Frequency-Consistent Optimization for Image Enhancement Networks (Supplementary Material)

Bing Li*
bing0123@mail.ustc.edu.cn
Naishan Zheng*
nszheng@mail.ustc.edu.cn
Qi Zhu
zqcrafts@mail.ustc.edu.cn
Jie Huang
hj0117@mail.ustc.edu.cn
Feng Zhao ✉
fzhao956@ustc.edu.cn

Department of Automation
University of Science and Technology of China
Hefei 230027, China

1 Model Architecture

In this section, we provide a detailed description of the baseline architectures that were incorporated with the FDI module in our study.

MPRNet. MPRNet [7] has achieved remarkable success as a multi-stage architecture for progressive image restoration. Nevertheless, our primary purpose was to leverage the capabilities of straightforward single-stage networks. As a result, we selected only the first stage of the MPRNet as a baseline for comparison. Additionally, we opted to omit the application of multi-patch hierarchy on the input image. Instead, our input consists solely of the original image, while the output represents the final enhanced image.

SID. We simply modified the ConvNet of the SID [2] by reducing the number of input and output channels to 3 as a baseline, as it is a standard U-Net architecture.

UIEC^2. UIEC^2 [6] is a representative end-to-end trainable network architecture for underwater image enhancement tasks, which consists of three blocks: an RGB pixel-level block, an HSV global-adjust block, and an attention map block. Despite being a multi-branch architecture, we have chosen UIEC^2 as our baseline to demonstrate the effectiveness of our proposed plug-and-play module. Specifically, we incorporate three FDI modules in the middle of each of the three blocks mentioned earlier. By comparing our method against the UIEC^2 baseline, we aim to highlight the benefits and advancements that can be achieved in the context of complex multi-branch networks.

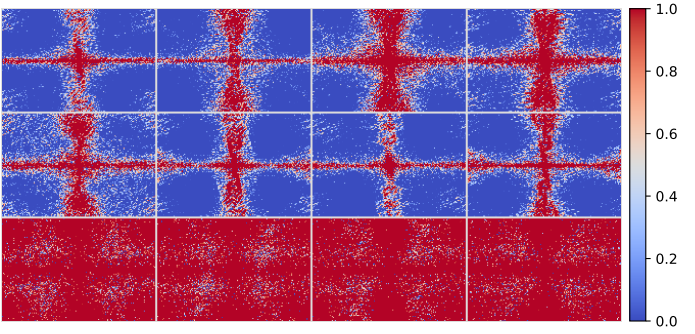


Figure 1: Visulation of Feature Spectra. Row 1, the spectra of the original features (F_{in}); row 2, the spectra of the reduced original features ($\hat{F}_{in}$); row 3, the spectra of the reduced decorrelated features ($\hat{F}_n$). Note that F_{in} is the input features of FDI module while $\hat{F}_{in}$ and $\hat{F}_n$ are the output features. It is evident that following the FDI processing, the output features exhibit a combination of frequency-independent features, characterized by a consistent response across different frequencies, as well as the original frequency-dependent features.

2 Visulization of Feature Spectra

In this section, we present the spectra of the features before and after the FDI module. Fig. 1 illustrates the observed changes in the spectra. It is evident that following the FDI processing, the output features exhibit a combination of frequency-independent features, characterized by a consistent response across different frequencies, as well as the original frequency-dependent features. This demonstrates that the FDI module introduces frequency-independent components while preserving the inherent frequency-dependent characteristics to the output features. The incorporation of both types of features contributes to a more comprehensive representation of the data, facilitating improved enhancement results. Note that we have separately normalized each spectrum for visual clarity.

3 Additional Ablation Studies

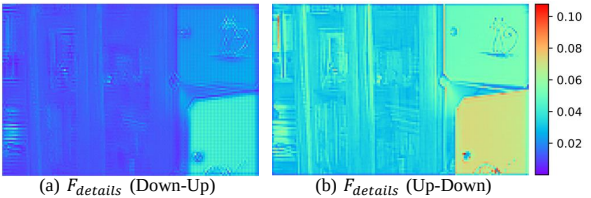
In this section, we conducted additional ablation experiments in order to demonstrate the rationality of the components of the FDI module. We list our additional ablation configurations as follows:

1). The effectiveness of Frequency Decorrelation Part: We exclusively employed the random shuffle operation as the plug-in module to investigate the performance of Frequency Decorrelation Part. 2). The effectiveness of Integration Part: To demonstrate the crucial role of integrating the initial information, we insert the FDI module into the baseline without incorporating Integration Part. 3). The effectiveness of Decorrelation Normalization (*i.e.* whitening): We replace the DN with the classic Instance Normalization [5] and PCA whitening to prove the superiority of DN in the FDI module. 4). The effectiveness of up-down expand operator: We replace the up-down operator with down-up-like ones.

