Supplementary Materials. FreqMamba: Viewing Mamba from a Frequency Perspective for Image Deraining

Anonymous Authors

This supplementary document is organized as follows:
Section 1 shows the details of the frequency dimension Mamba.
Section 2 shows a detailed ablation study on scanning methods.
Section 3 provides a comparison of GPU memory consumption between several methods.
Section 4 shows average error map comparison with or without degradation prior.
Section 5 shows more visual results.

1 ALGORITHM

We provide specific details of frequency dimension Mamba in this section, including wavelet packet transform and the locality strategy in frequency dimension scanning.

We use a 2-layer wavelet packet transform to decompose the original features into 16 sub-bands, arranged in order from upper left to lower right (distinguished by the red line in Fig. 1). Since the vanilla 2D scan of the full image range spans different 1-order sub-bands, it does not exactly follow the frequency dimension. We adopt a local strategy similar to LocalMamba [3] to make the scanning completely follow the frequency dimension. We provide an overview of the operations in Algorithm 1.

Algorithm 1 Frequency Dimension Mamba

Input: X: (B, C, H, W)
Output: Y: (B, C, H, W)

1. $X_i : \left(B, C, \frac{H}{2}, \frac{W}{2}\right) \leftarrow WT(X), i \in \{1, 2, 3, 4\}$
2. **for** $i = 1$ to 4 **do**
3. $\quad X_{i,j} : \left(B, C, \frac{H}{4}, \frac{W}{4}\right) \leftarrow WT(X_i), j \in \{1, 2, 3, 4\}$ // Wavelet package transform.
4. **end for**
5. $X_{wpt} : (B, C, H, W) \leftarrow$ Arrange $(X_{i,j}), i, j \in \{1, 2, 3, 4\}$
6. $X_{sequence} : (B, C, L) \leftarrow$ Frequency2DScan (X_{wpt}) // Frequency dimension scan(see Fig.3 in the main body or Fig. 1).
7. $Y : (B, C, L) \leftarrow$ Mamba $(X_{sequence})$
8. $Y_{out} : (B, C, H, W) \leftarrow$ 2DReconstruct $(X_{sequence})$
9. **return** Y_{out}

2 ABLATION STUDY

In this section, we compare the effects of three scanning methods on wavelet features: horizontal two-direction scanning, horizontal and vertical four-direction scanning (Fig. 3 (a) in the main body), and local scanning strategy (Fig. 3 (b) in the main body). The result is shown in Tab. 1. The performance of four-direction scanning is better than that of two-direction scanning, but it also increases the amount of computation. Our strategy does not increase the computational complexity and improves the performance compared with four-direction scanning.

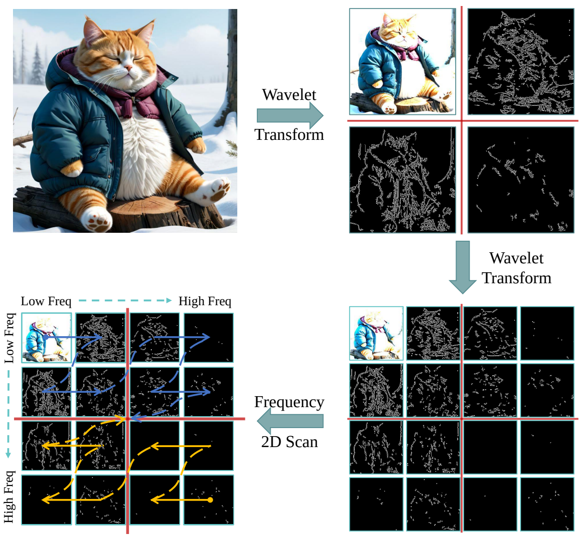

Figure 1: The illustration of frequency dimension scanning. Firstly, a 2-layer wavelet packet transform is used to decompose input features. Sub-bands are arranged as shown in the bottom-right of the figure. Lastly, we use a locality strategy to process the scanning.

