Gaussian Splatting-based Low-rank Tensor Representation for Multi-Dimensional Image Recovery

Supplementary Material

1. Proof of Lemma 1

Proof. ***1). 2D Gaussian splatting can explicitly generate arbitrary latent tensor*** $\mathcal{A}$ when the parameter $\theta_{\mathcal{A}}$ satisfies:

(i) $N = HW$, $\boldsymbol{\mu}_j = (x, y)^\top$ with $j = (x-1)W + y$ for $j = 1, \ldots, HW$;

(ii) $\boldsymbol{\Sigma}_j \to \mathbf{0}$.

For notational simplicity, we let all covariances $\boldsymbol{\Sigma}_j$ be equal to $\bar{\boldsymbol{\Sigma}}$. The latent tensor is given by:

$$\mathcal{A}(x', y') = \sum_{j=1}^{HW} \mathbf{c}_j \cdot \exp\left(-\frac{1}{2}\left(\begin{pmatrix} x' \\ y' \end{pmatrix} - \boldsymbol{\mu}_j\right)^\top \bar{\boldsymbol{\Sigma}}^{-1}\left(\begin{pmatrix} x' \\ y' \end{pmatrix} - \boldsymbol{\mu}_j\right)\right), \quad (1)$$

To represent the spatial correspondence between the position $\boldsymbol{\mu}_j$ and feature $\mathbf{c}_j$ of each 2D Gaussian primitive, we denote $\boldsymbol{\mu}_j = \boldsymbol{\mu}_h^w = (h, w)^\top$ and $\mathbf{c}_j = \mathbf{c}_h^w$, where $j = (h-1)W + w$. When $\bar{\boldsymbol{\Sigma}}$ approaches the zero matrix, Eq. (1) simplifies to:

$$\begin{aligned}
&\mathcal{A}(x', y') \\
&= \lim_{\bar{\boldsymbol{\Sigma}} \to \mathbf{0}} \sum_{j=1}^{HW} \mathbf{c}_j \cdot \exp\left(-\frac{1}{2}\left(\begin{pmatrix} x' \\ y' \end{pmatrix} - \boldsymbol{\mu}_j\right)^\top \bar{\boldsymbol{\Sigma}}^{-1}\left(\begin{pmatrix} x' \\ y' \end{pmatrix} - \boldsymbol{\mu}_j\right)\right) \\
&= \lim_{\bar{\boldsymbol{\Sigma}} \to \mathbf{0}} \sum_{h=1}^{H} \sum_{w=1}^{W} \mathbf{c}_h^w \cdot \exp\left(-\frac{1}{2}\left(\begin{pmatrix} x' \\ y' \end{pmatrix} - \boldsymbol{\mu}_h^w\right)^\top \bar{\boldsymbol{\Sigma}}^{-1}\left(\begin{pmatrix} x' \\ y' \end{pmatrix} - \boldsymbol{\mu}_h^w\right)\right) \\
&= \lim_{\bar{\boldsymbol{\Sigma}} \to \mathbf{0}} \mathbf{c}_1^1 \exp\left(-\frac{1}{2}\left(\begin{pmatrix} x' \\ y' \end{pmatrix} - \boldsymbol{\mu}_1^1\right)^\top \bar{\boldsymbol{\Sigma}}^{-1}\left(\begin{pmatrix} x' \\ y' \end{pmatrix} - \boldsymbol{\mu}_1^1\right)\right) + \cdots + \\
&\qquad \mathbf{c}_{x'}^{y'} \exp\left(-\frac{1}{2}\left(\begin{pmatrix} x' \\ y' \end{pmatrix} - \boldsymbol{\mu}_{x'}^{y'}\right)^\top \bar{\boldsymbol{\Sigma}}^{-1}\left(\begin{pmatrix} x' \\ y' \end{pmatrix} - \boldsymbol{\mu}_{x'}^{y'}\right)\right) + \cdots + \\
&\qquad \mathbf{c}_H^W \exp\left(-\frac{1}{2}\left(\begin{pmatrix} x' \\ y' \end{pmatrix} - \boldsymbol{\mu}_H^W\right)^\top \bar{\boldsymbol{\Sigma}}^{-1}\left(\begin{pmatrix} x' \\ y' \end{pmatrix} - \boldsymbol{\mu}_H^W\right)\right) \\
&= 0 + \cdots + \mathbf{c}_{x'}^{y'} + \cdots + 0 \\
&= \mathbf{c}_{x'}^{y'}.
\end{aligned} \quad (2)$$

Finally, we obtain:

$$\mathcal{A}(x', y') = \mathbf{c}_{x'}^{y'} = \mathbf{c}_j, \quad \text{where } j = (x'-1)W + y'. \quad (3)$$

Here, $\mathbf{c}_j$ is a learnable attribute of the 2D Gaussian primitive, and the 2D Gaussian splatting can explicitly generate an arbitrary latent tensor $\mathcal{A}$.

2). 1D Gaussian splatting can explicitly generate arbitrary transform matrix $\mathbf{T}$ when the parameter $\theta_{\mathbf{T}}$ satisfies:

(i) $K = B$, $\mu_k^r = k$ for $k = 1, \ldots, B$, $r = 1, \ldots, R$.

(ii) $\sigma_k^r \to 0^+$.

