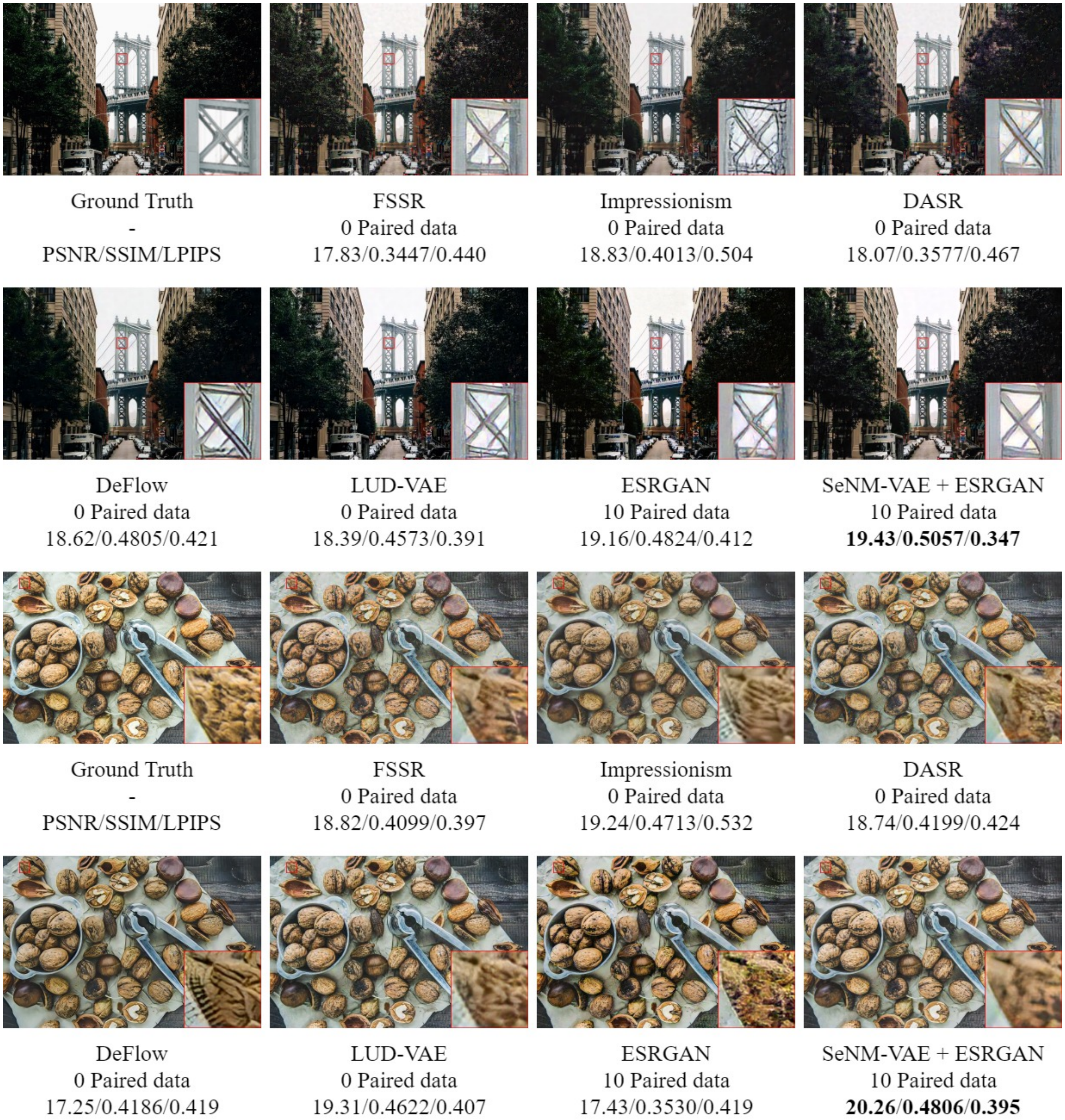
DeFlow
0 Paired data
16.54/0.4211/0.359

LUD-VAE
0 Paired data
16.58/0.4190/0.375

ESRGAN
10 Paired data
15.82/0.3648/0.377

SeNM-VAE + ESRGAN
10 Paired data
17.46/**0.4974**/**0.346**

Figure 9. Visual comparisons of real-world SR results on the AIM19 validation set. Performance metrics, including PSNR, SSIM, and LPIPS values, are provided for evaluation.