As shown in Tab. 1, we can draw the following conclusions: 1). The exclusion of either the Frequency Decorrelation Part or the Integration Part from our FDI module results in a noticeable decline in performance, indicating that both parts of our FDI module design have contributed to the performance improvement. 2). The utilization of solely random shuffle

Table 1: The ablation studies on the LOL dataset. The best results are highlighted in bold.

Baseline	Method	PSNR(↑)	SSIM(↑)	LPIPS(↓)
SID	baseline	19.16	0.7862	0.440
	w/o Frequency Decorrelation part	17.52	0.8102	0.258
	w/o Integration part	19.71	0.8232	0.230
	replace DN with IN	20.28	0.8284	0.234
	w/ FDI (ours)	**21.07**	**0.8360**	**0.227**
MPRNet	baseline	20.13	0.8170	0.266
	w/o Frequency Decorrelation part	18.72	0.8053	0.269
	w/o Integration part	20.85	0.8456	0.207
	replace DN with IN	21.10	0.8378	0.208
	replace ZCA with PCA	19.51	0.8241	0.191
	replace up-down with down-up	21.02	0.8239	0.190
	w/ FDI (ours)	**21.59**	**0.8512**	**0.154**

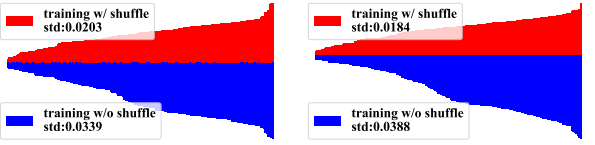


(a) $F_{details}$ (Down-Up) (b) $F_{details}$ (Up-Down)

Figure 2: $F_{details}$ feature map of compared down-up (less details) and our used up-down (more details) in expand operator.

operation results in a significant performance decrease in performance, even worse than the baseline. We attribute this performance drop to the inherent instability introduced by random shuffle during training, leading to suboptimal results. However, when both the DN and random shuffle are employed, performance improvement is observed. In this case, random shuffle helps prevent the network from being biased toward specific features, thereby enhancing the influence of the DN during training. 3). The absence of Integration part does not lead to a performance decline compared to the baseline. This is because the skip connections employed by SID and MPRNet, which combine deep features with shallow features, mitigate the information loss caused by the FDI module. However, it is important to note that while the skip connections can compensate for some information loss, the application of the Integration is still necessary to complement the information at the FDI insertion layer. 4). By replacing DN with IN in the FDI module, there is also a performance enhancement compared to the baseline. This improvement further implies the effectiveness of other designs of our module. Additionally, the FDI module with DN outperforms the FDI with IN, emphasizing the importance of the decorrelation capability introduced by DN in the optimization process. In summary, all the key designs contribute to the best performance of the module.

To further substantiate the superiority of our design, we visualize feature maps to demonstrate the indispensability of the up-down expand operator and random shuffle operator. As shown in Fig. 2, the up-down expand operator extracts more texture details compared to the down-up substitute. As observed in Fig. 3,the FDI with random shuffle in training achieves



(a) activations of 1.png (b) activations of 778.png

Figure 3: FDI layer channel activations during testing when training with shuffling or not. The activations have a lower standard deviation when applying the shuffle operator during training

more stable activations (lower deviation) across channels (means less biased to specific channels).

4 Additional Qualitative Results

In this section, we provide more qualitative results. Fig. 4, Fig. 5, Fig. 6, and Fig. 7 are visual results on the LOL dataset. We additionally introduced Zero-DCE++ [2] for visual comparison. Fig. 8, Fig. 9, Fig. 10, and Fig. 11 are visual results on the UIEB dataset. We can observe that other methods tend to exhibit color shifts or artifacts, while our methods can achieve consistently satisfactory visual results.

5 Retouching Bechmark

We evaluate the proposed module on the Five-K [1] dataset for the retouching task. We select MPRNet and CSRNet [3] as the baseline, and the evaluation is shown in Tab. 2 and shows FDI's effectiveness.

Table 2: Quantitative comparison on five-K dataset.

