Supplementary Material for "Multi-Scale Gradient-Guided Unrolling Architecture with Adaptive Mamba for Compressive Sensing"

Le Yang, Hongping Gan[*]
School of Software, Northwestern Polytechnical University, Xi'an, China
`yangle001@mail.nwpu.edu.cn; ganhongping@nwpu.edu.cn`

1. Overview

In this Supplementary Material, we provide more details and comparisons, including detailed experiments settings, performance under noise, more complexity analysis and ablation studies on learnable sampling matrix.

2. More Experiments

2.1. Experimental Settings

We utilize 400 images from the BSD400 dataset [1] as training data seeds. Through random horizontal and vertical flips followed by random cropping, we extract 102,400 patches of 128×128 pixels from these 400 images to form our actual training dataset. The Set11 dataset [3] is employed as the validation set. For optimization, we utilize the AdamW optimizer [4] with an initial learning rate of 1e-4, which is multiplied by 0.1 at epochs 30, 100, and 175 during the 200 training epochs, ultimately decreasing to 1e-7.

In our experiments (excluding ablation studies), the implemented MambaCS consists of 8 reconstruction stages, manifesting as 4 feature scales in the U-shaped structure with 32, 64, 128, and 256 channels, respectively. The sampling convolution kernel size is set equal to the training patch size, and the training batch size is set to 16. MambaCS's performance is evaluated on multiple public benchmark datasets including General100 [2], LIVE29 [6], OST300 [7], Set14 [8], and BSD68 [5], using Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index Measure (SSIM) as evaluation metrics. We download the execution code of other CS methods from their respective websites and execute them with their default settings to obtain comparative results. Our method is implemented in Python using the PyTorch 2.1.1 framework. All experiments are conducted on an Intel Xeon Silver 4216 CPU and GeForce RTX 3090 GPU with 24 GB RAM.

2.2. Performance under Noise

To evaluate the performance of MambaCS in noise-contaminated environments, we introduce various levels of Gaussian noise to images from the LIVE29 dataset. Subsequently, we perform reconstruction from these noise-corrupted images and conduct both quantitative comparisons and visual assessments against clean reference images. We employ Gaussian noise with four standard deviation levels: $\delta \in \{0.001, 0.002, 0.004, 0.008\}$.

Table 1. Comparison of PSNR (dB) and SSIM for different methods under varying Gaussian noise levels $\delta \in \{0, 0.001, 0.002, 0.004, 0.008\}$ at CS rates $\tau \in \{0.04, 0.10\}$.

noise	cs_ratio	TransCS		OCTUF		NesTD-Net		UFC-Net		CPP-Net		MambaCS	
		PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
0	0.04	24.87	0.6942	25.26	0.7113	25.30	0.7140	24.98	0.6995	25.52	0.7211	**25.94**	**0.7257**
	0.10	27.65	0.8195	28.23	0.8345	28.26	0.8373	27.88	0.8254	28.46	0.8386	**28.93**	**0.8400**
0.001	0.04	24.76	0.6840	25.12	0.7007	25.14	0.7021	24.85	0.6876	25.32	0.7068	**25.69**	**0.7123**
	0.10	27.19	0.7878	27.61	0.7985	27.60	0.8000	27.28	0.7884	27.74	0.8032	**28.04**	**0.8036**
0.002	0.04	24.67	0.6747	25.00	0.6903	25.01	0.6911	24.73	0.6769	25.16	0.6945	**25.49**	**0.7002**
	0.10	26.80	0.7600	27.14	0.7687	27.13	0.7689	26.83	0.7571	27.23	0.7703	**27.44**	**0.7719**
0.004	0.04	24.48	0.6579	24.78	0.6716	24.78	0.6711	24.52	0.6568	24.90	0.6726	**25.16**	**0.6777**
	0.10	26.15	0.7134	26.43	0.7201	26.40	0.7191	26.10	0.7064	26.42	0.7165	**26.56**	**0.7190**
0.008	0.04	24.15	0.6288	24.40	0.6393	24.38	0.6373	24.16	0.6233	24.47	0.6366	**24.65**	**0.6408**
	0.10	25.14	0.6416	25.37	0.6482	25.33	0.6463	25.02	0.6317	25.29	0.6395	**25.36**	**0.6424**

