DualDn: Dual-domain Denoising via Differentiable ISP

Supplementary Material

1 Experiment Details

Here we provide additional results for denoising synthetic images, real-captured images, and the DND benchmark. Additionally, our project website (`https://openimaginglab.github.io/DualDn/`) offers clear and detailed comparisons.

1.1 Results for Synthetic Datasets

As described in the main text, we trained 3 different backbones using the same training strategy, with random noise level $K \sim [0.0002, 0.02]$ and amplification ratio $\alpha \sim [0, 1]$ on a synthetic dataset. In Tab. 1, we present a comprehensive comparison table testing different K and α. Our DualDn consistently outperforms single-domain methods across all test cases. Moreover, we provide additional visualization denoising results of synthetic datasets in Fig. 7 and Fig. 8.

Table 1: Full comparison of denoising performance. Dual-domain denoising outperforms single-domain methods in every case of noise levels K and amplification ratio α.

		K = 0.0002						**K = 0.002**						**K = 0.02**					
Backbones		SwinIR		MIRNet-v2		Restormer		SwinIR		MIRNet-v2		Restormer		SwinIR		MIRNet-v2		Restormer	
		PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
	Raw denoising	36.77	0.938	40.89	0.972	41.44	0.972	33.80	0.873	36.76	0.942	36.98	0.941	26.96	0.595	31.47	0.865	32.08	0.873
α = **0.2**	sRGB denoising	36.11	0.928	36.12	0.945	42.42	0.975	32.71	0.820	34.78	0.921	37.72	0.944	26.32	0.513	30.20	0.806	33.09	0.890
	Ours	**37.30**	**0.943**	**41.07**	**0.972**	**42.86**	**0.977**	**34.59**	**0.891**	**37.21**	**0.944**	**38.60**	**0.953**	**28.95**	**0.709**	**32.35**	**0.883**	**33.98**	**0.906**
	Raw denoising	35.53	0.928	39.33	0.964	39.71	0.962	32.49	0.853	35.28	0.929	35.45	0.927	25.84	0.576	30.03	0.838	30.65	0.850
α = **0.5**	sRGB denoising	35.64	0.927	35.03	0.935	40.92	0.968	31.71	0.808	33.61	0.905	36.48	0.935	25.41	0.519	28.88	0.777	31.88	0.872
	Ours	**36.29**	**0.936**	**39.62**	**0.964**	**41.32**	**0.970**	**33.62**	**0.884**	**35.81**	**0.931**	**37.18**	**0.941**	**27.89**	**0.694**	**31.05**	**0.862**	**32.64**	**0.888**
	Raw denoising	34.39	0.919	38.00	0.956	38.24	0.953	31.30	0.839	34.04	0.916	34.16	0.914	24.82	0.584	28.80	0.817	29.44	0.831
α = **0.8**	sRGB denoising	34.84	0.923	34.06	0.925	39.59	0.960	30.52	0.793	32.57	0.891	35.30	0.923	23.95	0.500	27.60	0.754	30.62	0.848
	Ours	**35.05**	**0.926**	**38.36**	**0.956**	**39.97**	**0.962**	**32.47**	**0.870**	**34.63**	**0.919**	**35.94**	**0.930**	**26.53**	**0.664**	**29.93**	**0.845**	**31.48**	**0.872**

1.2 Results for Smartphone Datasets

In the main text (Sec. 4.2), we present visual comparisons demonstrating that DualDn achieves higher-quality denoising results compared to the camera ISP. To fairly evaluate the generalization capability of DualDn, we conducted a quantitative comparison with the camera denoising results through a user study. We recruited 50 participants to compare 29 results. For each result, participants were asked to respond to three questions shown in Fig. 1. The distribution of

preferences across methods indicates that our results are more favored by human subjects. Specifically, our method (represented by red bars) received significantly more choices for less noise, more details, and better visual perception compared to the camera ISP (represented by blue bars).

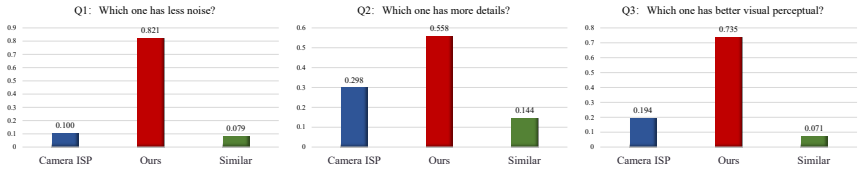


Fig. 1: Distributions for the three questions in the smartphone dataset's user study.