Ground Truth
-
PSNR/SSIM/LPIPS

FSSR
0 Paired data
17.83/0.3447/0.440

Impressionism
0 Paired data
18.83/0.4013/0.504

DASR
0 Paired data
18.07/0.3577/0.467

DeFlow
0 Paired data
18.62/0.4805/0.421

LUD-VAE
0 Paired data
18.39/0.4573/0.391

ESRGAN
10 Paired data
19.16/0.4824/0.412

SeNM-VAE + ESRGAN
10 Paired data
19.43/**0.5057**/**0.347**

Ground Truth
-
PSNR/SSIM/LPIPS

FSSR
0 Paired data
18.82/0.4099/0.397

Impressionism
0 Paired data
19.24/0.4713/0.532

DASR
0 Paired data
18.74/0.4199/0.424

DeFlow
0 Paired data
17.25/0.4186/0.419

LUD-VAE
0 Paired data
19.31/0.4622/0.407

ESRGAN
10 Paired data
17.43/0.3530/0.419

SeNM-VAE + ESRGAN
10 Paired data
20.26/**0.4806**/**0.395**

Figure 10. Visual comparisons of real-world SR results on the AIM19 validation set. Performance metrics, including PSNR, SSIM, and LPIPS values, are provided for evaluation.

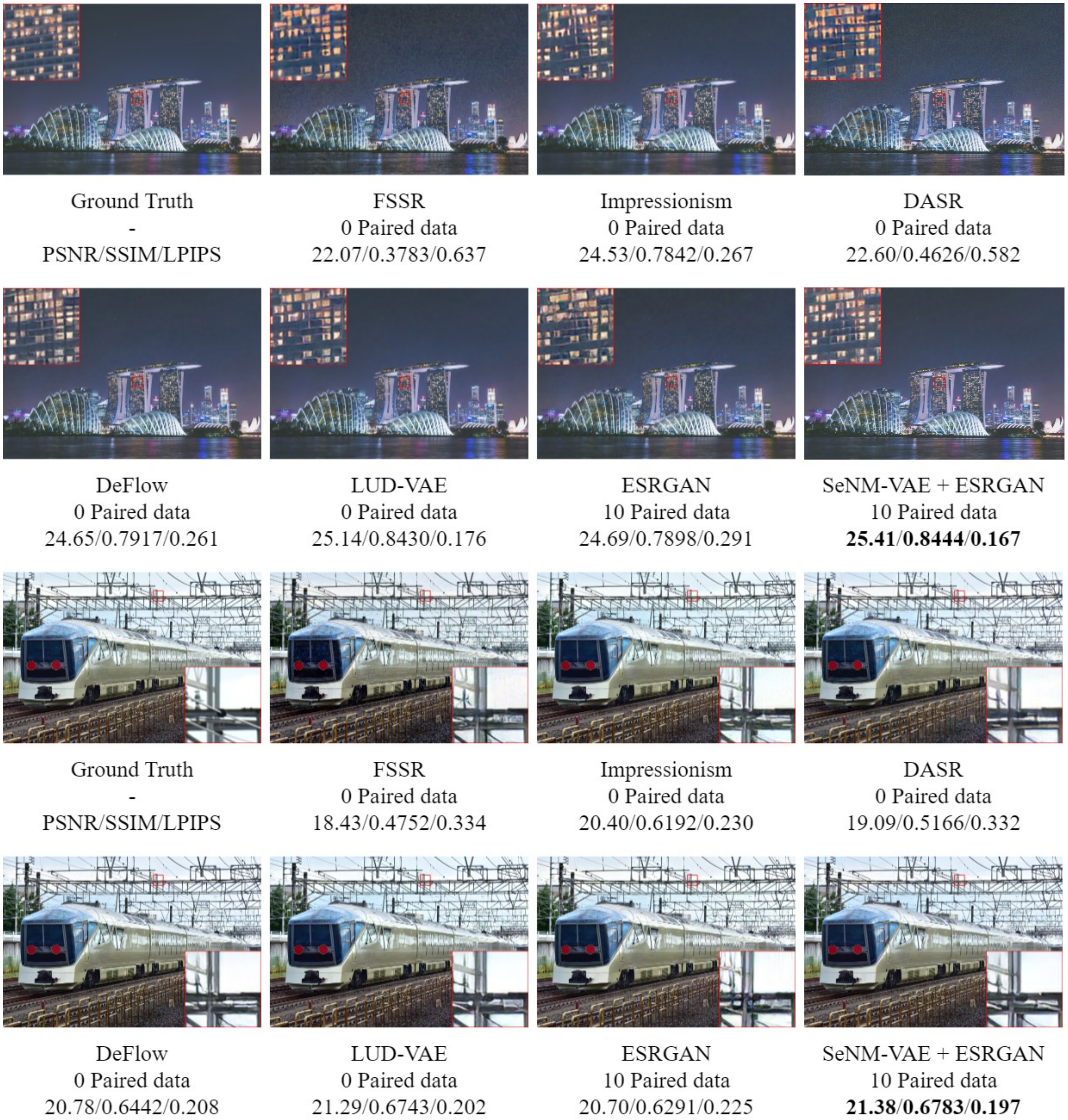
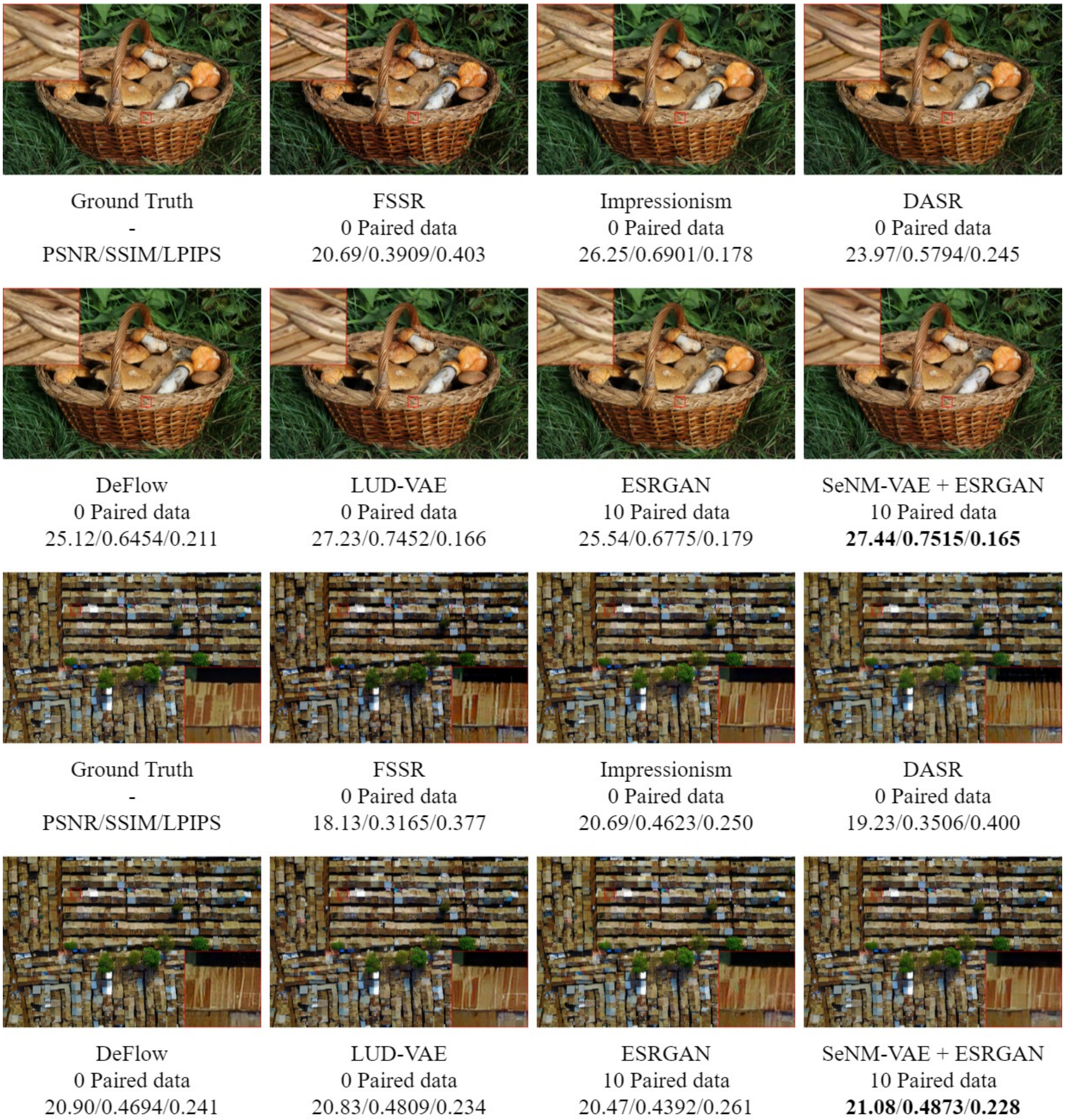