For notational simplicity, we let all variances σ_k^r be equal to $\bar{\sigma}$. The transform matrix is given by:

$$\mathbf{T}(z, r) = \sum_{k=1}^{B} c_k^r \cdot \exp\left(-\frac{(z-k)^2}{2\bar{\sigma}^2}\right). \quad (4)$$

As the variance $\bar{\sigma}$ approaches zero from the right, Eq. (4) simplifies to:

$$\begin{aligned}
\mathbf{T}(z, r) &= \lim_{\bar{\sigma} \to 0^+} \sum_{k=1}^{B} c_k^r \cdot \exp\left(-\frac{(z-k)^2}{2\bar{\sigma}^2}\right) \\
&= \lim_{\bar{\sigma} \to 0^+} c_1^r \cdot \exp\left(-\frac{(z-1)^2}{2\bar{\sigma}^2}\right) + \ldots \\
&\qquad + c_z^r \cdot \exp\left(-\frac{(z-z)^2}{2\bar{\sigma}^2}\right) + \ldots \\
&\qquad + c_B^r \cdot \exp\left(-\frac{(z-B)^2}{2\bar{\sigma}^2}\right) \\
&= 0 + \cdots + c_z^r + \cdots + 0 \\
&= c_z^r.
\end{aligned} \quad (5)$$

Since c_z^r is a learnable attribute of the 1D Gaussian primitive, the 1D Gaussian splatting can explicitly generate an arbitrary transform matrix $\mathbf{T}$.

2. Proof of Theorem 1

Proof. Under the conditions of Lemma 1, the 2D and 1D Gaussian splatting can represent arbitrary latent tensor $\mathcal{A}$ and arbitrary transform matrix $\mathbf{T}$. Let $\mathcal{A}_{[r]} = \mathbf{U}_{[r]}\, \mathbf{S}_{[r]}\, \mathbf{V}_{[r]}^{\mathrm{H}}$ with unitary matrices $\mathbf{U}_{[r]}, \mathbf{V}_{[r]}$ and diagonal matrix $\mathbf{S}_{[r]}$, and $\mathbf{T}$ is any invertible transform matrix, then the proposed GSLR degrades to the classical t-SVD method.

3. The Impact of Hyperparameters

The proposed GSLR includes four hyperparameters: the numbers of 2D and 1D Gaussian splatting primitives (i.e., N and K), the trade-off parameter λ of the low-rank constraint, and the dimensionality R of the latent tensor.

To investigate the influence of these hyperparameters on multi-dimensional image recovery performance, we vary one parameter at a time while keeping the others fixed, and the corresponding PSNR results are shown in Fig. 1. The impact of the number of 2D and 1D Gaussian primitives (*i.e.*, N and K) for the proposed GSLR are shown in Fig. 1 (a) and (b). Both parameters exhibit a similar pattern that

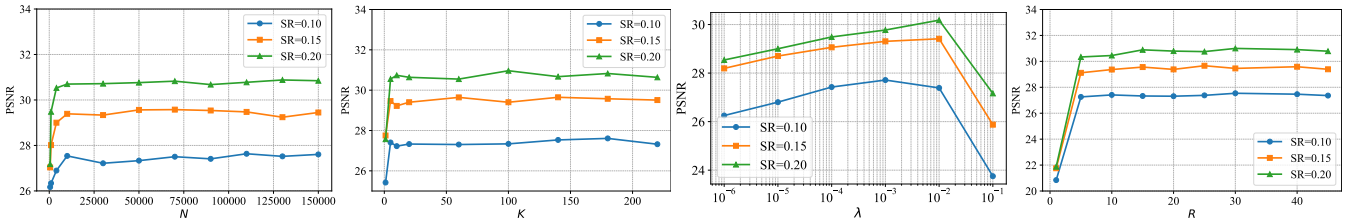

Figure 1. Quantitative performance with respect to different hyperparameter settings on *toy* dataset under tube missing. (a) the number of 2D Gaussian primitives N, (b) the number of 1D Gaussian primitives K, (c) the trade-off parameter λ, and (d) the dimensionality R of the latent tensor along the third dimension.

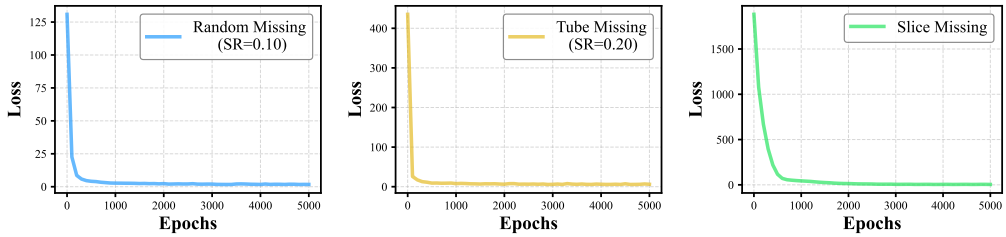

Figure 2. The loss function values versus iterations on *Toy* dataset under three missing scenarios.

the performance rapidly improves at small values and then stabilizes, indicating that GSLR is relatively robust to N and K once they reach a sufficient amounts. For the trade-off parameter λ, the PSNR improves as λ increases from a small value but begins to decline once λ excessively large as shown in Fig. 1 (c), which underscores the importance of selecting an appropriate value. For the dimensionality R, the PSNR results become relatively stable when R is sufficiently large as shown in Fig. 1 (d), suggesting that GSLR can achieve satisfactory performance across a broad range of values.

