Supplementary Material for *"Latent Diffusion Prior Enhanced Deep Unfolding for Snapshot Spectral Compressive Imaging"*

Zongliang Wu[1,2*], Ruiying Lu[3*], Ying Fu[4], and Xin Yuan[2(✉)]

[1] Zhejiang University, Hangzhou, China
[2] School of Engineering, Westlake University, Hangzhou, China.
{wuzongliang,xyuan}@westlake.edu.cn
[3] School of Cyber Engineering, Xidian University, Xi'an, China.
ruiyinglu_xidian@163.com
[4] School of Computer Science and Technology, Beijing Institute of Technology, Beijing, China. fuying@bit.edu.cn

1 The Derivation Details in Deep Unfolding

Derivation of Eq. (5) : Follow the Eq. (4) in the paper, the GAP alternatively solving two sub-problems. The sub-problem of x can be written as

$$\boldsymbol{x}^{k+1} = \arg\min_{\boldsymbol{x}} \frac{1}{2}\|\boldsymbol{x} - \boldsymbol{v}^k\|_2^2 + \frac{\lambda}{2}\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|_2^2, \tag{S1}$$

where λ is a penalty factor. Then get the Lagrangian function of Eq. (S1) is

$$L(\boldsymbol{x}, \lambda) = \frac{1}{2}||\boldsymbol{x} - \boldsymbol{v}^k||_2^2 + \lambda(\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}). \tag{S2}$$

Then the optimal conditions of Eq. (S2) are

$$\frac{\partial}{\partial \boldsymbol{x}} L(\boldsymbol{x}, \lambda) = (\boldsymbol{x} - \boldsymbol{v}^k) + \boldsymbol{A}^\top \lambda = 0, \tag{S3}$$

$$\frac{\partial}{\partial \lambda} L(\boldsymbol{x}, \lambda) = \boldsymbol{A}\boldsymbol{x} - \boldsymbol{y} = 0. \tag{S4}$$

According to Eq. (S3) and (S4), we have

$$\lambda = -(\boldsymbol{A}\boldsymbol{A}^\top)^{-1}(\boldsymbol{y} - \boldsymbol{A}\boldsymbol{v}^k). \tag{S5}$$

Replace the Eq. (S5) in the Eq. (S4)

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{v}^k + \boldsymbol{A}^\top(\boldsymbol{A}\boldsymbol{A}^\top)^{-1}(\boldsymbol{y} - \boldsymbol{A}\boldsymbol{v}^k). \tag{S6}$$

The Eq. (S2) is the Eq. (5) in the paper.

*Z. Wu and R. Lu—Contribute equally

Derivation of Eq. (6) : The v sub-problem can be written as

$$\boldsymbol{v}^{k+1} = \arg\min_{\boldsymbol{v}} \tau R(\boldsymbol{v}) + \frac{\lambda}{2} \left\| \boldsymbol{v} - \boldsymbol{x}^{k+1} \right\|_2^2. \tag{S7}$$

In a deep unfolding framework, the constraint $R(\boldsymbol{v})$ is learned rather than handcrafted. Thus $R(\boldsymbol{v})$ is just implicitly expressed in a neural network, *i.e.*, a 'denoiser' $\mathcal{D}$ in deep unfolding. The update of $\boldsymbol{v}$ is:

$$\boldsymbol{v}^{(k+1)} = \mathcal{D}_{k+1}(\boldsymbol{x}^{(k+1)}). \tag{S8}$$

With the prior $\boldsymbol{z}$ introduced, we can get the Eq. (6) in the paper.

2 More Network Details

Aggregation Module The Fig. S1 shows the aggregation operation in Fig. 3 (a) of the paper Trident Transformer (TT). It includes a dual-step concatenation and convolution to aggregate three types of flow features. The output feature size is the same as the Trident Transformer input.

