Supplemental Materials: Asymmetric Mask Scheme for Self-Supervised Real Image Denoising

Xiangyu Liao[1], Tianheng Zheng[1], Jiayu Zhong[1], Pingping Zhang[2], and Chao Ren[1]*

[1] Sichuan University
[2] Dalian University of Technology
{liaoxiangyu1,zhengtianheng,jiayuzhong}@stu.scu.edu.cn
zhpp@dlut.edu.cn,chaoren@scu.edu.cn

1 Overview

The content of the supplementary material includes:

- Analysis of why the mask strategy can achieve self-supervised denoising are provided in Sec. 1.1.
- Details about constructing mask index matrix set with different number k of denoising branches, and strategies of training and inference when the mask ratio of each denoising branches is approximate to 75% are provided in Sec. 1.2.
- Details about training, fine-tuning, and inference of our denoising models are provided in Sec. 1.3.
- Visualization of different strategies to eliminate the checkerboard effect is provided in Sec. 1.4.
- Analysis of the Eq.5 in Sec. 1.5.
- Analysis of the computational complexity of our method is provided in Sec. 1.6.
- Generalization of our approach in Sec. 1.7.
- Details about the denoiser in AP-BSN, and visual results of AP-BSN and AMSNet when the receptive filed of denoiser is not limited are provided in Sec. 1.8.
- The limitations of our method are discussed in Sec. 1.9.
- More visual denoising results are provided in Sec. 1.10.

1.1 Remove Noise by Mask Strategies

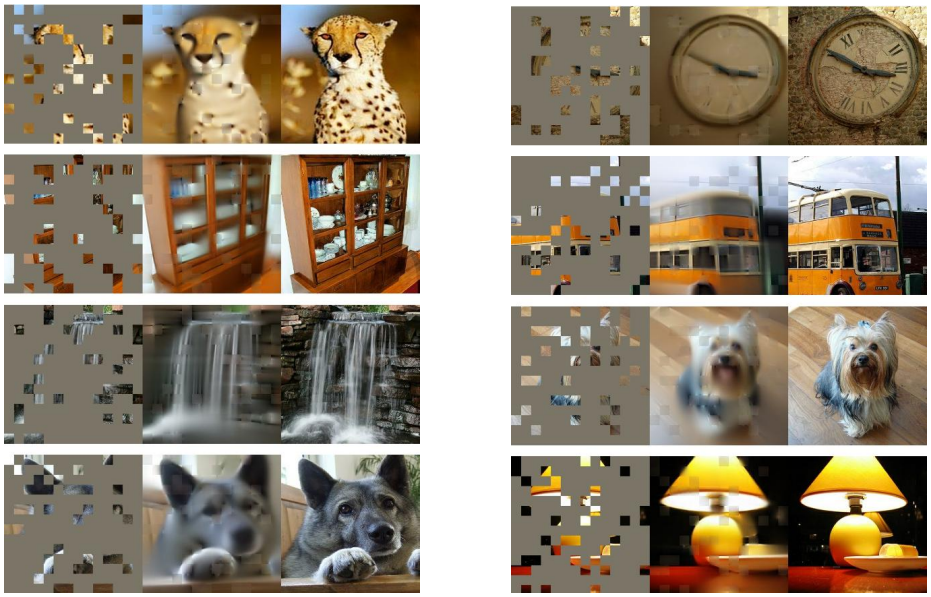
Mask Autoencoders (MAE) [2] discover that if parts of an image are masked, the remaining areas can be used to reconstruct the masked parts, as shown in Fig. 1a and Fig. 1b, which demonstrates the interrelation between the masked parts and such unmasked parts can be approximated using deep models. Now,

* Corresponding author.

let's consider a noise-free original image $I \in R^{c\times h\times w}$, where c is the channels of the image and h, w represents the height and width of the image. If we add a noise signal $N \in R^{c\times h\times w}$ where each signal is independent and has a zero mean, we obtain a synthetic noisy image denoted as $I_N = I + N$. For our single mask training scheme, if we mask some areas of I_N with a mask index matrix M, we can guide our denoiser D_E to reconstruct the content in the masked area of I_N by optimizing the proposed mask loss:

$$\arg\min_{\theta} \parallel \tilde{M} \odot (D_E(I_N \odot M, \theta) - I_N) \parallel_1 \tag{1}$$

where $\tilde{M}$ is the complement of M, and θ is the parameter of D_E. Since the noise N is mutually independent, it is impossible to reconstruct the masked noise $M \odot N$ from the unmasked noise $\tilde{M} \odot N$, leading to the clean estimation. For more details, please refer to [3]. Then, for real noise that does not meet the assumption of being uncorrelated, we use pixel downsampling (PD) to disrupt the correlation between noise signals, making them independent [4,13]. Therefore, the real noise can be well removed by the previous mask strategy.



(a) (b)

Fig. 1: The left column shows the parts where a large area has been masked, the middle column is the image restored by the neural network from the left column, and the right column is the original image that has not been masked. Through the neural network, the masked parts are restored using the unmasked areas.

1.2 Mask Matrix Construction

As the Fig. 2 illustrated, it shows how to construct our mask matrix. To ensure that all mask index matrix can ultimately cover all the noisy pixels, we distribute all pixels as evenly and randomly as possible into different index matrix. This ensures that each of the final k denoising branches is responsible for processing a proportion of noisy pixels that is around $\frac{1}{k}$.