4. More Experimental Results

Experimental Setup. All experiments are conducted on a Linux platform equipped with an Intel Xeon E3-1231 v3 CPU, an NVIDIA RTX 2080 Ti GPU, CUDA 12.4, and PyTorch 2.8.0.

Quantitative and Qualitative Results. Quantitatively, as summarized in Tabs. 1 and 2, our method achieves state-of-the-art performance, consistently outperforming competing methods on both PSNR and SSIM metrics across all datasets. Qualitatively, the visual comparisons in Figs. 3 to 8 further validate our model's superiority. Our approach generates reconstructions that are visually more faithful to the GT, capturing the local high-frequency information and structures more accurately than others, even in challenging scenarios with very low sampling rates.

Numerical Convergence. To evaluate the numerical convergence of the proposed GSLR, we plot the loss function values versus iterations on *Toy* dataset under three missing scenarios in Fig. 2. The loss function values decrease rapidly in the initial iterations and then gradually stabilize, demonstrating the empirical convergence behavior.

5. Limitations and Future Work

The proposed GSLR enjoys a powerful representation capability for multi-dimensional images, though a few limitations remain that can be addressed in future work.

First, GSLR requires numerous Gaussian primitives to handle complex scenes. This enhances its ability to capture high-frequency information but also increases memory consumption. Future work could explore content-aware strategies to more accurately capture the high-frequency information while reducing memory usage. Second, we evaluated the representation capability of GSLR on multi-dimensional image recovery tasks in this paper. In future work, it could be extended to other multi-dimensional image tasks, such as multi-dimensional image compression, and even to higher-order data, such as videos and point clouds.

Table 1. The detailed quantitative results for random missing. The **best** and second-best values are highlighted.

Method		TNN		TRLRF		TCTV		HLRTF		LRTFR		FLRTF		GSLR	
Data	SR	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Color image	0.02	16.548	0.234	14.502	0.102	21.211	0.650	18.039	0.287	19.784	0.439	14.590	0.101	**23.621**	**0.736**
Plane	0.05	19.042	0.360	18.148	0.277	23.744	0.723	21.152	0.451	23.766	0.637	19.242	0.362	**26.030**	**0.809**
(512 × 512 × 3)	0.10	21.624	0.494	21.844	0.505	26.678	0.812	24.585	0.661	26.954	0.799	22.226	0.514	**28.304**	**0.869**
Color image	0.02	13.934	0.096	13.324	0.082	15.408	0.231	14.228	0.121	16.374	0.185	12.954	0.079	**17.793**	**0.286**
Baboon	0.05	15.970	0.161	15.268	0.125	18.202	0.368	16.737	0.202	18.635	0.290	16.376	0.163	**18.756**	**0.384**
(512 × 512 × 3)	0.10	17.440	0.257	17.722	0.268	20.190	0.515	18.688	0.365	19.569	0.398	18.154	0.287	**20.408**	**0.522**
Color image	0.02	15.749	0.212	14.660	0.114	18.773	0.486	16.299	0.232	19.092	0.423	14.039	0.104	**21.414**	**0.603**
House	0.05	18.018	0.333	18.047	0.306	21.651	0.628	19.077	0.374	21.570	0.571	18.390	0.354	**23.228**	**0.704**
(512 × 512 × 3)	0.10	20.217	0.461	20.553	0.473	24.284	0.756	21.963	0.568	23.721	0.697	20.479	0.488	**24.823**	**0.784**
Color image	0.02	16.443	0.232	15.212	0.116	20.403	0.535	17.784	0.292	21.531	0.474	15.081	0.127	**23.906**	**0.659**
Female	0.05	18.981	0.335	18.209	0.262	23.647	0.647	20.586	0.422	25.145	0.662	17.138	0.234	**26.299**	**0.753**
(256 × 256 × 3)	0.10	21.291	0.464	21.738	0.452	26.514	0.738	23.196	0.572	27.042	0.751	21.796	0.475	**28.156**	**0.815**
MSI	0.02	23.656	0.708	24.463	0.656	28.957	0.893	28.699	0.834	29.823	0.852	23.054	0.467	**31.712**	**0.923**
Toy	0.05	27.368	0.822	31.325	0.867	33.631	0.951	34.691	0.948	34.676	0.943	30.564	0.858	**37.148**	**0.983**
(256 × 256 × 31)	0.10	31.538	0.904	34.969	0.935	38.405	0.977	40.759	0.983	39.564	0.980	37.873	0.963	**43.630**	**0.995**
MSI	0.02	25.515	0.645	26.437	0.603	32.257	0.874	31.217	0.827	33.012	0.871	24.250	0.512	**34.247**	**0.913**
Flowers	0.05	28.853	0.757	31.779	0.804	36.287	0.937	36.971	0.940	37.717	0.945	32.137	0.823	**40.736**	**0.979**
(256 × 256 × 31)	0.10	32.385	0.853	36.062	0.902	40.436	0.971	41.934	0.977	41.656	0.978	39.403	0.950	**45.189**	**0.991**
MSI	0.02	18.092	0.277	17.941	0.279	23.353	0.700	20.649	0.487	23.288	0.669	17.580	0.301	**24.461**	**0.751**
Beads	0.05	20.473	0.485	21.300	0.513	26.688	0.839	25.683	0.785	26.780	0.817	21.187	0.554	**29.887**	**0.923**
(256 × 256 × 31)	0.10	23.636	0.689	27.272	0.796	30.282	0.920	31.364	0.930	30.294	0.893	27.492	0.809	**34.750**	**0.974**
MSI	0.02	23.250	0.629	24.007	0.527	29.370	0.876	28.518	0.818	29.488	0.846	20.056	0.342	**31.231**	**0.935**
Feathers	0.05	27.039	0.771	30.754	0.834	33.613	0.941	34.264	0.937	33.917	0.916	29.861	0.800	**36.882**	**0.979**
(256 × 256 × 31)	0.10	30.918	0.871	34.820	0.919	37.958	0.972	39.627	0.977	38.451	0.969	36.476	0.939	**41.524**	**0.989**
MSI	0.02	23.526	0.517	23.766	0.510	28.926	0.822	29.708	0.831	30.903	0.870	24.536	0.612	**32.208**	**0.917**
PaviaU	0.05	27.261	0.736	29.968	0.825	32.975	0.922	35.602	0.951	35.696	0.955	33.318	0.906	**37.010**	**0.971**
(256 × 256 × 80)	0.10	31.272	0.868	33.221	0.911	37.407	0.967	40.511	0.983	41.697	0.984	40.571	0.978	**42.234**	**0.989**

