PlaNet: Learning to Mitigate Atmospheric Turbulence in Planetary Images Supplementary Materials

Yifei Xia[1,2†], Chu Zhou[3†], Chengxuan Zhu[4], Chao Xu[4], Boxin Shi[1,2*]

[1]National Key Laboratory for Multimedia Information Processing, School of Computer Science, Peking University
[2]National Engineering Research Center of Visual Technology, School of Computer Science, Peking University
[3]National Institute of Informatics, Japan
[4]National Key Lab of General AI, School of Intelligence Science and Technology, Peking University
{yfxia, peterzhu, shiboxin}@pku.edu.cn, zhou_chu@hotmail.com, xuchao@cis.pku.edu.cn

Details About Our Synthetic Dataset

In this section, we give a thorough overview of the celestial objects featured in our synthetic dataset and the process of turbulence simulation, corresponding to Footnote 5 of the main paper.

Selected celestial bodies. To adequately expose the network to prior knowledge of celestial bodies, we downloaded as many 3D models of celestial bodies as possible from the NASA website[1]. These models are drawn based on the data from space probes and cover various common types of celestial bodies in the solar system, including:

- 9 planets, namely Mercury, Venus, Venus (surface), Earth, Mars, Jupiter, Saturn, Uranus and Neptune. Planets are the largest bodies orbiting the Sun. For Venus, there are two different models representing the planet with and without its atmosphere;
- 4 dwarf planets, namely Ceres, Makemake, Haumea, and Eris. Dwarf planets are celestial bodies that, like planets, orbit the Sun, but they are not dominant in their orbital zone;
- 3 asteroids, namely Bennu, Itokawa and Vesta. Asteroids are small, rocky bodies that orbit the Sun and are found mainly in the asteroid belt between the orbits of Mars and Jupiter;
- 11 satellites of various planets, namely the Moon (of Earth), Phobos (of Mars), Io, Europa, Ganymede and Callisto (of Jupiter), Enceladus, Tethys, Dione, Rhea and Iapetus (of Saturn). Satellites are celestial bodies that orbit planets.

Examples of these four types of celestial bodies are shown in Figure 6. It should be specifically noted that Venus, Mars, Jupiter, Saturn, and the Moon are common targets of real-world planetary imaging. To ensure fairness, we designated these five celestial bodies strictly for the test set and did not utilize them in the training process of our network.

Details of the simulation pipeline. Here, we provide more details on the turbulence simulation pipeline, as illustrated in Fig. 2 of the main paper. We first use the layer-wise parameter configurator to assign D_i and r_{0_i} to each layer. Initially, we randomly select parameters D and r_0, where D and r_0, respectively, represent the diameter of the photographer's lens and the total Fried parameter along the path. Since the observed turbulence intensity depends on $\frac{D}{r_0}$, we actually select the parameters D and $\frac{D}{r_0}$ for our study. The two parameters are randomly chosen for each ground truth and its corresponding disturbed images, as recommended by (Zhang et al. 2022). In our case, the value of D is uniformly distributed between 0.1 m and 0.4 m, while $\frac{D}{r_0}$ is uniformly distributed between 1 and 3, corresponding to the common apertures of the planetary observation telescopes and the typical turbulence magnitude during planetary photography.

To obtain the r_{0_i} for each layer, we configure them according to Table 4 in (Li et al. 2020), so that the variance of phase fluctuation between the actual beam and the approximate beam is minimized, and r_0 satisfies the relation with r_{0_i} (for $i = 1, 2, 3$) as

$$r_0 = \left(\sum_{i=1}^{3} r_{0_i}^{-5/3} \right)^{-3/5}. \quad (1)$$

The specific configuration is shown in Table 3, where α is a constant that can be solved by substituting the total Fried parameter r_0 along with the physically configured r_{0_i} into Equation (1), and D_i is the equivalent aperture at the bottom of the atmosphere of the i-th layer, defined as

$$D_i = \begin{cases} D_0 + \theta_0 \sum_{j=i}^{2} \mathcal{V}_{j+1}(C), & i = 1, 2 \\ D_0, & i = 3 \end{cases}$$

Here, θ_0 is the angular extent of the celestial body to the observer. As shown in Figure 7, the decreasing trend in mean variance with increasing N demonstrates the effectiveness of finer layer-wise segmentation in capturing the subtle variations in atmospheric turbulence, yet as N increases further, the diminishing returns in variance reduction suggest that $N = 3$ represents an optimal trade-off. Following this, we simulate different turbulence magnitudes for each layer in a top-to-bottom sequence using the pre-trained model provided by (Mao, Chimitt, and Chan 2021). Subsequently, the

[†]These authors contributed equally. Chu Zhou completed most of his contributions during his PhD studies at Peking University.