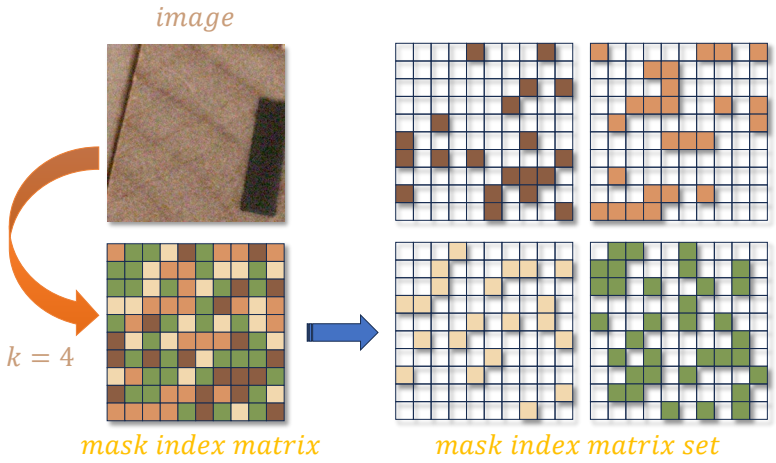


Fig. 2: Construction of mask index matrix set for a single image with $k = 4$.

When the mask ratio of each denoising branch is set to be larger than 50%, we propose a different strategy. Take the mask ratio of 75% as an expmale. We implement an excessive masking strategy to balance the masking areas across 4 denoising branches. To satisfy the assumption of noise independent, we apply a PD operation P_s on noisy image I_N to obtain noise-independent sub-samples I_s. Then, four mask index matrix are constructed for each sample, like Fig. 2. These mask matrices are organized into four mask matrix sets according to the corresponding sub-sample and denoising branches, denoted as $M_s^1, \ldots, M_s^4$, and $\sum_{i=1}^{4} M_s^i = 3$. Their complementary matrix sets, $\tilde{M}_s^1, \ldots, \tilde{M}_s^4$ and $\sum_{i=1}^{4} \tilde{M}_s^i = 1$. Then , we aggregate the outputs of those denoising branches and compute the average frequency of pixel masking to derive the final denoising result. The formula for the final denoised image is as follows:

$$I_{DN} = P_s^{-1}\left(\frac{\sum_{i=1}^{4} D_i(\tilde{M}_s^i, I_s, \theta)}{\sum_{i=1}^{4} M_s^i}\right) \tag{2}$$

1.3 Details of Training and Inference

During the training phase, 160×160 pixel image blocks are extracted from the SIDD MEDIUM [1] dataset for input. The optimization process is driven by the

AdamW [10] optimizer with a learning rate set to $1e^{-4}$ and betas configured as [0.9, 0.999]. To dynamically control the learning rate throughout training, the Cosine Annealing Warm Restarts scheduler is employed. The loss function used in this stage is $\mathcal{L}_m$. After training, the obtained model is denoted as AMSNet-B, and we transition to a fine-tuning phase. Here, the cropped image blocks are 320×320, and the base learning rate is adjusted to $1e^{-5}$. Fine-tuning is performed using the loss function $\mathcal{L}_t$ and the model is denoted as AMSNet-P. This two-phase training approach enables the model to first learn essential features and then refine its performance for improved denoising results. During training, we use P_5 operation and we set $k = 2$.

The strategy during inference is different from that during training. In the inference phase, we follow [4], and use the P_2 operation. For the SIDD and DND benchmarks, we obtain metrics by submitting our results to their respective online evaluation servers. For the SIDD vlidation and the PolyU, the metrics are computed locally.

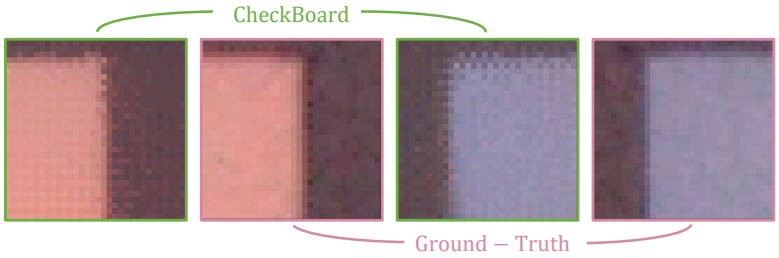


Fig. 3: Checkerboard effect and the ground-truth. The ground-truth appears smoother, while the checkerboard seems staggered.

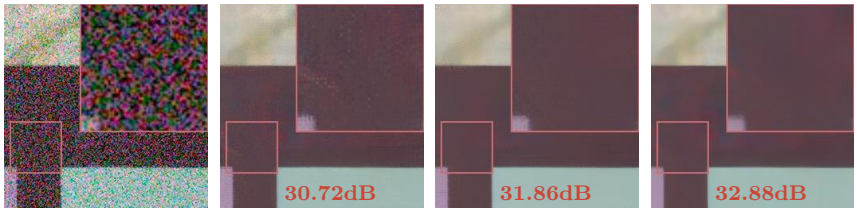
1.4 Checkerboard Effect and Solutions

The denoising results obtained based on the baseline **AMSNet-B** have a strong checkerboard effect like Fig. 3. By introducing a priori smoothing loss for fine-tuning, the resulting **AMSNet-P** greatly weakens the checkerboard effect like Fig. 4c. However, the checkerboard effect still remains. The random refinement enhancement strategy [4] is used on the basis of **AMSNet-P** to basically eliminate the checkerboard effect, which is recorded as **AMSNet-P-E** like Fig. 4d.

The introduction of prior smoothing loss leads to a certain degree of over-smoothing, which is a common issue in self-supervised denoising tasks [9]. We will strive to improve this issue in our subsequent work.

1.5 Details of Eq.5

In Eq.5, each binary matrix $\tilde{M}_s^i$ has $P\%$ of its elements as 1, and each binary matrix M_s^i has $(100 - P)\%$ of its elements as 1, where $P = 100/k$ and $\tilde{M}_s^i = \mathbb{I} - M_s^i$. Since the positions to one elements in each matrix $\tilde{M}_s^i$ do not overlap



(a) *Noisy* (b) **AMSNet-B** (c) **AMSNet-P** (d) **AMSNet-P-E**

Fig. 4: Visualization of different strategies to eliminate the checkerboard effect.

with each other, the sum of all matrices satisfies $\sum_{i=1}^{k} \tilde{M}_s^i = \mathbb{I}$, which cover all pixels. Additionally, because $\sum_{i=1}^{k}(\tilde{M}_s^i + M_s^i) = k\mathbb{I}$, so $\sum_{i=1}^{k} M_s^i = (k-1)\mathbb{I}$.