Method	MPRNet	MPRNet-FDI	CSRNet	CSRNet-FDI
PSNR(dB)↑	22.58	**23.13(+0.55)**	23.57	**23.81(+0.24)**
SSIM↑	0.9289	**0.9385(+0.0096)**	0.9394	**0.9415(+0.0021)**
LPIPS↓	0.0786	**0.0702(+0.0084)**	0.0749	**0.0708(+0.0041)**

References

[1] Vladimir Bychkovsky, Sylvain Paris, Eric Chan, and Frédo Durand. Learning photographic global tonal adjustment with a database of input / output image pairs. In *The Twenty-Fourth IEEE Conference on Computer Vision and Pattern Recognition*, 2011.

[2] Chen Chen, Qifeng Chen, Jia Xu, and Vladlen Koltun. Learning to see in the dark. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3291–3300, 2018.

[3] Jingwen He, Yihao Liu, Yu Qiao, and Chao Dong. Conditional sequential modulation for efficient global image retouching. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIII 16*, pages 679–695. Springer, 2020.

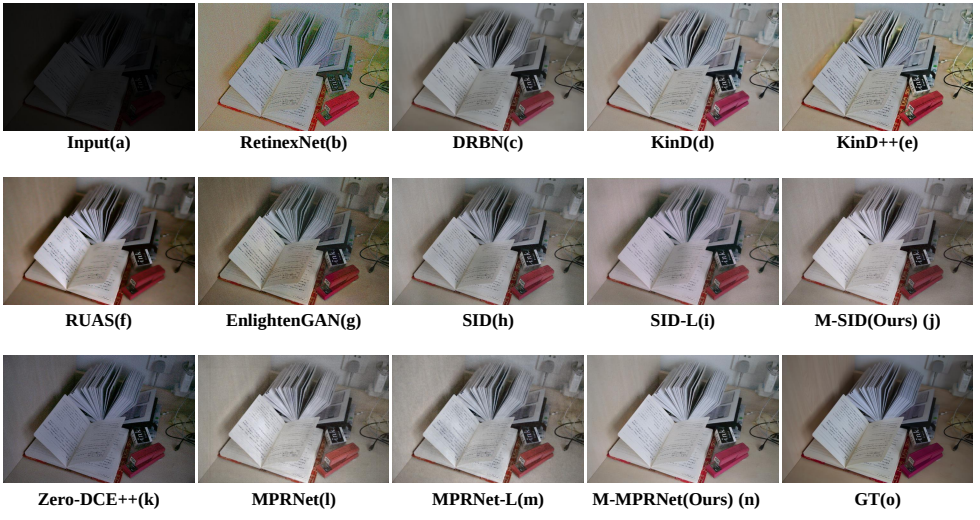


Figure 4: Visual comparison of different LLE methods on the LOL dataset.

[4] Chongyi Li, Chunle Guo, and Chen Change Loy. Learning to enhance low-light image via zero-reference deep curve estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(8):4225–4238, 2021.

[5] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*, 2016.

[6] Yudong Wang, Jichang Guo, Huan Gao, and Huihui Yue. Uiecˆ 2-net: Cnn-based underwater image enhancement using two color space. *Signal Processing: Image Communication*, 96:116250, 2021.

[7] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, Ming-Hsuan Yang, and Ling Shao. Multi-stage progressive image restoration. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14821–14831, 2021.

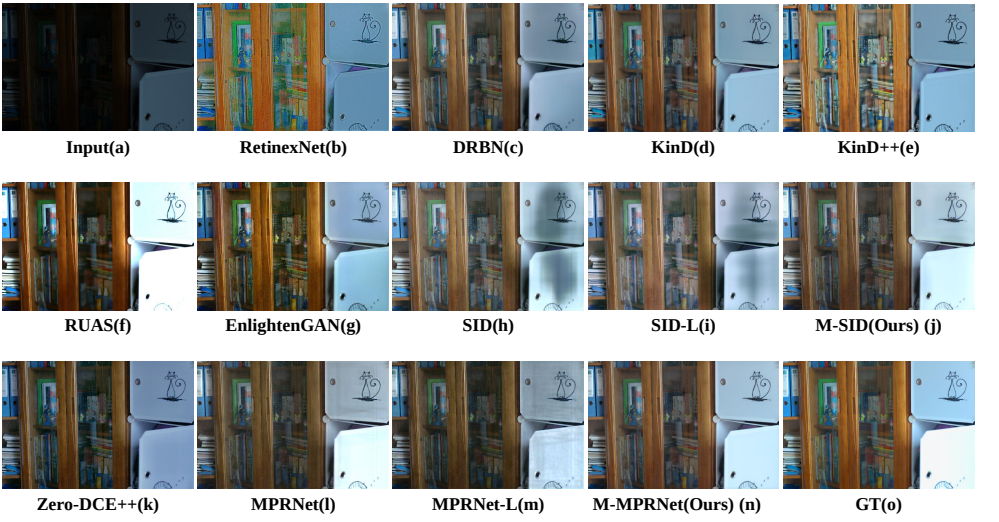


Figure 5: Visual comparison of different LLE methods on the LOL dataset.