Figure 11. Visual comparisons of real-world SR results on the NTIRE20 validation set. Performance metrics, including PSNR, SSIM, and LPIPS values, are provided for evaluation.



Ground Truth
-
PSNR/SSIM/LPIPS

FSSR
0 Paired data
20.69/0.3909/0.403

Impressionism
0 Paired data
26.25/0.6901/0.178

DASR
0 Paired data
23.97/0.5794/0.245

DeFlow
0 Paired data
25.12/0.6454/0.211

LUD-VAE
0 Paired data
27.23/0.7452/0.166

ESRGAN
10 Paired data
25.54/0.6775/0.179

SeNM-VAE + ESRGAN
10 Paired data
27.44/**0.7515**/**0.165**

Ground Truth
-
PSNR/SSIM/LPIPS

FSSR
0 Paired data
18.13/0.3165/0.377

Impressionism
0 Paired data
20.69/0.4623/0.250

DASR
0 Paired data
19.23/0.3506/0.400

DeFlow
0 Paired data
20.90/0.4694/0.241

LUD-VAE
0 Paired data
20.83/0.4809/0.234

ESRGAN
10 Paired data
20.47/0.4392/0.261

SeNM-VAE + ESRGAN
10 Paired data
21.08/**0.4873**/**0.228**

Figure 12. Visual comparisons of real-world SR results on the NTIRE20 validation set. Performance metrics, including PSNR, SSIM, and LPIPS values, are provided for evaluation.

References

[1] Liangyu Chen, Xiaojie Chu, Xiangyu Zhang, and Jian Sun. Simple baselines for image restoration. In *ECCV*, pages 17–33. Springer, 2022. 2, 4

[2] Rewon Child. Very deep vaes generalize autoregressive models and can outperform them on images. *arXiv:2011.10650*, 2020. 3

[3] Manuel Fritsche, Shuhang Gu, and Radu Timofte. Frequency separation for real-world super-resolution. In *ICCVW*, pages 3599–3608. IEEE, 2019. 4

[4] Zixuan Fu, Lanqing Guo, and Bihan Wen. srgb real noise synthesizing with neighboring correlation-aware noise model. In *CVPR*, pages 1683–1691, 2023. 2, 4

[5] Geonwoon Jang, Wooseok Lee, Sanghyun Son, and Kyoung Mu Lee. C2n: Practical generative noise modeling for real-world denoising. In *ICCV*, pages 2350–2359, 2021. 2