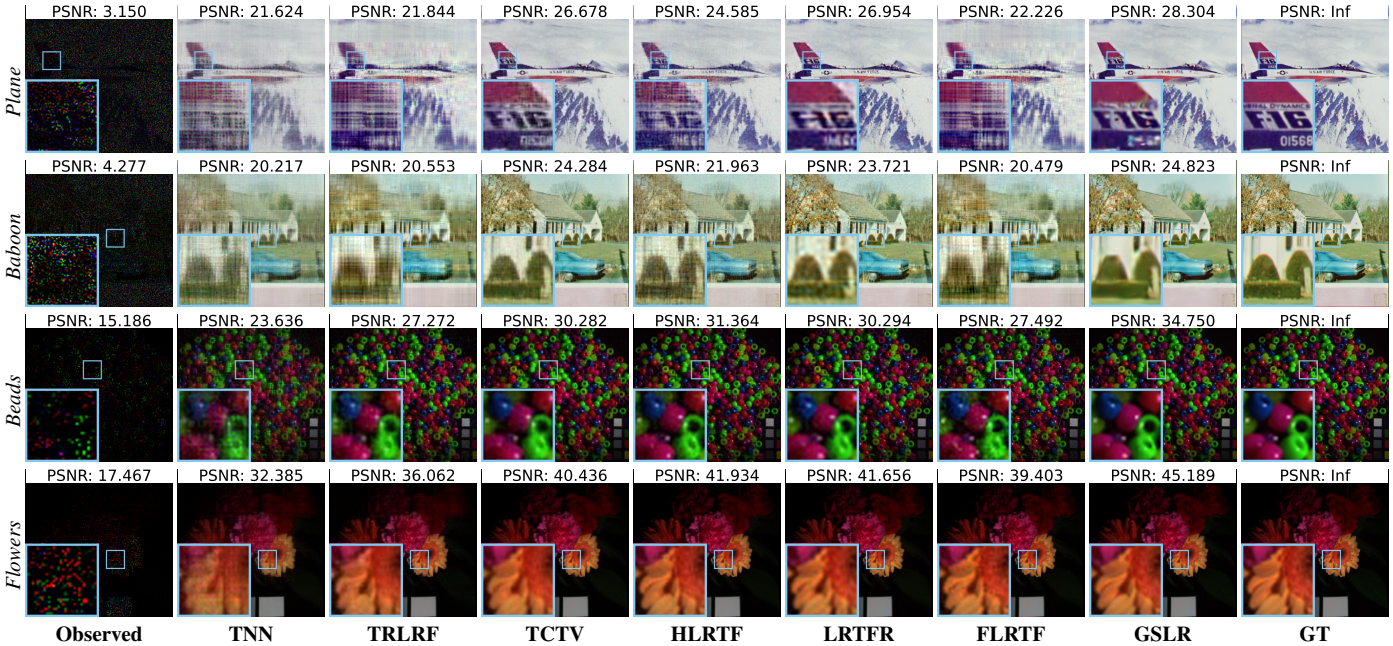

Figure 3. Reconstructed results and zoomed-in details by different methods under the random missing (SR = 0.10).