[*]Corresponding author.

[1]https://solarsystem.nasa.gov/resources/

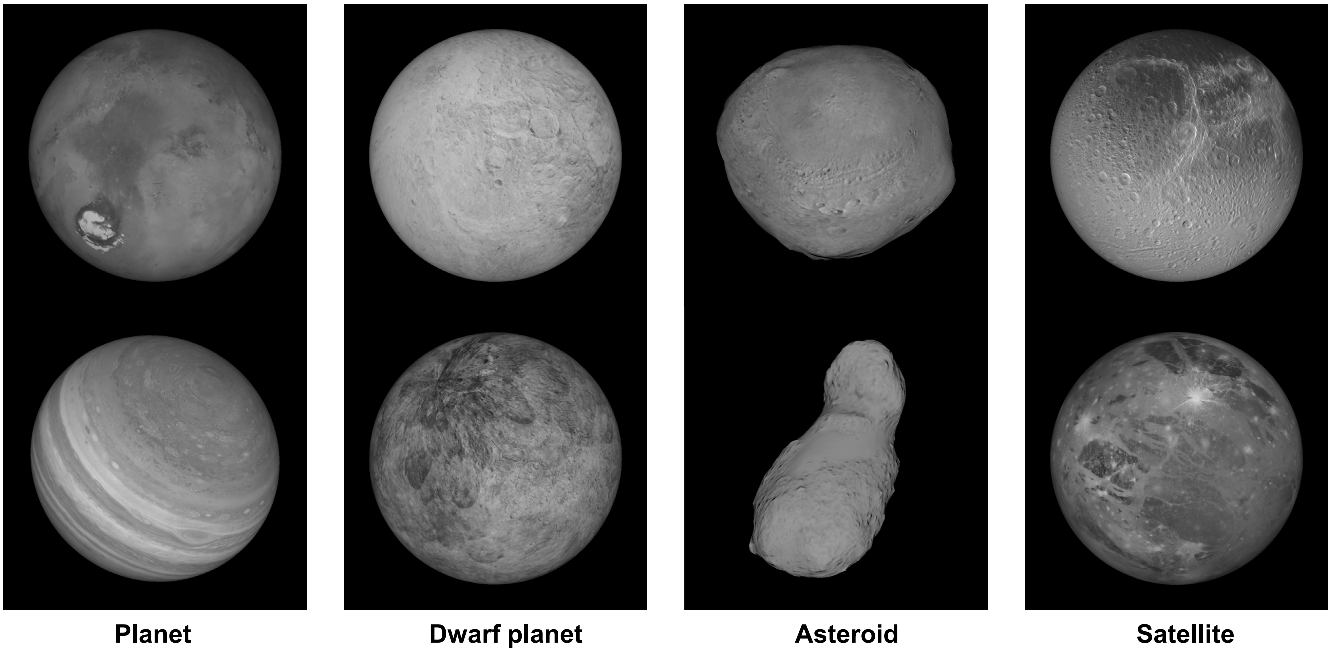

Figure 6: Visual representation of four types of celestial objects included in our synthetic dataset. From left to right, the images show two examples for each type: planets (Mars and Jupiter), dwarf planets (Eris and Makemake), asteroids (Bennu and Itokawa), and satellites (Dione and Ganymede).