1.3 Results for DND Benchmark

In the main text (Sec. 4.1), we demonstrated that DualDn generally outperforms the intermediate Raw output. However, due to the zipper pattern in the DND's ground truth caused by its overly simplified demosaicing process, our final sRGB result's PSNR/SSIM scores increased only slightly. As shown in Fig. 3, the left column displays the intermediate raw output of DualDn evaluated with DND's black-box ISP, while the right column shows the DualDn's output with our ISP. The left column achieves better metrics in Fig. 3 since the zipper pattern around the image edges produced by DND's ISP is more consistent with its ground truth.

However, this introduces a dilemma for fully testing DualDn on the DND benchmark: the DND ISP is not open-source, so we cannot use it for image processing, and training DualDn with a simplified demosaicing algorithm results in obvious denoising artifacts due to the presence of zipper patterns, as we will verify in Sec. 2.2. Nevertheless, we recruited 20 participants to conduct a Mean Opinion Score (MOS) experiment with image quality ratings ranging from 1 (Bad) to 5 (Good). Each participant was asked to compare 20 denoising results generated by CycleISP, Restormer, and DualDn. The average score in Fig. 2 indicates that people perceive our DualDn to have better visual quality. Moreover, we provide additional visualization results in Fig. 9.

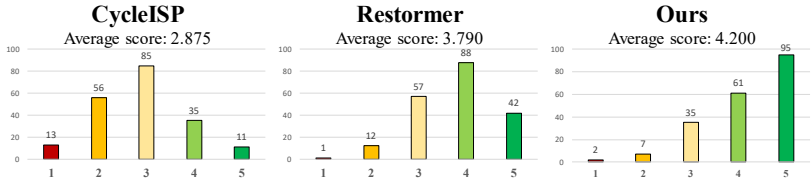


Fig. 2: MOS experiment with visual quality comparison in the DND Benchmark.

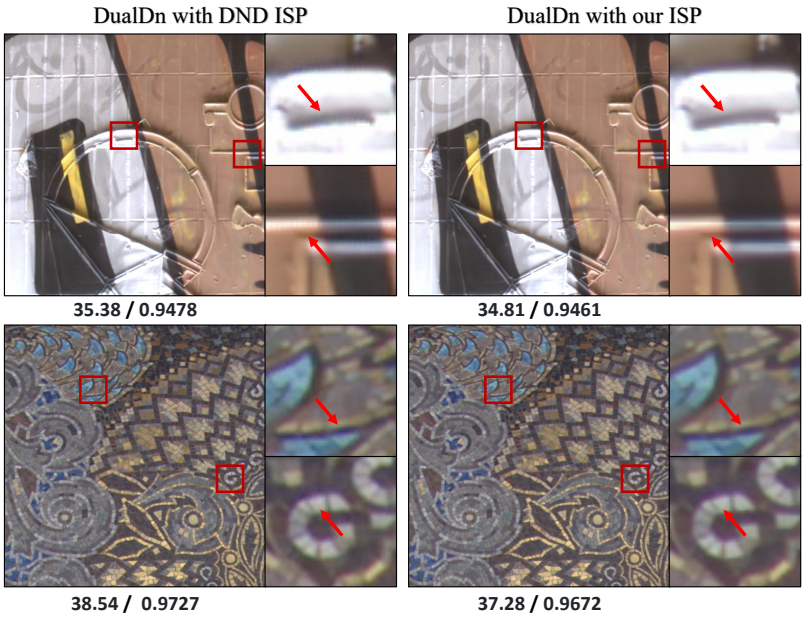


Fig. 3: Test images with slight noise on DND benchmark with different ISPs. As can be seen in the zoom-in patches, DND ISP produces obvious zipper patterns around image edges. Although two results are visually similar, the evaluation matrixs (PSNR/SSIM) drop a lot due to DND ISP's over-simplified demosaicing.