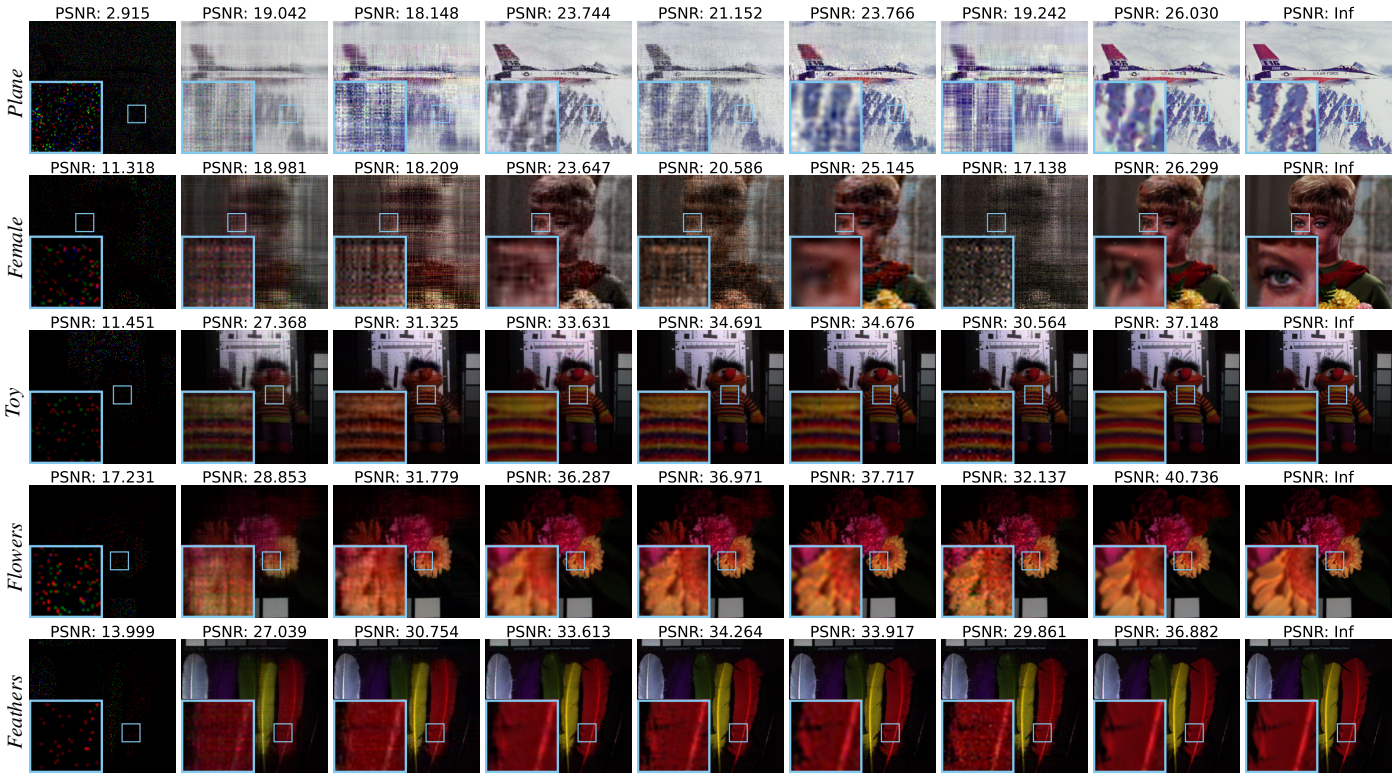

Figure 4. Reconstructed results and zoomed-in details by different methods under the random missing (SR = 0.05).

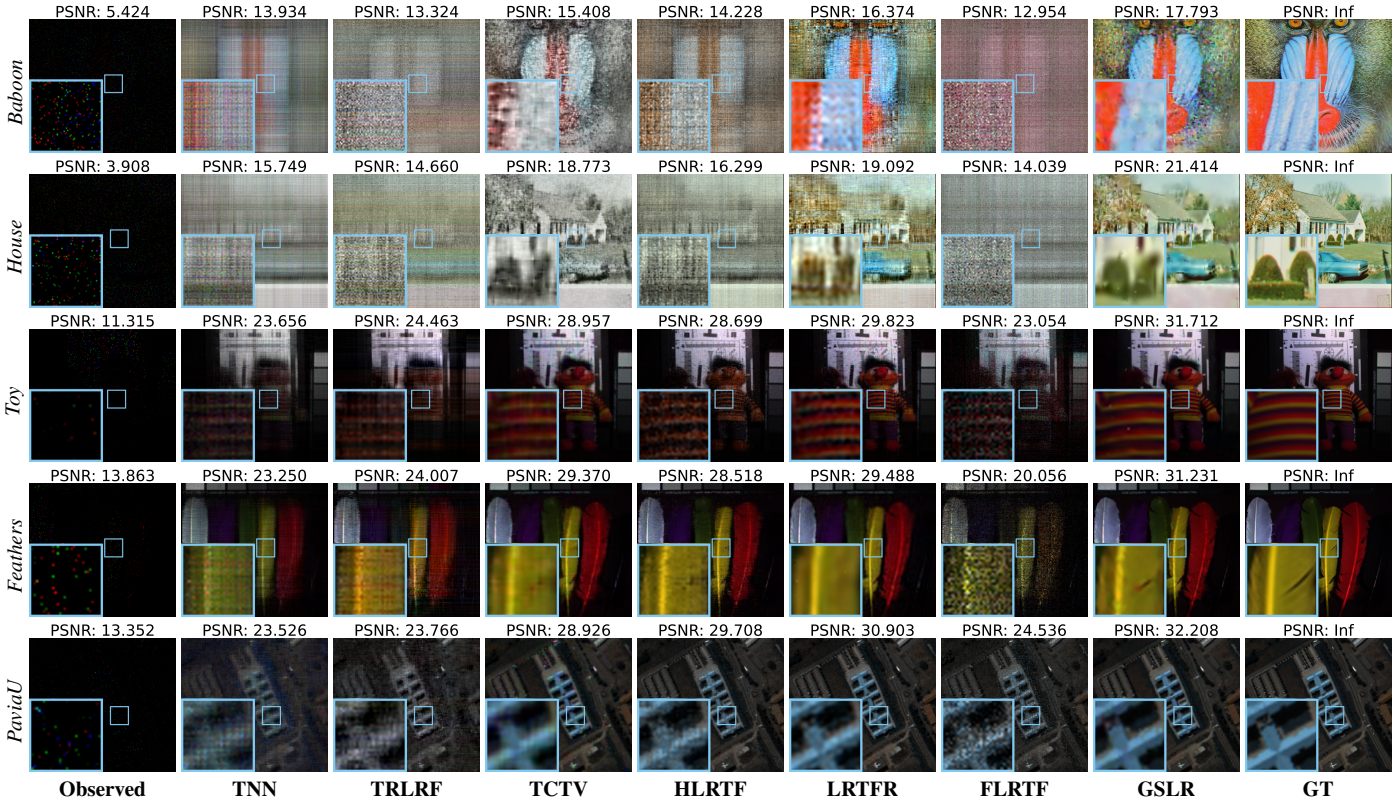

Figure 5. Reconstructed results and zoomed-in details by different methods under the random missing (SR = 0.02).