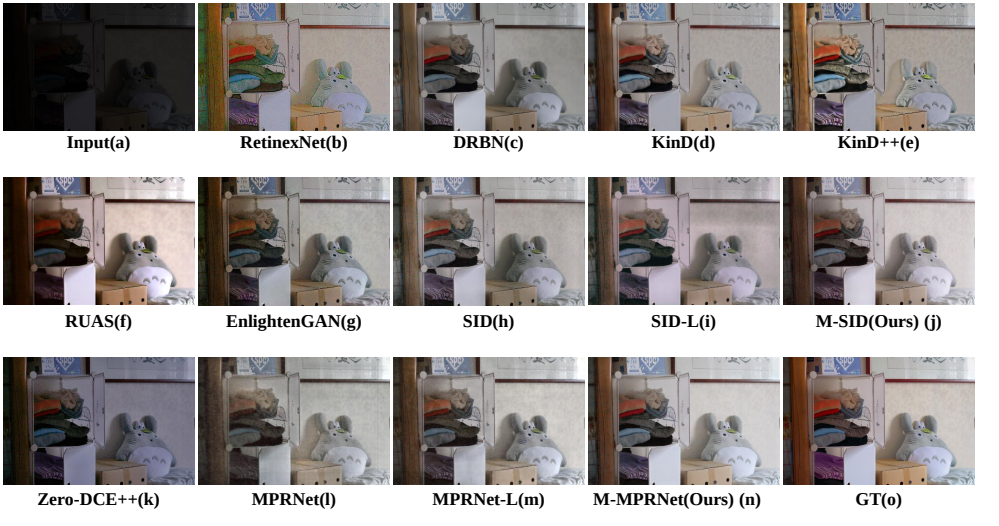


Figure 6: Visual comparison of different LLE methods on the LOL dataset.

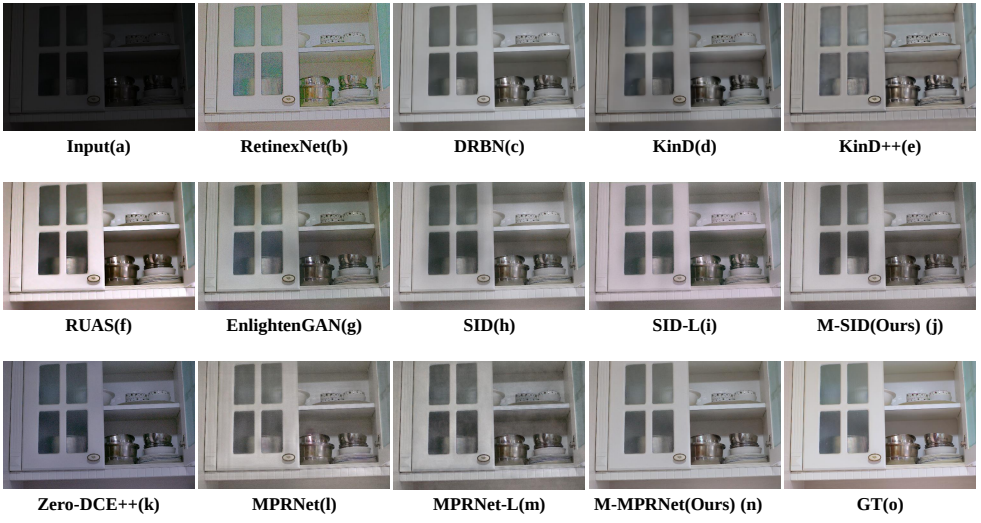


Figure 7: Visual comparison of different LLE methods on the LOL dataset.