2 Method Details

2.1 Noise Map Fusion Block

In our work, we present a Noise map Fusion Block (NFB) to effectively fuse the noise map and input feature, aiming to improve dual-domain denoising performance. As mentioned in the main text (Sec. 3.3), we adopt both raw and sRGB noise maps, so the NFB needs to be fit for both domains that denoising applies. We use a switch mechanism (shown in the bottom left of Fig. 4) to change the formation process of noise maps in different domains, and it switches to the route with ISP when applied in the sRGB domain.

Specifically, as shown in Fig. 4, given the **Input** $\in \mathbb{R}^{H\times W\times N}$, where N equals 4 or 3 (packed raw or sRGB images), we concatenate it with the corresponding noise map after pixel-wise and depth-wise convolutions. Then, we expand the number of channels from N to C for denoising. Finally, a skip connection is adopted as residual learning for the final **Output**. As network architecture is a vital element in feature fusion, we also conduct an experiment on different fusion methods adopted in previous studies to find the optimal fusion mechanism. As shown in Tab. 2, our DualDn can effectively fuse noise maps under the dual-domain denoising network architecture.

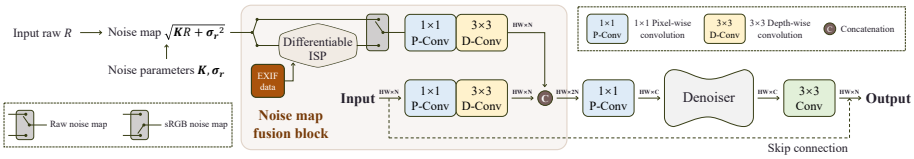


Fig. 4: Detailed architecture of noise map fusion block.

Table 2: Testing different fusion methods at different noise levels K.

	Method:	**Gated** [2]		**SKFF** [7]		**Attention** [6]		**Ours**	
Noise Level		PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
K**=0.002**		38.48	0.952	38.55	0.953	38.57	0.953	**38.61**	**0.953**
K**=0.02**		33.82	0.903	33.88	0.905	33.90	0.905	**33.98**	**0.906**

2.2 Differentiable AHD Demosaicing

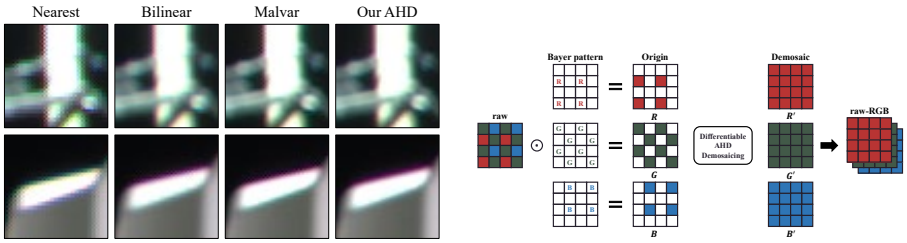
ISPs adopt demosaicing algorithms to interpolate information from the raw image space to the sRGB image space. Simple demosaicing algorithms, such as nearest or bilinear interpolation, produce zipper patterns around image edges even in the absence of noise, as shown in Fig. 5a. Furthermore, if a simple demosaicing algorithm is adopted, high-frequency noise in the raw image will be entangled by the demosaicing process and amplified by subsequent ISP modules, resulting in images with more severe noise. To verify this, we trained DualDn with different demosaicing algorithms and tested the final results. As shown in Tab. 3, simple demosaicing algorithms can "distract" dual-domain denoisers, causing them to misinterpret the zipper pattern as extra noise and thereby reducing denoising performance to some extent.

To that end, we adopt the adaptive homogeneity-directed (AHD) demosaicing algorithm [3] and modify it into a differentiable one. Given the original raw image, we extract its corresponding R-G-B channels using the Bayer pattern and use the AHD algorithm for interpolation to obtain the final raw-RGB image. Fig. 5b explains the overall scheme for our AHD algorithm, and Algorithm 1 shows the details. As a result, our simplified AHD algorithm reduces the zipper pattern and improves the performance of DualDn, as shown in Fig. 5a and Tab. 3.

Lastly, we test our ISP's robustness with a widely-used one [1]. Using the well-known RawPy [4] as a reference, on 50 clean raw images, our ISP achieves an average PSNR of 41.7dB with better visual quality, as shown in Fig. 6.