Table 2. The detailed quantitative results for tube missing. The **best** and second-best values are highlighted.

Method		TNN		TRLRF		TCTV		HLRTF		LRTFR		FLRTF		GSLR	
Data	SR	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Color image	0.10	19.439	0.360	16.785	0.185	25.564	0.772	21.598	0.496	23.164	0.616	17.354	0.259	**26.124**	**0.814**
Plane	0.15	21.092	0.453	19.679	0.399	26.909	0.817	23.366	0.553	25.271	0.747	20.364	0.453	**27.710**	**0.853**
(512 × 512 × 3)	0.20	22.505	0.535	21.854	0.521	27.875	0.848	24.941	0.640	27.158	0.825	21.532	0.499	**28.798**	**0.874**
Color image	0.10	17.045	0.219	16.124	0.164	19.470	0.432	17.915	0.268	18.734	0.331	16.526	0.203	**19.560**	**0.435**
Baboon	0.15	17.866	0.287	16.733	0.248	20.141	0.501	18.837	0.359	19.346	0.395	17.979	0.306	**20.209**	**0.507**
(512 × 512 × 3)	0.20	18.558	0.353	18.126	0.339	20.758	0.562	19.426	0.422	19.949	0.470	18.710	0.378	**20.785**	**0.563**
Color image	0.10	18.831	0.369	16.900	0.278	23.475	0.709	20.390	0.450	21.393	0.574	18.253	0.361	**23.531**	**0.710**
House	0.15	20.230	0.459	19.516	0.438	24.515	0.765	22.032	0.562	22.719	0.676	19.533	0.464	**24.567**	**0.771**
(512 × 512 × 3)	0.20	21.409	0.535	20.853	0.521	25.304	0.802	23.093	0.641	24.045	0.754	20.595	0.513	**25.459**	**0.827**
Color image	0.10	19.471	0.362	17.839	0.216	26.334	0.735	21.854	0.495	25.213	0.672	16.871	0.172	**26.781**	**0.778**
Female	0.15	21.370	0.455	20.580	0.349	27.602	0.777	23.704	0.603	26.864	0.749	19.508	0.382	**28.304**	**0.817**
(256 × 256 × 3)	0.20	22.828	0.534	22.610	0.501	28.483	0.807	25.216	0.653	28.161	0.794	22.549	0.512	**29.204**	**0.839**
MSI	0.10	20.787	0.592	15.324	0.093	27.146	0.857	23.105	0.740	23.748	0.658	17.327	0.323	**27.318**	**0.868**
Toy	0.15	22.657	0.660	18.184	0.407	29.066	0.895	25.477	0.803	26.312	0.746	18.947	0.500	**29.347**	**0.911**
(256 × 256 × 31)	0.20	24.516	0.735	20.262	0.536	30.155	0.918	27.592	0.860	27.438	0.825	21.718	0.439	**30.501**	**0.938**
MSI	0.10	24.863	0.604	20.147	0.173	31.501	0.863	28.111	0.776	28.828	0.714	20.899	0.305	**32.128**	**0.897**
Flowers	0.15	26.837	0.676	22.091	0.350	32.980	0.898	30.039	0.839	31.352	0.815	22.456	0.331	**34.178**	**0.933**
(256 × 256 × 31)	0.20	28.144	0.739	26.360	0.637	34.055	0.919	30.993	0.871	32.814	0.895	26.029	0.614	**34.930**	**0.943**
MSI	0.10	17.805	0.247	14.499	0.119	22.595	0.625	19.461	0.386	20.222	0.483	16.542	0.275	**22.964**	**0.707**
Beads	0.15	18.793	0.322	15.153	0.170	23.848	0.703	20.815	0.500	22.593	0.623	17.477	0.311	**24.334**	**0.773**
(256 × 256 × 31)	0.20	19.659	0.391	17.766	0.316	25.002	0.759	22.001	0.587	24.284	0.726	17.915	0.300	**25.602**	**0.820**
MSI	0.10	21.588	0.561	17.352	0.140	27.777	0.860	24.053	0.732	24.929	0.678	18.150	0.325	**28.547**	**0.911**
Feathers	0.15	23.445	0.655	20.945	0.505	29.266	0.892	25.871	0.783	27.058	0.780	19.327	0.289	**31.242**	**0.941**
(256 × 256 × 31)	0.20	24.879	0.722	21.310	0.362	30.735	0.918	27.443	0.840	28.540	0.819	22.765	0.603	**32.704**	**0.957**
MSI	0.10	20.623	0.296	17.160	0.112	25.616	0.664	21.950	0.407	23.153	0.489	19.555	0.359	**25.836**	**0.723**
PaviaU	0.15	21.535	0.365	17.759	0.176	26.751	0.727	23.167	0.503	25.585	0.682	20.614	0.412	**27.220**	**0.788**
(256 × 256 × 80)	0.20	22.613	0.440	20.659	0.330	27.820	0.774	24.415	0.579	27.248	0.753	21.179	0.437	**28.452**	**0.831**