Table 1: Ablation study of the effects of different scanning strategies.

Scanning method	PSNR ↑	SSIM ↑
two-direction	39.11	0.9796
four-direction	39.13	0.9801
local scanning	39.18	0.9814

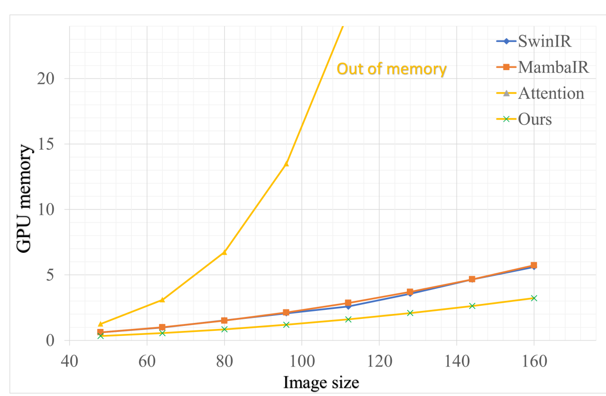

Figure 2: Memory consumption of different methods. Our method has minimal consumption and maintains linear growth.

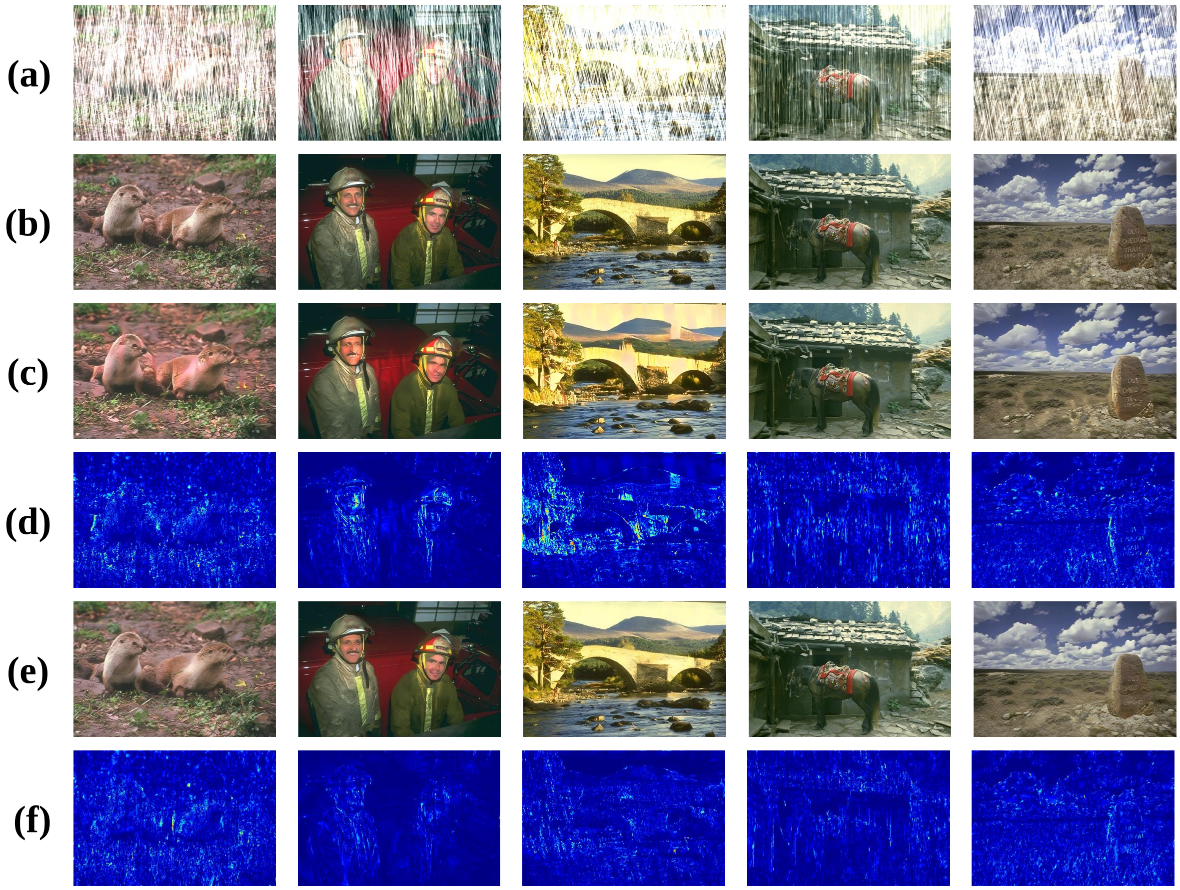

Figure 3: Comparison of more error maps with and without degradation priors. From top to bottom, there are (a) Inputs, (b) GT, (c) Outputs w/o prior, (d) Error maps w/o prior, (e) Outputs with prior, (f) Error maps with prior.

3 COMPARISON

In this section, we compare our method with MambaIR [2], SwinIR [4], and Attention [5] mechanism in terms of GPU memory. As image resolution increases, the memory consumption of the attention mechanism increases rapidly, while the memory consumption of our method still increases linearly and is lower than that of SwinIR and MambaIR.

4 ERROR MAP COMPARISON

Obtaining degraded priors using Mamba's input-dependent properties is one of our core efforts, and we show more examples in Fig. 3 to illustrate its importance. Especially when the degradation is pretty severe, the error reduction provided by the degradation prior is obvious. Further, we calculate the average error map of 100 images in the Rain100H test set as shown in Fig. 4.

5 MORE VISUALIZATION RESULTS