Unfolding Denoiser Structure The denoiser design follows the U-shape backbone structure of RLDUF [2], including stage interaction and block interaction, but replaces the key encoder/decoder part by our Trident Transformer. Following RLDUF [2], the 5-, 9-, and 10-stage unfolding network also uses the 'share parameter' strategy to share the 2 to $N-1$ stage parameter in the unfolding. Thus the total parameters for more stage unfolding are not increased. However, in the ablation study of LDM guidance location in Table S1, we observe that when applying the 'share parameter' strategy for training with more than three stages, the network performance is suppressed upon incorporating LDM guidance in every stage. This observation suggests that LDM guidance may interfere with shared parameters, leading to a suboptimal local solution. Consequently, we only use the prior and TT in the last stage for 5-, 9-, and 10-stage unfolding. This reduces the total parameters for 5-, 9-, and 10-stage unfolding, making them fewer than the 3-stage's.

Table S1: Ablation study of the location of using LDM guidance in 3- and 9-stage unfolding. 'Last stage' denotes using LDM prior only in the last DUN stage.

Stage	CPF position	PSNR (dB)	Params (M)	FLOPs (G)
3	Last stage	38.21	2.78	31.84
3	All stages	38.40	3.01	33.80
9	Last stage	40.09	2.78	88.68
9	All stages	38.74	3.01	97.71

Table S2: The PSNR (dB) comparisons of using LDM prior in different model sizes.

Method	Stage		
	3	5	9
Full Model	38.40	39.49	40.09
w/o LDM	38.11	39.17	39.66
Difference	+0.29	+0.32	+0.43

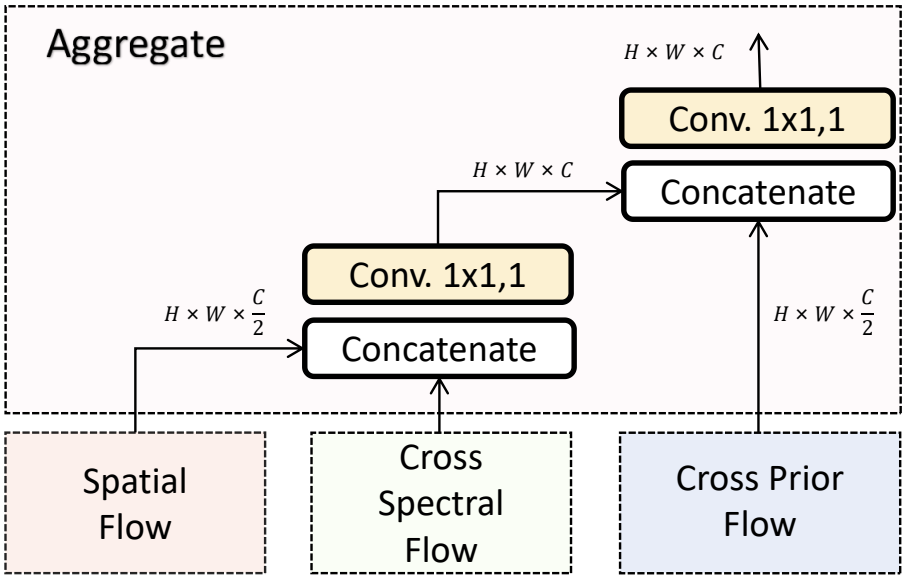

Fig. S1: The aggregation operation in the Trident Transformer.

3 More Ablation Studies

Impact of LDM Prior To assess the influence of the LDM prior, we conducted experiments where both the LDM (LE, LE′, and the DM) and the 'CPF' module, which is used to fuse priors in each DUN stage within the Trident Transformer, were removed. The results in Table S2 illustrate that as the number of stages in the DUN increases, the significance of the LDM prior becomes increasingly obvious. This observation can indicate the LDM's growing contribution to enhancing DUN to break the limitations of a regression model.

Updating the Diffusion Model We compare the difference of training diffusion model (DM) in Table S3. The 'Separate' training approach involves updating DM by only predicting noise according to the Eq. (12) in the paper instead of jointly training with DUN. In contrast, the 'Joint' approach represents the standard configuration, where $\mathcal{L}_{\text{diff}} = \|\hat{\boldsymbol{z}} - \boldsymbol{z}\|_1$ as described in the paper. The result illustrates that the separate training detracts from the effective integration of LDM priors with the DUN. This may be because of denoising the network's limitations, leading to discrepancies between the generated $\hat{\boldsymbol{z}}$ and the target $\boldsymbol{z}$. However, joint training enables the DUN to adapt and accommodate these

DM Train	PSNR (dB)
Separate	37.81
Joint	38.40

Table S3: DM training strategy comparison.

variances.