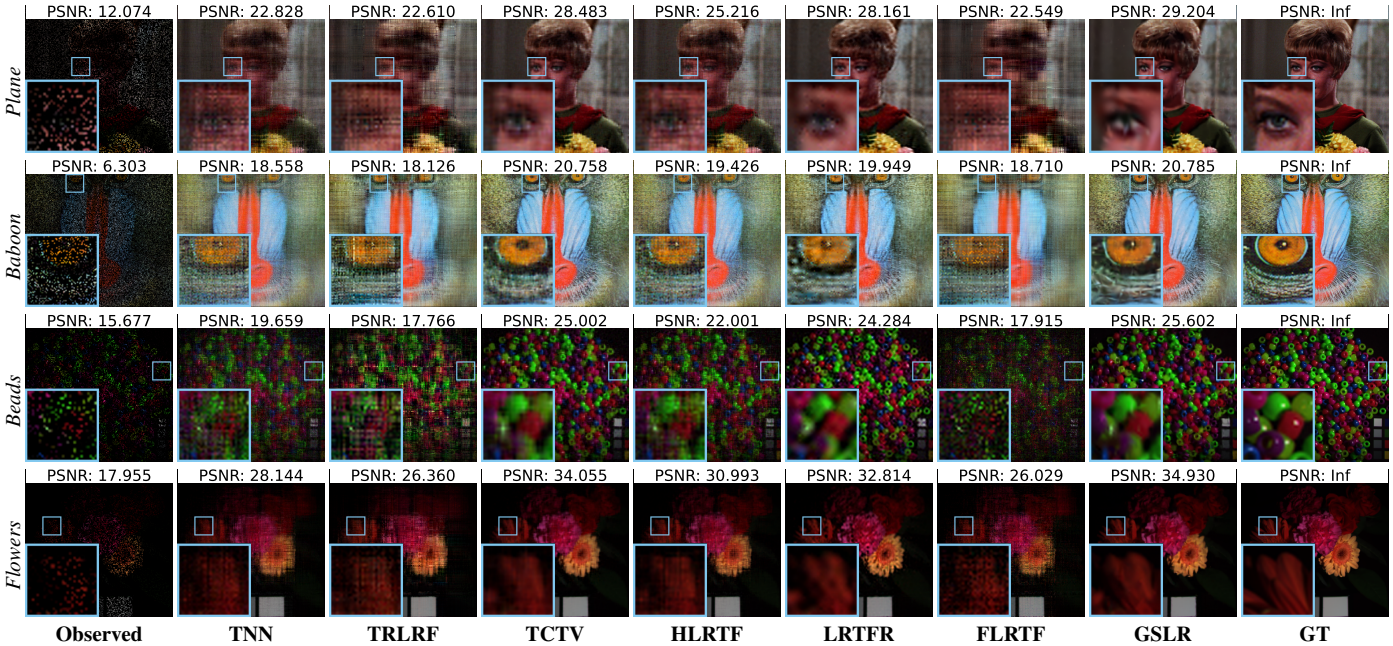

Figure 6. Reconstructed results and zoomed-in details by different methods under the tube missing (SR = 0.20).