1.6 Computational Complexity

Our AMSNet applies an asymmetric strategy during training and inference, making it plug-and-play for various denoising methods. Depending on the specific requirements, different complexities of denoisers can be selected. To address various computational demands, we selected several representative denoisers to meet different requirements for real-time performance and restoration capabilities.

Tab. 1 displays the theoretical FLOPS and parameter counts of two typical BSN methods and our method (with five denoisers) when the input image size is 160×160 pixels. Compared with representative BSN methods, AMSNet incurs a similar denoiser's FLOPs(G) cost but achieves significantly improved performance about 1.2dB to APBSN when using Restormer.

Fig. 5 presents the corresponding denoising results on the SIDD Validation set and the average processing time per image (256×256 pixels). All tests were conducted on the same NVIDIA RTX 3090 GPU. AP-BSN (36.74dB) needs 0.26 seconds per image, while AMSNet (DNCNN as denoiser, 36.93dB) only needs 0.038 seconds, AMSNet (Restormer as denoiser, 37.93dB) needs 1.802 seconds with SOTA performance. LG-BPN even takes about 8 seconds per image by its official code.

Table 1: Computational complexity of different denoisers.

Scheme/Denoiser	Parameters (M)	FLOPs (G)
AP-BSN [4]	3.656	100.642
LG-BPN [11]	3.656	100.642
AMSNet(Restormer)	26.112	110.345
AMSNet(DeamNet)	2.228	114.21
AMSNet(NAFNet)	29.056	12.810
AMSNet(Unet)	7.936	69.873
AMSNet(DnCNN)	5.898	0.115

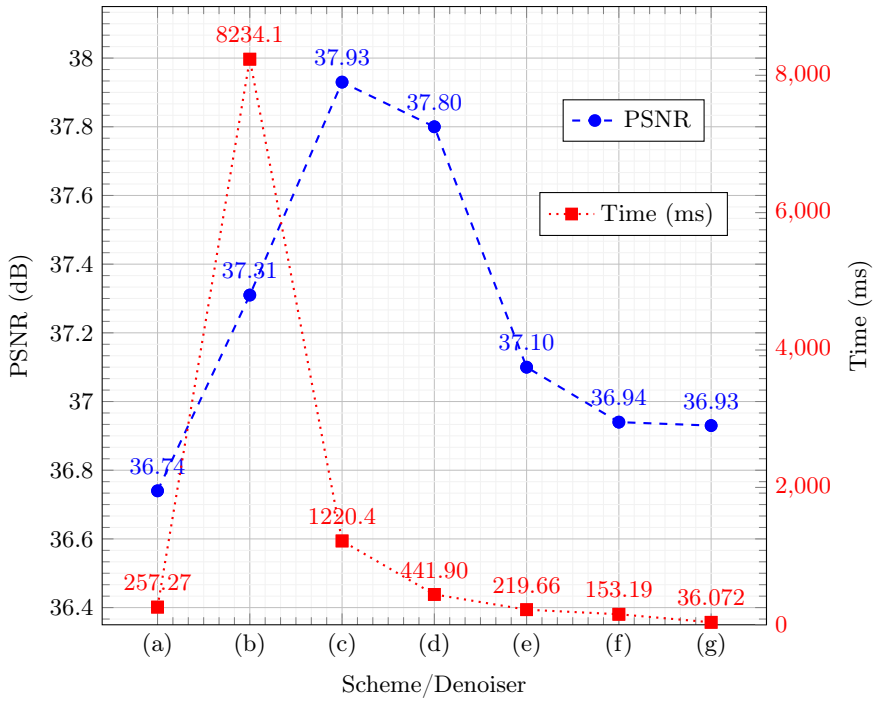


Fig. 5: PSNR on SIDD Validation and the runtime cost. the (a) to (g) represents the (a) AP-BSN, (b) LG-BPN, (c) AMSNet(Restormer), (d) AMSNet(DeamNet), (e) AMSNet(NAFNet), (f) AMSNet(Unet), (g) AMSNet(DnCNNN), respectively.

1.7 Generalization

Our approach AMSNet exhibits good generalization, only requires training on noisy dataset (e.g., SIDD-MEDIUM) but achieves SOTA performance on other datasets (e.g., DND, PolyU) and other self-captured real noisy images.

We ues the model AMSNet-B (Restormer is used as denoiser and it is trained on the widely used SIDD Medium) with P_1 during inference phase and achieve good denoising results on widely used dataset CBSD68 [7] with AWGN ($\sigma \in [10, 25, 50]$), as shown in the Tab. 2. Without targeted training, our method still demonstrates good denoising performance as the Fig. 6.

Table 2: Performance of AMSNet-B on CBSD68 with AWGN.

σ	10	25	50
PSNR (dB)/ SSIM	29.89/0.857	27.57/0.812	25.82/0.717

(a) noisy image from CBSD68 with AWGN ($\sigma = 10$). **(b)** denoising result. **(c)** ground-truth.

Fig. 6: Denoising results on CBSD68_0042 with AWGN ($\sigma = 10$). the PSNR is 35.39 dB.

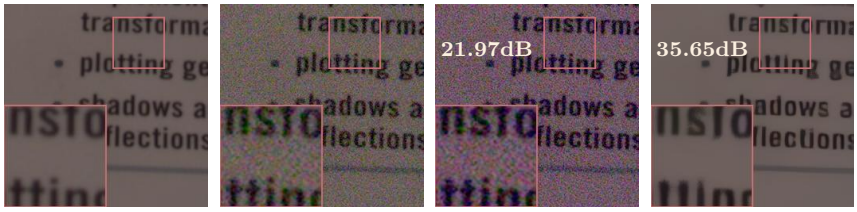
1.8 Identity Mapping Removal

In our experiments, the denoiser D_A of AP-BSN is used, and its specific structure is shown in the Fig. 10. In our ablation experiments, we replace the dilated convolution with an equivalent regular convolution in all MDC module and remove the restriction on the receptive field to verify the versatility of our method in removing identity mapping.