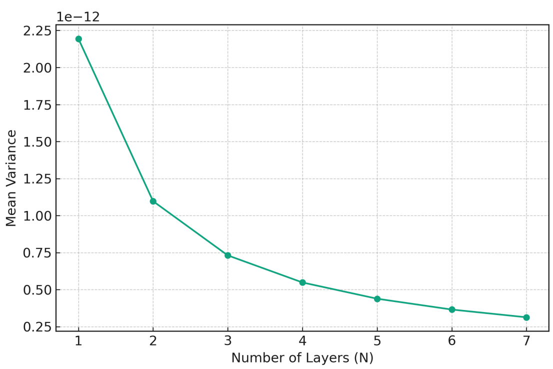

Figure 7: Mean variance of atmospheric turbulence magnitude as a function of the number of discrete atmospheric layers (N) used in the simulation. Each layer is configured with distinct values of D_i and r_{0_i}, contributing to the overall turbulence profile captured by our pipeline. Our choice $N = 3$ significantly reduces the mean variance of turbulence magnitude, without incurring too much computational cost.

simulation passes through a noise generator to yield the final simulation result. An additional visual comparison of the photon simulator PhoSim (Peterson et al. 2015) and our simulation method is presented in Figure 8, showing that our simulation method can produce similar result to the physically accurate photon simulator, while much more efficient in time.

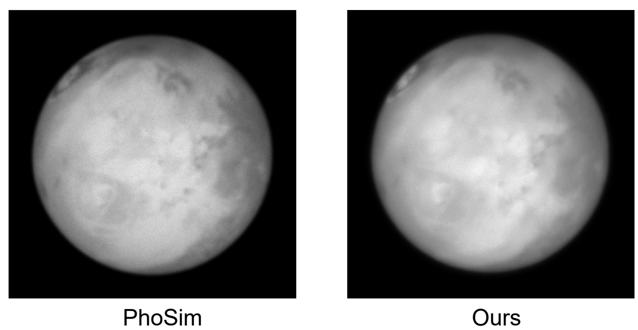

Figure 8: Comparison between PhoSim (Peterson et al. 2015) and our simulation method, demonstrating that our simulation method is a close approximation to the physically accurate photon simulator PhoSim (Peterson et al. 2015).

Table 3: This table presents the configuration of each atmospheric layer in our turbulence simulation pipeline, as referenced in Fig. 2 of the main paper. We provide the values of $\mathcal{V}_i(C)/\mathrm{m}$ and r_{0_i}/m for each layer, which are essential for reducing the variance of phase fluctuation between the actual and approximate beams. The values of r_{0_i} are set according to Table 4 in (Li et al. 2020), making sure that Equation (1) is followed for the connection between D_i and r_{0_i}.

i	$\mathcal{V}_i(C)/\mathrm{m}$	r_{0_i}/m
1	4877	0.6844 α
2	3860	0.3941 α
3	1263	0.0895 α

Table 4: Quantitative evaluation of different layer configurations. PSNR and SSIM values are shown for each configuration.

C	$\mathcal{V}(C)$	PSNR (dB)	SSIM
Ours	Ours	**27.78**	**0.9007**
Doubled	Ours	22.12	0.7928
Halved	Ours	22.73	0.8138
Ours	All the Same	25.56	0.8499
Ours	Reversed	20.92	0.7779

Evaluation of the turbulence simulator. We compare different layer configurations by creating datasets and summarizing the quantitative results in Table 4, where C is the overall turbulence strength, and $\mathcal{V}(C)$ represents its variation across layers (see Equation 2 in the main paper for details). First, we vary C while keeping $\mathcal{V}(C)$ constant, noting a significant performance drop. Then, we keept C constant and modified $\mathcal{V}(C)$, either making strength uniform across layers or reversing its order, both of which lead to significant degradation. The *all the same* case in the table refers to no distance awareness. These experiments show that our distance-aware simulator configuration achieves superior performance.

Why not color image? Taking pictures of planets typically involves using black-and-white CMOS or CCD sensors to capture images with different band filters due to the faint light of planets. This technique, which focuses on grayscale imaging, maximizes image clarity and resolution by capturing all incoming photons, resulting in higher clarity and signal-to-noise ratio (SNR), especially in scenarios with short exposure times. As planetary objects are usually faint, as mentioned in (Li et al. 2020), grayscale images are preferred for more detailed observations. Before creating color images from these grayscale pictures, it is essential to perform turbulence removal operations, allowing for the processing of grayscale images only.