Deep Unfolding Framework We compare other deep unfolding frameworks used in recent SOTA methods like HQS (used in DAHUST [1] and PADUT [4]) and PGD (RDLUF [2]) do not show better performance than GAP according to Tab. S4 where we replaced GC-GAP in our paper with HQS, PGD, and normal GAP in our 3-stage DUN for ablation study. We conjecture that it is the robustness of GAP [7] that enables it to demonstrate more stable performance than other methods under the influences of additional LDM priors and a two-stage training process.

Method	w HQS	w PGD	w GAP
PSNR	37.87	37.90	38.01

Table S4: Unfolding framework comparison in our 3-stage network.

4 More Real Data Results

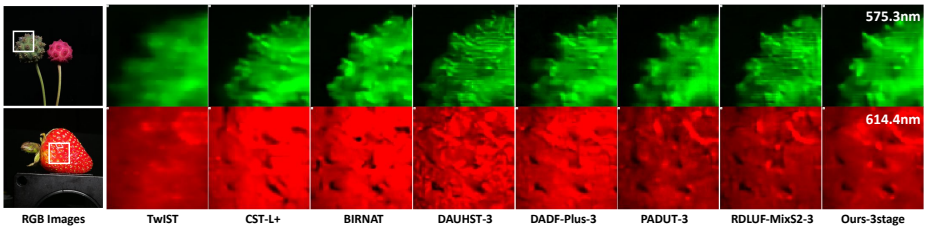

Fig. S2: The comparisons of the remaining 2 real data reconstruction results.

Table S5: Numerical comparisons of average 5 real scene measurement reconstruction results by no-reference IQA. The traditional NIQE [6] and Transformer-based MUSIQ [3] are selected for companies.

Metric	TwIST	DESCI	CST-L-Plus	BIRNAT	DAUHST-3	PADUT-3	DADF	RDLUF-3	Ours-3
NIQE ↓	11.64	11.27	11.68	11.23	10.64	10.27	10.03	9.83	**9.75**
MUSIQ ↑	3.82	3.99	4.04	4.21	4.26	4.10	4.25	4.27	**4.31**

We visually compare the remaining 4 out of 5 real-world data in Fig. S2. The visual results show that our method has better reconstruction quality in real data. The zoomed part Fig. S2 indicates our method not only preserves essential texture details but removes more artefacts compared with other methods.

The real data does not have ground truth to compare, but for a more convincing conclusion, we further calculate[1] no-reference image quality assessment

[1] We apply an open source IQA toolbox for calculation: https://github.com/chaofengc/IQA-PyTorch.

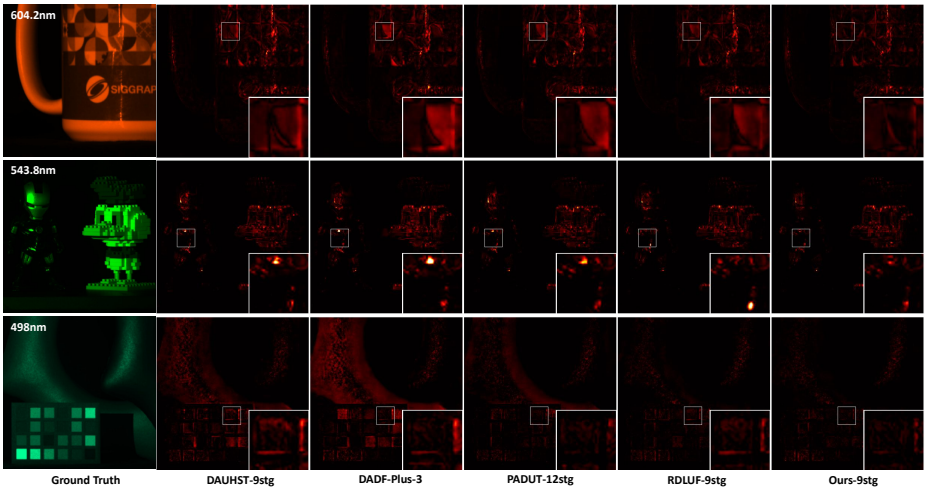

Fig. S3: The error maps of the other 3 synthetic data reconstruction results.