Fig. 7, Fig. 8, and Fig. 9 shows the performances of AMSNet and AP-BSN when using a denoiser with an unrestricted receptive field. In the AP-BSN framework, when the receptive field of the denoiser is unrestricted, an identity mapping

(a) *GT* **(b)** *Noisy* **(c)** *AP-BSN* [4] **(d)** *AMSNet*

Fig. 7: Denoising effects under different frameworks. When the receptive field of the denoiser is unconstrained, AMSNet successfully removes noise, but AP-BSN suffers from identity mapping from noise to noise.



(a) *GT* **(b)** *Noisy* **(c)** *AP-BSN* [4] **(d)** *AMSNet*

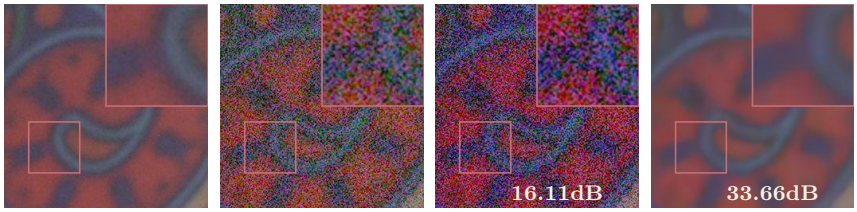
Fig. 8: Denoising effects under different frameworks. When the receptive field of the denoiser is unconstrained, AMSNet successfully removes noise, but AP-BSN suffers from identity mapping from noise to noise.

from noise to noise occurs, whereas our framework maintains good denoising effects.

1.9 Limitations

Although our method has achieved state-of-the-art results in the field of self-supervised denoising, there are still some limitations as follows:

- During the inference phase, we use all branches (k branches and $k \geq 2$) directly to generate the final denoised image, the denoiser D_E is called k times. Although this brings considerable performance improvement, achieving complete denoising of the entire image, it results in a manifold increase in computational cost. We will continue to address this issue in subsequent work.
- Due to the introduction of the the widely used random refinement enhancement strategy in self-supervised denoising tasks [4, 9, 11], additional computational overhead has emerged. We have noticed this issue and are actively exploring more efficient augmentation methods.
- To eliminate the influence of the checkerboard effect, we introduce prior smoothness loss. This measure significantly reduces the checkerboard effect and enhances the final denoising quality, but it also results in smooth effect to some extent, which is a common issue in self-supervised denoising [9]. We will actively explore further enhancement solutions.



(a) *GT* **(b)** *Noisy* **(c)** *AP-BSN* [4] **(d)** *AMSNet-P-E*

Fig. 9: Denoising effects under different frameworks. When the receptive field of the denoiser is unconstrained, AMSNet successfully removes noise, but AP-BSN suffers from identity mapping from noise to noise.

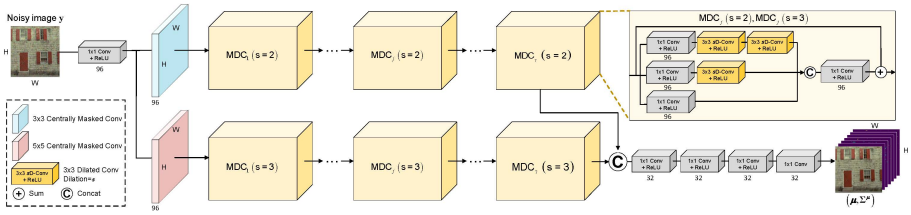
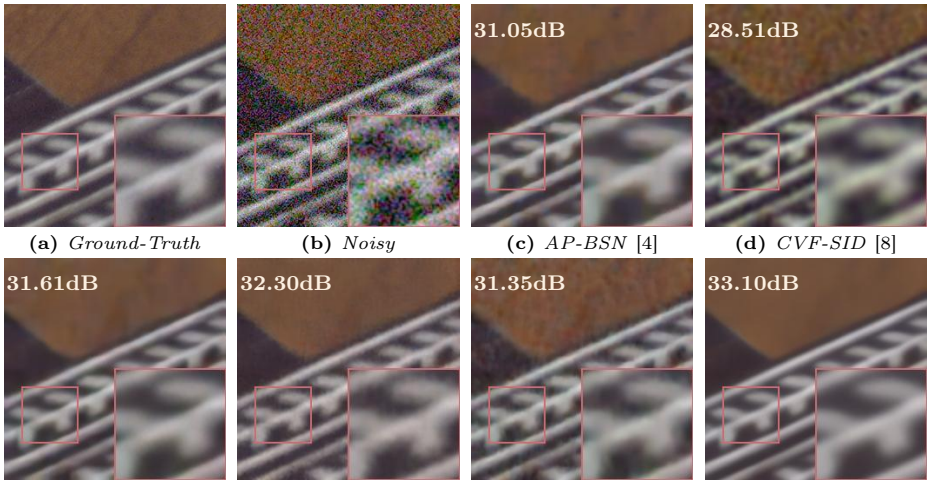


Fig. 10: Denoiser in AP-BSN [4]. The network is designed by D-BSN [12].

– Since the restored pixels are reconstructed from surrounding relevant noisy pixels with strong noisy interfence, there may be minor color shifting. As a common problem in selfsupervision, we still achieved better results. Possible solutions include adding a color correction module at the end of the network or incorporating local color consistency loss.
– Due to the introduction of the PD strategy, there might be a decrease in performance when handling sporadic texture regions. For example, the restoration effect may exhibit over-smoothing in areas with dense textures, which affects the final restoration quality.