Table 3: Testing DualDn (with various demosaicing algorithms) under different K.

	Nearest			**Bilinear**			**Malvar** [5]			**Ours**		
Noise Level	PSNR↑	SSIM↑	NIQE↓	PSNR↑	SSIM↑	NIQE↓	PSNR↑	SSIM↑	NIQE↓	PSNR↑	SSIM↑	NIQE↓
K**=0.002**	39.16	0.947	6.969	40.02	0.957	7.384	40.49	0.962	6.148	**41.24**	**0.968**	**6.047**
K**=0.02**	34.63	0.895	7.239	35.18	0.905	7.441	35.57	0.919	7.076	**36.09**	**0.929**	**7.009**



(a) Visual comparisons of different demosaicing. **(b)** The overall scheme of AHD demosaicing.

Fig. 5: Explanations of our AHD demosaicing and compare it with other algorithms.

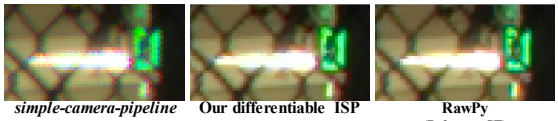


Fig. 6: Compare DualDn ISP with other ISPs on clean raw images.

Algorithm 1: Our AHD demosaicing

Input : original raw r and channels R, G, B
Output: demosaic channels $R^{'}$, $G^{'}$, $B^{'}$

Initialization: bilinear-interpolated R_0, G_0, B_0;
Calculate row/column direction interpolation:
begin
  $G_{row} = conv_{2d}(r, f)$, $G_{col} = conv_{2d}(r, f^T)$;
  $R_{row} = Blur(R_0 - G_0) + G_{row}$, $R_{col} = Blur(R_0 - G_0) + G_{col}$;
  $B_{row} = Blur(B_0 - G_0) + G_{row}$, $B_{col} = Blur(B_0 - G_0) + G_{col}$;
end
where, $f = [-1, 2, 2, 2, -1]/4$, $Blur()$ is a 3×3 Gaussian blur kernel;
Convert (R_0, G_0, B_0) **to Lab color space** (L, a, b);
Calculate row/column gradients $\delta_{row}, \delta_{col}$:
for $i = L, a, b$ **do**
  $\delta_{row} = \delta_{row} + conv_{2d}(i, h_1) + conv_{2d}(i, h_2)$;
  $\delta_{col} = \delta_{col} + conv_{2d}(i, h_1^T) + conv_{2d}(i, h_2^T)$;
end
where, $h_1 = [1, 2, -3, 0, 0]$, $h_2 = [0, 0, -3, 2, 1]$;
Select the row/column interpolation with smaller δ:
for $j = R, G, B$ **do**
  if $\delta_{row} < \delta_{col}$ **then**
    $j^{'} = j_{row}$;
  else
    $j^{'} = j_{col}$;
  end
end

References

1. Abdelhamed, A.: Simple camera pipeline. `https://github.com/AbdoKamel/simple-camera-pipeline` (2020), [Online; accessed 1-March-2024]
2. Chen, L., Chu, X., Zhang, X., Sun, J.: Simple baselines for image restoration. In: ECCV. pp. 17–33. Springer (2022)
3. Hirakawa, K., Parks, T.W.: Adaptive homogeneity-directed demosaicing algorithm. IEEE TIP **14**(3), 360–369 (2005)
4. Letmaik: Rawpy. `https://github.com/letmaik/rawpy`, [Online; accessed 1-March-2024]
5. Malvar, H.S., He, L.w., Cutler, R.: High-quality linear interpolation for demosaicing of bayer-patterned color images. In: 2004 IEEE International Conference on Acoustics, Speech, and Signal Processing. vol. 3, pp. iii–485. IEEE (2004)
6. Zamir, S.W., Arora, A., Khan, S., Hayat, M., Khan, F.S., Yang, M.H.: Restormer: Efficient transformer for high-resolution image restoration. In: CVPR. pp. 5728–5739 (2022)
7. Zamir, S.W., Arora, A., Khan, S., Hayat, M., Khan, F.S., Yang, M.H., Shao, L.: Learning enriched features for fast image restoration and enhancement. IEEE TPAMI **45**(2), 1934–1948 (2022)