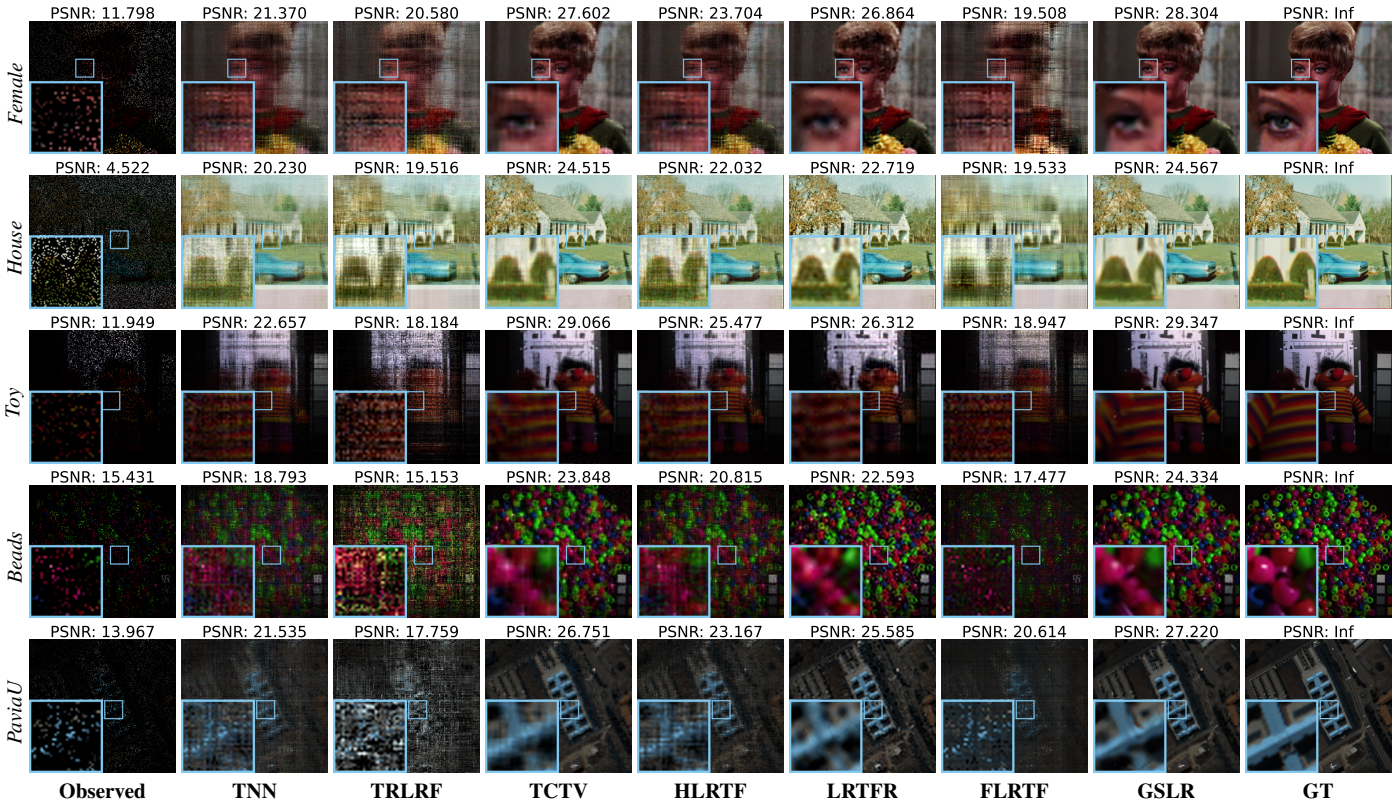

Figure 7. Reconstructed results and zoomed-in details by different methods under the tube missing (SR = 0.15).

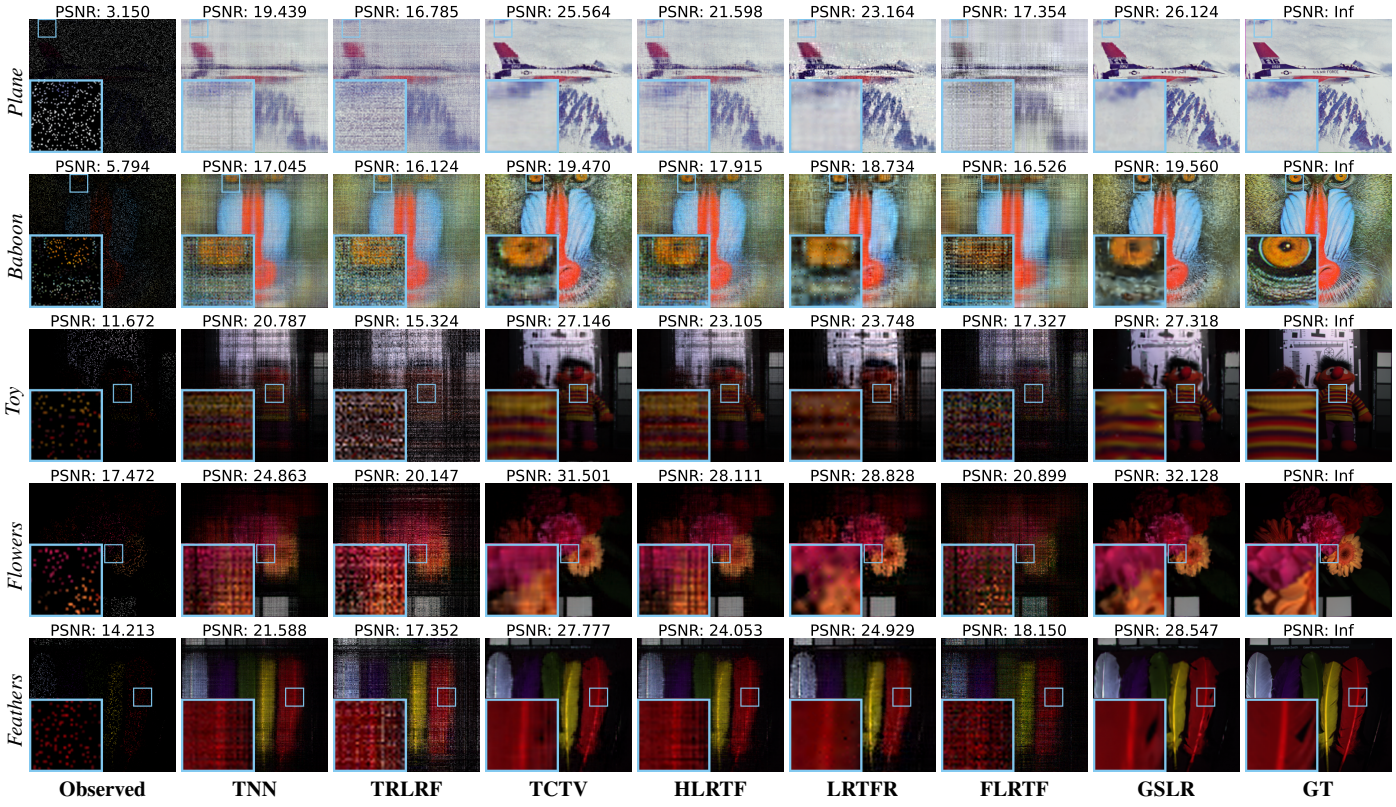

Figure 8. Reconstructed results and zoomed-in details by different methods under the tube missing (SR = 0.10).