1.10 More Visual Results

Fig. 11 and Fig. 12 show the denoising reusults on SIDD validation. Fig. 13 and Fig. 14 show the denoising reusults on PolyU Validation. Figs. 15 to 22 show the results of denoising real noisy images with different methods. We capture noisy images under two conditions: using a Redmi K30 Plus with an ISO setting of 6400 and using a Canon EOS M5 camera at ISO 25600 with an exposure time of 1/1250s. Our method achieves the best visual effects on real noisy images. Due to the substantial computational expense and resource occupation of LG-BPN, which is almost ten times that of conventional methods, we do not employ it in the denoising phase of the real self-captured noisy images. Fig. 23 shows the denoising results test at higher ISO and resolution (Nikon Z5, ISO 51200, 6040×4032), which demonstrates the effectiveness of our approach and we will add it in the future version.

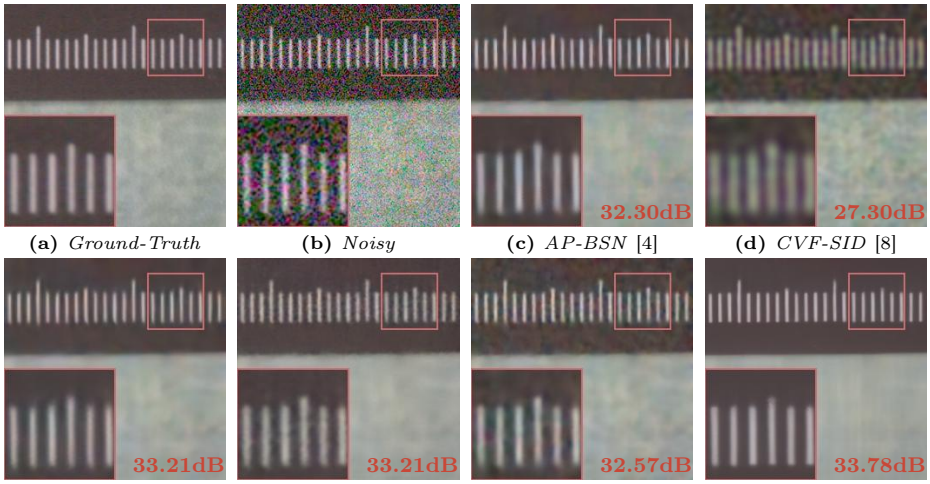


(a) *Ground-Truth* **(b)** *Noisy* **(c)** *AP-BSN* [4] **(d)** *CVF-SID* [8]
(e) *LGBPN* [11] **(f)** *BNN-LAN* [5] **(g)** *SCPGabN* [6] **(h)** *AMSNet-P-E*

Fig. 11: Visual comparison of our method against other denoising methods on the SIDD validation dataset.

References

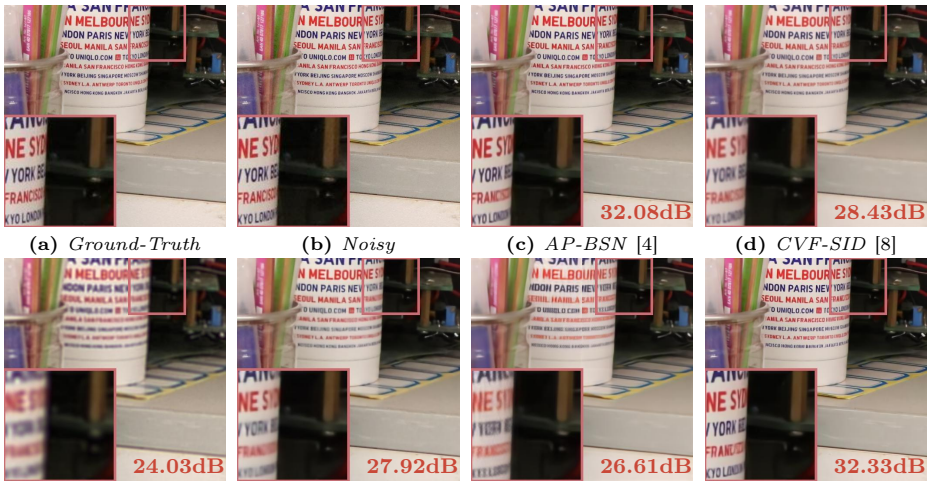
1. Abdelhamed, A., Lin, S., Brown, M.S.: A high-quality denoising dataset for smartphone cameras. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (June 2018)
2. He, K., Chen, X., Xie, S., Li, Y., Dollár, P., Girshick, R.: Masked autoencoders are scalable vision learners. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 16000–16009 (June 2022)
3. Krull, A., Buchholz, T.O., Jug, F.: Noise2void-learning denoising from single noisy images. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2129–2137 (July 2019)
4. Lee, W., Son, S., Lee, K.M.: Ap-bsn: Self-supervised denoising for real-world images via asymmetric pd and blind-spot network. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 17725–17734 (June 2022)
5. Li, J., Zhang, Z., Liu, X., Feng, C., Wang, X., Lei, L., Zuo, W.: Spatially adaptive self-supervised learning for real-world image denoising. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 9914–9924 (June 2023)
6. Lin, X., Ren, C., Liu, X., Huang, J., Lei, Y.: Unsupervised image denoising in real-world scenarios via self-collaboration parallel generative adversarial branches. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 12642–12652 (October 2023)
7. Martin, D., Fowlkes, C., Tal, D., Malik, J.: A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In: Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001. vol. 2, pp. 416–423. IEEE (2001)
8. Neshatavar, R., Yavartanoo, M., Son, S., Lee, K.M.: Cvf-sid: Cyclic multi-variate function for self-supervised image denoising by disentangling noise from image.