In this section, we provide more visual comparisons in Fig. 5, Fig. 6, and Fig. 7 to validate the effectiveness of the proposed method.

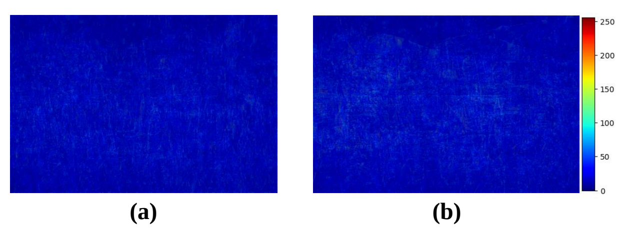

Figure 4: Average error maps of 100 images in the Rain100H test set. (a) Average error map with degradation prior and (b) Average error map without degradation prior.

REFERENCES

[1] Codruta O Ancuti, Cosmin Ancuti, Mateu Sbert, and Radu Timofte. 2019. Dense-haze: A benchmark for image dehazing with dense-haze and haze-free images. In *2019 IEEE international conference on image processing (ICIP)*. IEEE, 1014–1018.

[2] Hang Guo, Jinmin Li, Tao Dai, Zhihao Ouyang, Xudong Ren, and Shu-Tao Xia. 2024. MambaIR: A Simple Baseline for Image Restoration with State-Space Model. *arXiv preprint arXiv:2402.15648* (2024).

(a) Input (b) PreNet (c) Restormer (d) Ours (e) GT

Figure 5: Visual comparison of image deraining results on Rain100H [7] dataset.

(a) Input (b) MSBDN (c) AECR-Net (d) Ours (e) GT

Figure 6: Visual comparison of image dehazing results on DenseHaze [1] dataset.

[3] Tao Huang, Xiaohuan Pei, Shan You, Fei Wang, Chen Qian, and Chang Xu. 2024. LocalMamba: Visual State Space Model with Windowed Selective Scan. *arXiv preprint arXiv:2403.09338* (2024).

[4] Jingyun Liang, Jiezhang Cao, Guolei Sun, Kai Zhang, Luc Van Gool, and Radu Timofte. 2021. Swinir: Image restoration using swin transformer. In *ICCV*.

[5] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*. 5998–6008.