[6] Xiaozhong Ji, Yun Cao, Ying Tai, Chengjie Wang, Jilin Li, and Feiyue Huang. Real-world super-resolution via kernel estimation and noise injection. In *CVPRW*, pages 466–467, 2020. 2, 4

[7] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv:1412.6980*, 2014. 2, 3

[8] Shayan Kousha, Ali Maleky, Michael S Brown, and Marcus A Brubaker. Modeling srgb camera noise with normalizing flows. In *CVPR*, pages 17463–17471, 2022. 2, 4

[9] Wooseok Lee, Sanghyun Son, and Kyoung Mu Lee. Ap-bsn: Self-supervised denoising for real-world images via asymmetric pd and blind-spot network. In *CVPR*, pages 17725–17734, 2022. 4

[10] Xin Lin, Chao Ren, Xiao Liu, Jie Huang, and Yinjie Lei. Unsupervised image denoising in real-world scenarios via self-collaboration parallel generative adversarial branches. In *ICCV*, pages 12642–12652, 2023. 4

[11] Reyhaneh Neshatavar, Mohsen Yavartanoo, Sanghyun Son, and Kyoung Mu Lee. Cvf-sid: Cyclic multi-variate function for self-supervised image denoising by disentangling noise from image. In *CVPR*, pages 17583–17591, 2022. 4

[12] Yizhong Pan, Xiao Liu, Xiangyu Liao, Yuanzhouhan Cao, and Chao Ren. Random sub-samples generation for self-supervised real image denoising. In *ICCV*, pages 12150–12159, 2023. 4

[13] Zhengzhong Tu, Hossein Talebi, Han Zhang, Feng Yang, Peyman Milanfar, Alan Bovik, and Yinxiao Li. Maxim: Multi-axis mlp for image processing. In *CVPR*, pages 5769–5780, 2022. 4

[14] Xintao Wang, Ke Yu, Shixiang Wu, Jinjin Gu, Yihao Liu, Chao Dong, Yu Qiao, and Chen Change Loy. Esrgan: Enhanced super-resolution generative adversarial networks. In *ECCVW*, pages 0–0, 2018. 2, 3, 4

[15] Zhendong Wang, Xiaodong Cun, Jianmin Bao, Wengang Zhou, Jianzhuang Liu, and Houqiang Li. Uformer: A general u-shaped transformer for image restoration. In *CVPR*, pages 17683–17693, 2022. 4

[16] Yunxuan Wei, Shuhang Gu, Yawei Li, Radu Timofte, Longcun Jin, and Hengjie Song. Unsupervised real-world image super resolution via domain-distance aware training. In *CVPR*, pages 13385–13394, 2021. 4

[17] Valentin Wolf, Andreas Lugmayr, Martin Danelljan, Luc Van Gool, and Radu Timofte. Deflow: Learning complex image degradations from unpaired data with conditional flows. In *CVPR*, pages 94–103, 2021. 2, 4

[18] Zongsheng Yue, Qian Zhao, Lei Zhang, and Deyu Meng. Dual adversarial network: Toward real-world noise removal and noise generation. In *ECCV*, pages 41–58. Springer, 2020. 2, 4

[19] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Ming-Hsuan Yang. Restormer: Efficient transformer for high-resolution image restoration. In *CVPR*, pages 5728–5739, 2022. 4

[20] Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE Trans Image Process*, 26(7):3142–3155, 2017. 2, 4

[21] Kai Zhang, Yawei Li, Wangmeng Zuo, Lei Zhang, Luc Van Gool, and Radu Timofte. Plug-and-play image restoration with deep denoiser prior. *IEEE Trans. Pattern Anal. Mach. Intell.*, 44(10):6360–6376, 2021. 2, 4

[22] Dihan Zheng, Xiaowen Zhang, Kaisheng Ma, and Chenglong Bao. Learn from unpaired data for image restoration: A variational bayes approach. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2022. 2, 4