(a) *Ground-Truth* **(b)** *Noisy* **(c)** *AP-BSN* [4] **(d)** *CVF-SID* [8]
(e) *LG-BPN* [11] **(f)** *BNN-LAN* [5] **(g)** *SCPGabN* [6] **(h)** *AMSNet-P-E*

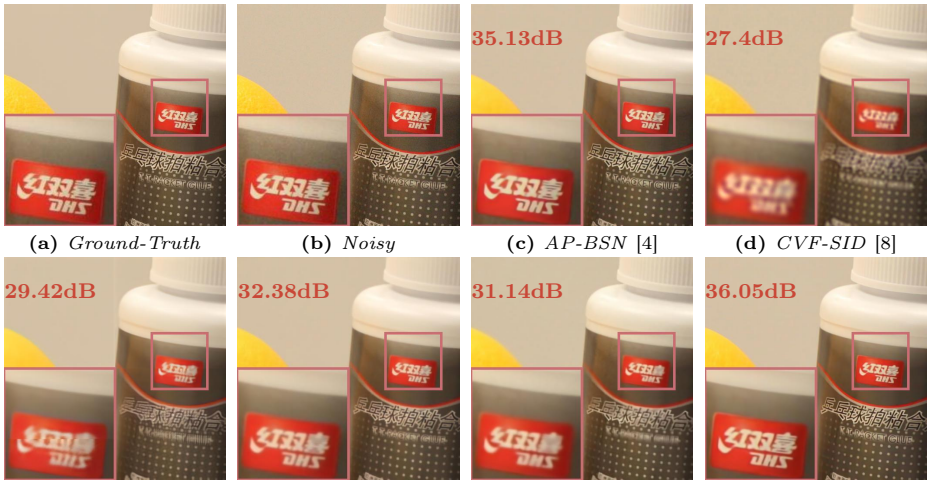
Fig. 12: Visual comparison of our method against other denoising methods on the SIDD validation dataset.

In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 17583–17591 (June 2022)
9. Pan, Y., Liu, X., Liao, X., Cao, Y., Ren, C.: Random sub-samples generation for self-supervised real image denoising. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 12150–12159 (October 2023)
10. Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., Lerer, A.: Automatic differentiation in pytorch (2017)
11. Wang, Z., Fu, Y., Liu, J., Zhang, Y.: Lg-bpn: Local and global blind-patch network for self-supervised real-world denoising. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 18156–18165 (June 2023)
12. Wu, X., Liu, M., Cao, Y., Ren, D., Zuo, W.: Unpaired learning of deep image denoising. In: European conference on computer vision (ECCV). pp. 352–368. Springer (2020)
13. Zhou, Y., Jiao, J., Huang, H., Wang, Y., Wang, J., Shi, H., Huang, T.: When awgn-based denoiser meets real noises. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 34, pp. 13074–13081 (2020)



(a) *Ground-Truth* **(b)** *Noisy* **(c)** *AP-BSN* [4] **(d)** *CVF-SID* [8]
(e) *LG-BPN* [11] **(f)** *BNN-LAN* [5] **(g)** *SCPGabN* [6] **(h)** *AMSNet-P-E*

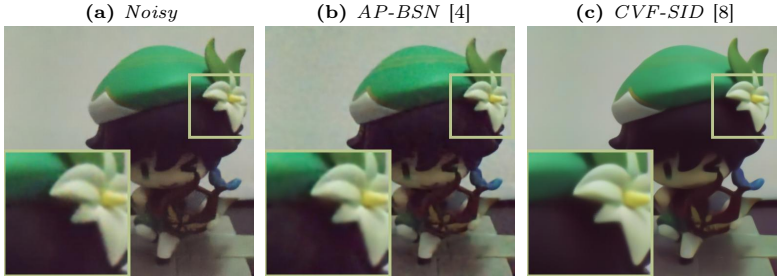
Fig. 13: Visual comparison of our method against other denoising methods on the PolyU validation dataset.



(a) *Ground-Truth* **(b)** *Noisy* **(c)** *AP-BSN* [4] **(d)** *CVF-SID* [8]
(e) *LG-BPN* [11] **(f)** *BNN-LAN* [5] **(g)** *SCPGabN* [6] **(h)** *AMSNet-P-E*

Fig. 14: Visual comparison of our method against other denoising methods on the PolyU validation dataset.

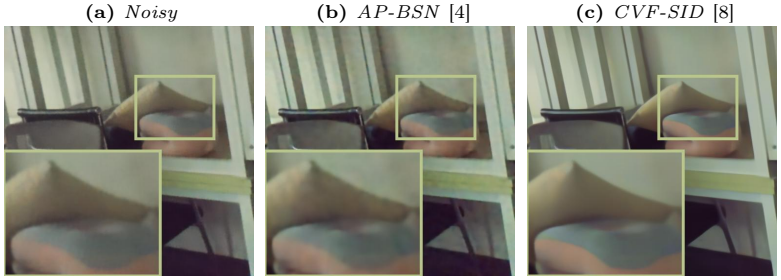
(a) *Noisy* **(b)** *AP-BSN* [4] **(c)** *CVF-SID* [8]



(d) *BNN-LAN* [5] **(e)** *SCPGabN* [6] **(f)** *AMSNet-P-E*

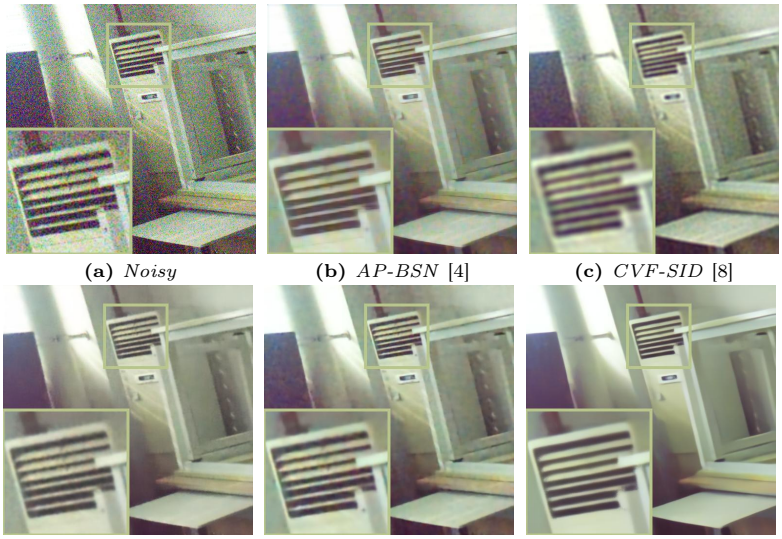
Fig. 15: The denoising results of noisy images taken with Redmi K30.