[6] Chen Wei, Wenjing Wang, Wenhan Yang, and Jiaying Liu. 2018. Deep retinex decomposition for low-light enhancement. *arXiv preprint arXiv:1808.04560* (2018).

[7] Wenhan Yang, Robby T Tan, Jiashi Feng, Jiaying Liu, Zongming Guo, and Shuicheng Yan. 2017. Deep joint rain detection and removal from a single image. In *CVPR*.

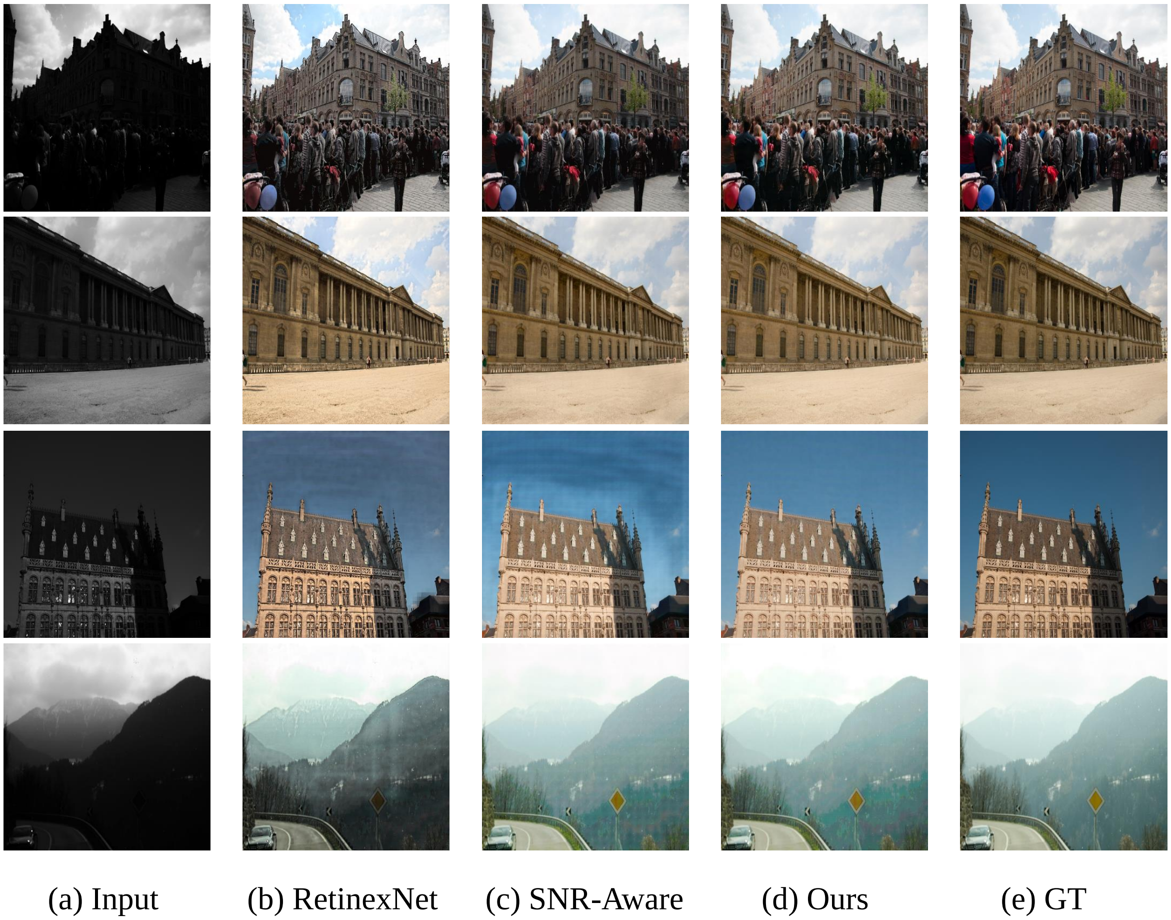

Figure 7: Visual comparison of low-light image enhancement results on LOL-V2-Syn [6] dataset.