The performance of MambaCS is evaluated using average PSNR and SSIM metrics across different noise levels, alongside comparative results from other SOTA methods. As shown in Table 1, MambaCS outperforms other methods in both PSNR and SSIM metrics across all noise levels. Fig. 1 presents a visual comparison of reconstructed images between MambaCS and other methods. It is evident that MambaCS maintains stable and refined reconstruction quality even under higher noise levels.

2.3. More complexity analysis

Table 2 compares inference times, GPU memory usage and sampling time between MambaCS and mainstream DUNs at CS rate $\tau = 0.10$. Inference times are averaged from 100 runs on a single-channel 256×256 image, while sampling times are averaged from 100 operations across 32 single-channel 256×256 images. We also compare complexity and performance with ST-Net (a MambaCS variant built on

---

[*] Corresponding author

Figure 1. Visual comparisons of reconstructed image by MambaCS and other methods under Gaussian noise $\delta = 0.008$ at CS rates $\tau \in \{0.10, 0.04\}$.

Swin Transformer) for comprehensive evaluation between models of similarly scaled.

Table 2. Complexity comparisons

Method	Params. (M)	FLOPs (G)	Inference times (s)	Inference memory (MB)	Sample times (ms)	PSNR (dB)
LTwIST	23.28	158.90	0.31346	552	29.0850	27.42
NesTD-Net	5.36	372.58	0.23674	6140	2.5257	27.73
OCTUF	0.4	189.30	0.18746	1273	54.6924	27.77
CPP-Net	16.9	166.93	0.19615	2234	87.7457	27.93
ST-Net	45.86	90.15	0.56208	736	9.4815	27.89
MambaCS	44.91	**87.74**	**0.16948**	**546**	9.8591	**28.43**

2.4. Ablation Study on Learnable Sampling Matrix

For reference, we provide comparative experiments using a fixed Gaussian sampling matrix with $\tau = 0.10$ on LIVE29 in Tab. 3. The results demonstrate that our MambaCS can achieve optimal performance even without using learnable sampling matrix.

Table 3. Comparison using fixed sampling matrix

Method	LTwIST	NesTD-Net	OCTUF	UFC-Net	CPP-Net	MambaCS
PSNR	25.32	25.58	25.35	25.47	25.86	**26.03**

References

[1] Pablo Arbeláez, Michael Maire, Charless Fowlkes, and Jitendra Malik. Contour detection and hierarchical image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(5):898–916, 2010. 1

[2] Chao Dong, Chen Change Loy, and Xiaoou Tang. Accelerating the super-resolution convolutional neural network. In *European Conference on Computer Vision (ECCV)*, pages 391–407. Springer, 2016. 1

[3] Kuldeep Kulkarni, Suhas Lohit, Pavan Turaga, Ronan Kerviche, and Amit Ashok. Reconnet: Non-iterative reconstruction of images from compressively sensed measurements. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 449–458, 2016. 1

[4] Ilya Loshchilov and Frank Hutter. Fixing weight decay regularization in adam. *arXiv:1711.05101*, 2017. 1

[5] David Martin, Charless Fowlkes, Doron Tal, and Jitendra Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 416–423. IEEE, 2001. 1

[6] Hamid R Sheikh, Muhammad F Sabir, and Alan C Bovik. A statistical evaluation of recent full reference image quality assessment algorithms. *IEEE Transactions on Image Processing*, 15(11):3440–3451, 2006. 1

[7] Xintao Wang, Ke Yu, Chao Dong, and Chen Change Loy. Recovering realistic texture in image super-resolution by deep spatial feature transform. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 606–615, 2018. 1

[8] Roman Zeyde, Michael Elad, and Matan Protter. On single image scale-up using sparse-representations. In *Curves and Surfaces*, pages 711–730. Springer, 2012. 1