(a) *Noisy* **(b)** *AP-BSN* [4] **(c)** *CVF-SID* [8]



(d) *BNN-LAN* [5] **(e)** *SCPGabN* [6] **(f)** *AMSNet-P-E*

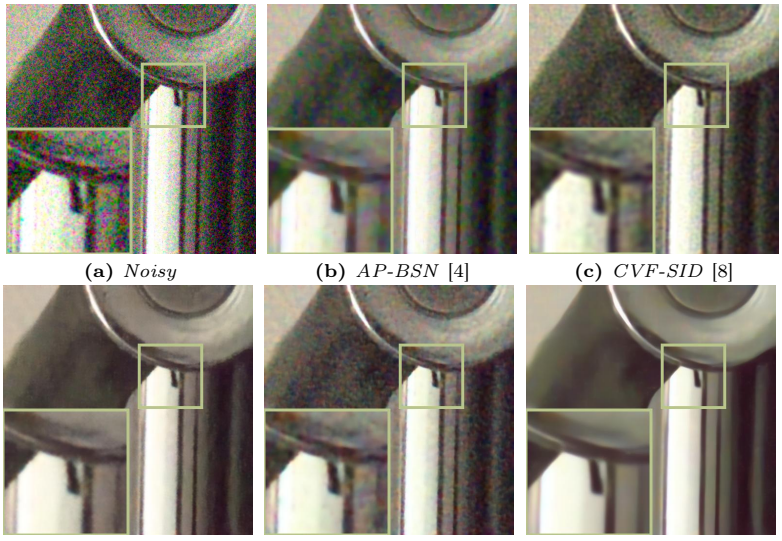
Fig. 16: The denoising results of noisy images taken with Redmi K30.



(a) *Noisy* **(b)** *AP-BSN* [4] **(c)** *CVF-SID* [8]

(d) *BNN-LAN* [5] **(e)** *SCPGabN* [6] **(f)** *AMSNet-P-E*

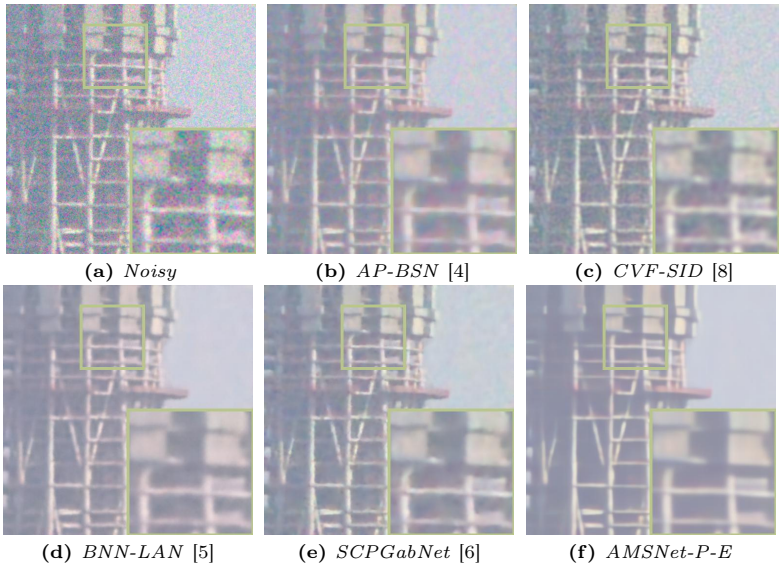
Fig. 17: The denoising results of noisy images taken with Redmi K30.



(a) *Noisy* **(b)** *AP-BSN* [4] **(c)** *CVF-SID* [8]

(d) *BNN-LAN* [5] **(e)** *SCPGabNet* [6] **(f)** *AMSNet-P-E*

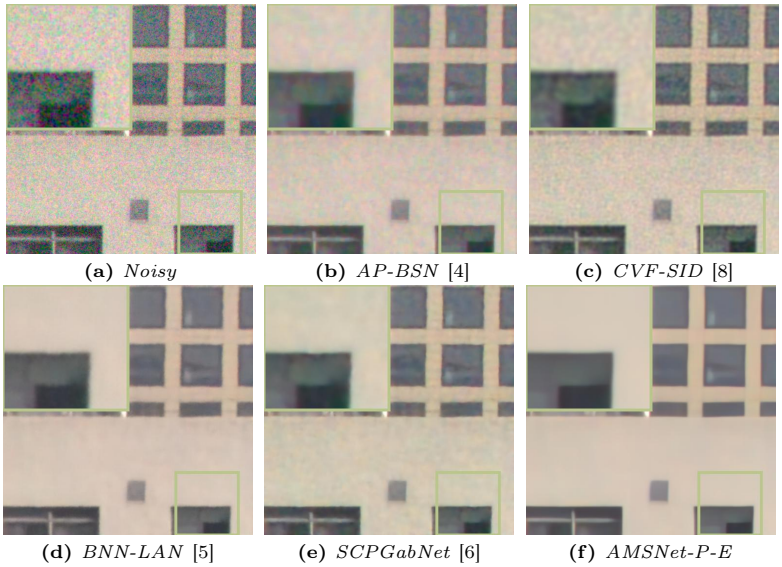
Fig. 18: The denoising results of noisy images taken with Canon EOS M5 camera.