Fig. 7: Additional qualitative comparison on synthetic datasets. (under amplification ratio $\alpha = 0$, various noise level K)

Fig. 8: Additional qualitative comparison on synthetic datasets. (under amplification ratio $\alpha = 1$, various noise level K)

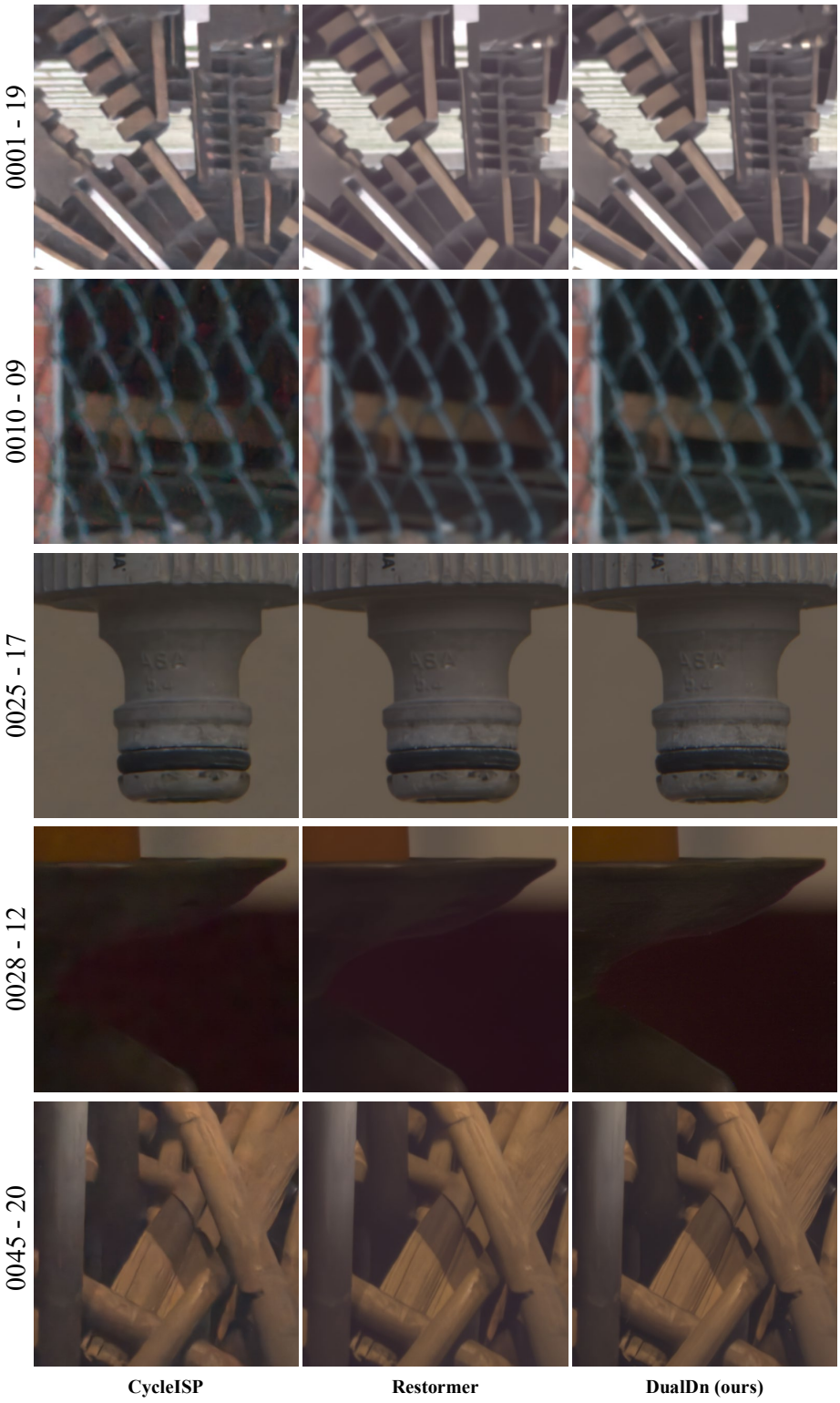


Fig. 9: Additional qualitative comparison on DND benchmark.