(NR-IQA) metrics. For our computational imaging task, we choose widely-used NR-IQA metrics in low-level vision tasks including the traditional NIQE [6] and the Transformer-based MUSIQ [3] in Table. S5. In the average NIQE and MUSIQ scores of 5 scenes, our method still gains the best score which can support our superiority in visual comparisons.

For calculation details, because the existing IQA tool only supports RGB image assessment, following the previous work [5], we converted the output HSIs to synthetic-RGB (sRGB) images via the CIE (International Commission on Illumination) color-matching function for NIQE and MUSIQ calculation. The conversation approximates a weighted average process of the intensity of different wavelengths, and thus will not influence the spatial quality of images.

5 More Synthetic Data Results

Given that synthetic data is relatively difficult to distinguish visual differences, we visualize error maps of each result in Table S3. The brighter parts denote larger errors compared to the ground truth. The zoomed parts show our method has fewer errors in reconstructing edges and textures.

6 More Implementation Details

In the paper, Table 2, the inference time and training time of all models are calculated on a single Nvidia RTX3090 GPU. The training batch size is set to 1, total epoch number is 300 with 5000 samples for each. The initial learning rate is 4×10^{-4}. Other training of our model also follows these settings.

The test batch size is 10 (*i.e.*, the output size is $10 \times 28 \times 256 \times 256$). The test process includes one iteration warm-up and 10 iteration time accumulation. The final inference time is the average of 10 iterations. We also test 50 and 100 iteration average times, they are the same as the 10 iteration average, thus 10 iterations are enough for comparison.

For FLOPs (Floating Point Operations) and network parameters calculation, we apply the code from the previous reconstruction method [2][2]. The input is a 256×256 measurement, and the output size is $28 \times 256 \times 256$.

For a fair comparison, all models in our paper calculate the FLOPs during inference shown in Tab. S6. Note that the latent encoder (LE) in our pre-trained part is excluded since it is only used to encode ground truth. The excluded part only needs 2.41G FLOPs, our method still provides fewer FLOPs than SOTA even if this part is included.

Table S6: The FLOPs of each module in our 3-stage model.

Module	Infer. Phase Model	LE (Excluded)	LE'	Diffusion	DUN
FLOPs (G)	33.80	2.41	2.41	0.10	31.70

References

1. Cai, Y., Lin, J., Wang, H., Yuan, X., Ding, H., Zhang, Y., Timofte, R., Gool, L.V.: Degradation-aware unfolding half-shuffle transformer for spectral compressive imaging. NeurIPS pp. 37749–37761 (2022)
2. Dong, Y., Gao, D., Qiu, T., Li, Y., Yang, M., Shi, G.: Residual degradation learning unfolding framework with mixing priors across spectral and spatial for compressive spectral imaging. In: CVPR. pp. 22262–22271 (2023)
3. Ke, J., Wang, Q., Wang, Y., Milanfar, P., Yang, F.: Musiq: Multi-scale image quality transformer. In: ICCV. pp. 5148–5157 (2021)
4. Li, M., Fu, Y., Liu, J., Zhang, Y.: Pixel adaptive deep unfolding transformer for hyperspectral image reconstruction. In: ICCV. pp. 12959–12968 (2023)
5. Meng, Z., Yuan, X.: Perception inspired deep neural networks for spectral snapshot compressive imaging. In: 2021 ICIP (ICIP). pp. 2813–2817. IEEE (2021)
6. Mittal, A., Soundararajan, R., Bovik, A.C.: Making a "completely blind" image quality analyzer. IEEE Signal processing letters **20**(3), 209–212 (2012)
7. Yuan, X.: Generalized alternating projection based total variation minimization for compressive sensing. In: ICIP. pp. 2539–2543 (2016)

[2] `https://github.com/ShawnDong98/RDLUF_MixS2/tree/master/simulation/train_code/ptflops`