(a) *Noisy* **(b)** *AP-BSN* [4] **(c)** *CVF-SID* [8]

(d) *BNN-LAN* [5] **(e)** *SCPGabNet* [6] **(f)** *AMSNet-P-E*

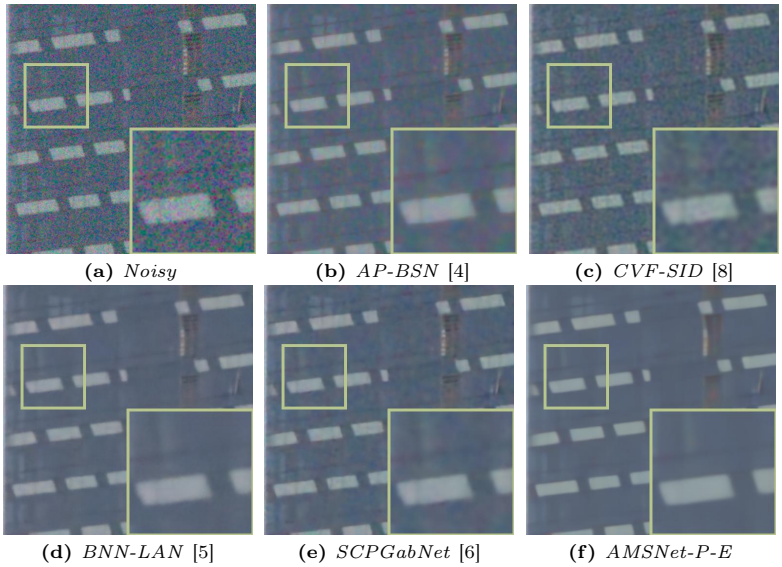
Fig. 19: The denoising results of noisy images taken with Canon EOS M5 camera.



(a) *Noisy* **(b)** *AP-BSN* [4] **(c)** *CVF-SID* [8]

(d) *BNN-LAN* [5] **(e)** *SCPGabNet* [6] **(f)** *AMSNet-P-E*

Fig. 20: The denoising results of noisy images taken with Canon EOS M5 camera.



(a) *Noisy* **(b)** *AP-BSN* [4] **(c)** *CVF-SID* [8]

(d) *BNN-LAN* [5] **(e)** *SCPGabNet* [6] **(f)** *AMSNet-P-E*

Fig. 21: The denoising results of noisy images taken with Canon EOS M5 camera.

(a) *Noisy* **(b)** *AP-BSN* [4] **(c)** *CVF-SID* [8]

(d) *BNN-LAN* [5] **(e)** *SCPGabNet* [6] **(f)** *AMSNet-P-E*

Fig. 22: The denoising results of noisy images taken with Canon EOS M5 camera.

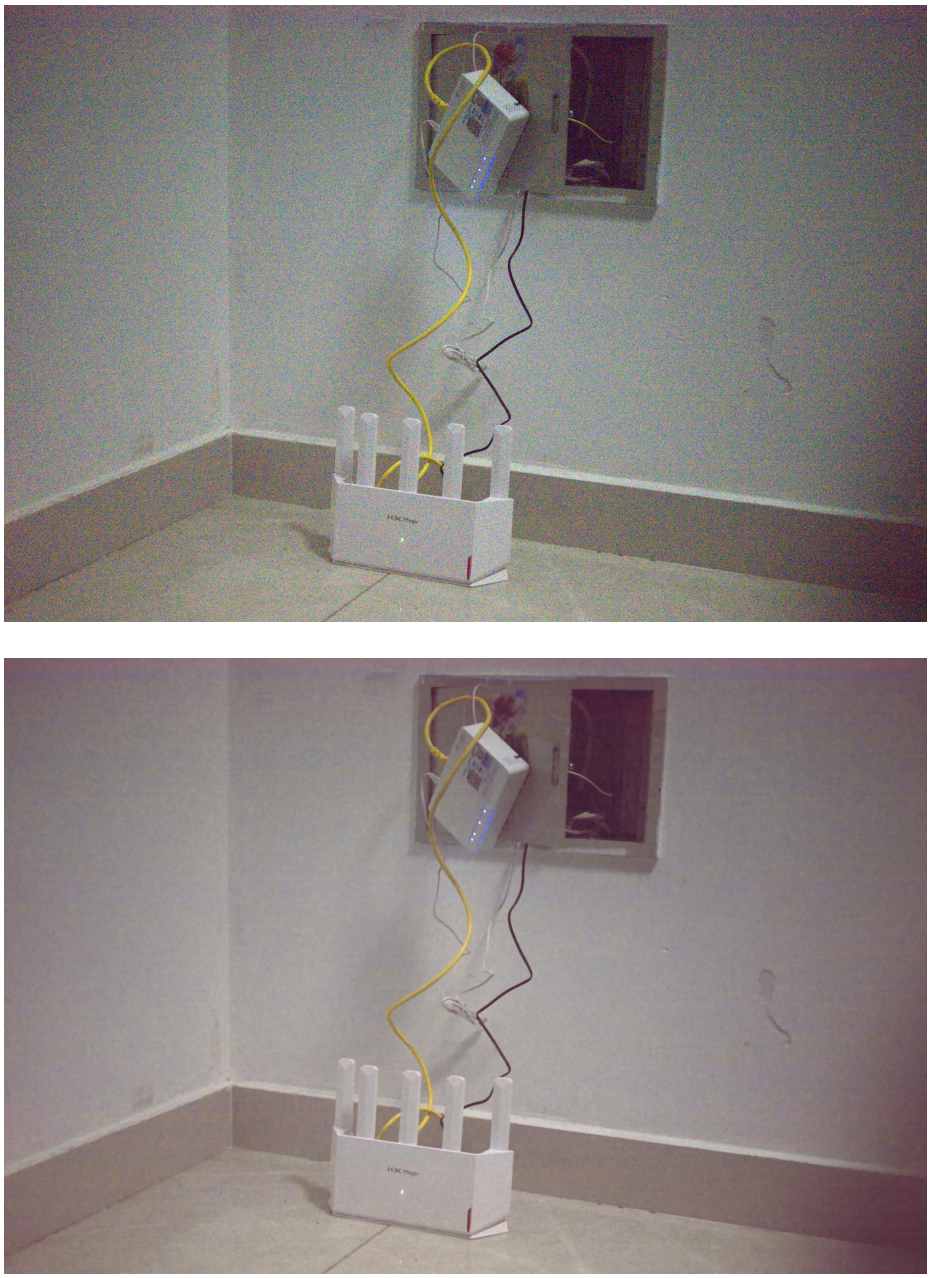


Fig. 23: The denoising results of noisy images taken with Nikon Z5 (ISO 51200, 6040×4032).