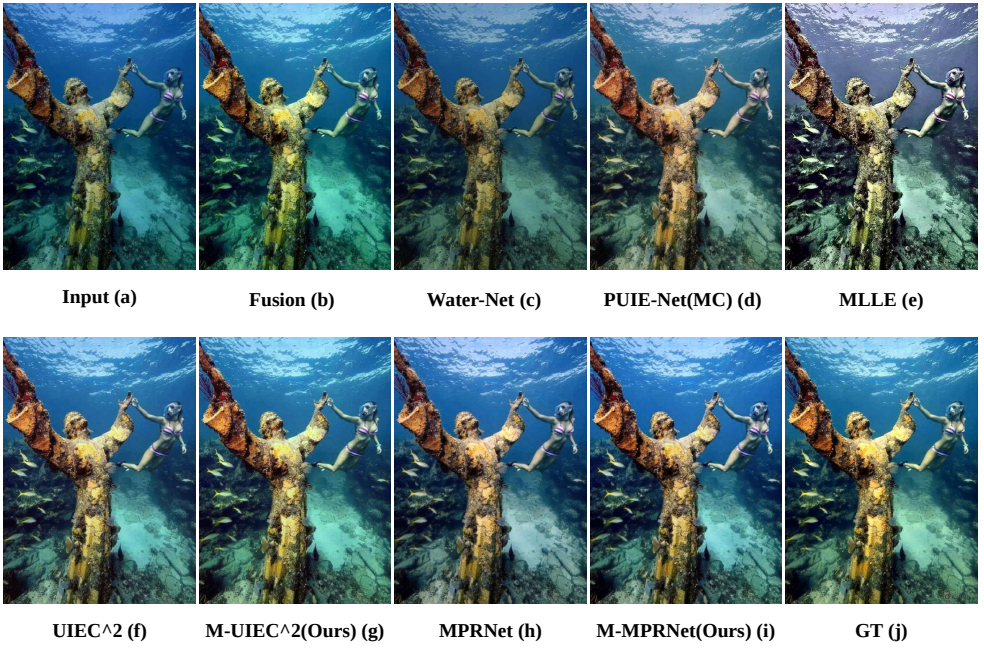


Figure 8: Visual comparison of different UIE methods on the UIEB dataset.

Figure 9: Visual comparison of different UIE methods on the UIEB dataset.

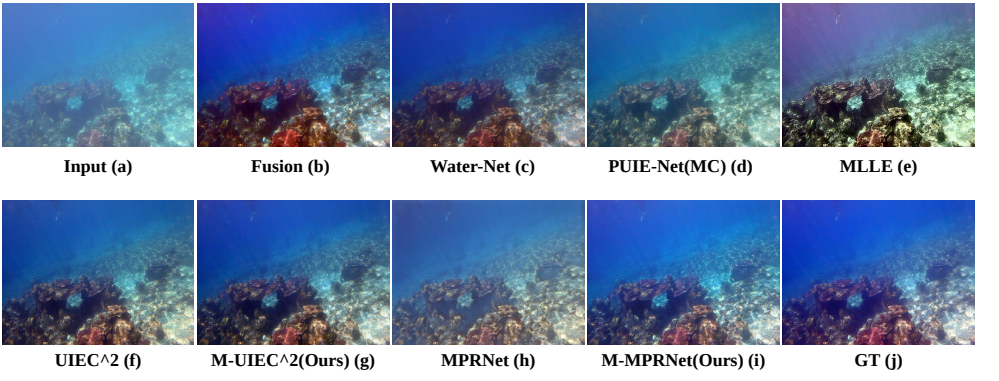


Figure 10: Visual comparison of different UIE methods on the UIEB dataset.

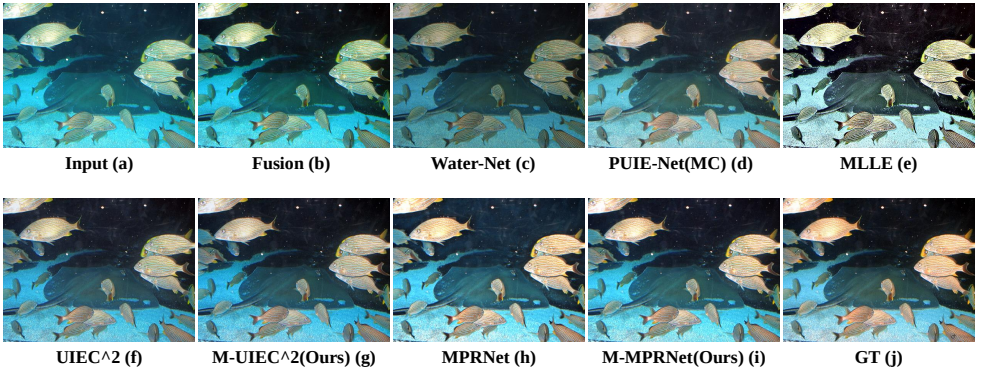


Figure 11: Visual comparison of different UIE methods on the UIEB dataset.