Details of PlaNet Architecture

In this section, we will give a more thorough explanation of the design of our PlaNet network, including a comprehensive demonstration of how the network processes arbitrary input frames, and why the edge-based supervision is able to be effective in practice.

Permutation-invariant feature aggregation. The core design for processing arbitrary input frames is the proposed OBG module. As discussed in Sec. 4 of the main paper, a CBAM block is used to aggregate the recalibrated feature maps, and two ConvBlocks are used to decode the aggregated feature maps into a single output image. To illustrate this, let us assume an arbitrary number of N input grayscale frames with shape $(N, 1, H, W)$. The features of each frame are extracted and calibrated independently to get a tensor of shape (N, C, H, W), where C is the number of feature channels for each frame. After passing through the CBAM block and two ConvBlocks, the tensor's shape is transformed to $(1, C, H, W)$ and $(1, 1, H, W)$, resulting in the final output.

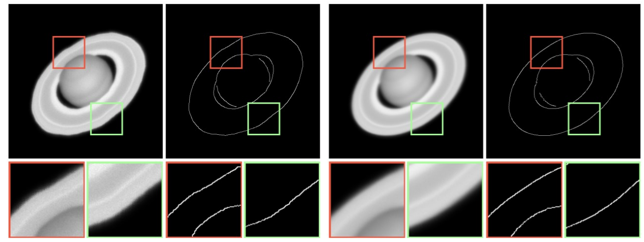

Figure 9: Visualization of the edge map showing the impact of atmospheric turbulence on individual frames and the efficacy of the averaging operation in mitigating distortion. Each frame shows its own unique distortions, which are significantly reduced in the stabilized image because of the zero-mean Gaussian distribution that is typical of atmospheric turbulence, leading to a less distorted edge map.

Edge-based supervision. We present a visualization of the edge map in Figure 9, which shows how each input frame is affected by atmospheric distortions. These distortions, mainly characterized by a zero-mean Gaussian distribution, are substantially reduced through our averaging process. This technique effectively stabilizes the image, significantly diminishing the effects of atmospheric distortion. Notably, while the boundary of the planet appears blurry with low contrast, our approach relies on edge supervision instead of direct supervision in the image domain. This strategy takes advantage of the stabilizing effect of the edges, ensuring improved definition and clarity. Therefore, even though some blurriness is inherent due to atmospheric conditions, our method avoids the reconstruction of an overly blurred image, preserving a balance between stability and image sharpness.

Additional Comparisons on Synthetic and Real Data

This section extends the comparison of the software AutoStakkert with four state-of-the-art learning-based methods (Mao et al. 2022; Li et al. 2021; Dudhane et al. 2022; Chan et al. 2022) to both synthetic and real datasets. Specifically, for synthetic data, Figure 10 corresponds to Footnote 6 in the main paper. Similarly, for real data, Figure 11 is aligned with Footnote 8 in the main paper.

References

Chan, K. C.; Zhou, S.; Xu, X.; and Loy, C. C. 2022. Basicvsr++: Improving video super-resolution with enhanced propagation and alignment. In *Proc. of IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Dudhane, A.; Zamir, S. W.; Khan, S.; Khan, F. S.; and Yang, M.-H. 2022. Burst image restoration and enhancement. In *Proc. of IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Li, N.; Thapa, S.; Whyte, C.; Reed, A. W.; Jayasuriya, S.; and Ye, J. 2021. Unsupervised non-rigid image distortion removal via grid deformation. In *Proc. of International Conference on Computer Vision*.

Li, R.; Luo, L.; Li, J.; and Gao, X. 2020. Simulation of anisoplanatic imaging containing optical system parameters through atmospheric turbulence. *Optik*, 204: 164177.

Mao, Z.; Chimitt, N.; and Chan, S. H. 2021. Accelerating atmospheric turbulence simulation via learned phase-to-space transform. In *Proc. of International Conference on Computer Vision*, 14759–14768.

Mao, Z.; Jaiswal, A.; Wang, Z.; and Chan, S. H. 2022. Single frame atmospheric turbulence mitigation: A benchmark study and a new physics-inspired transformer model. In *Proc. of European Conference on Computer Vision*, 430–446.

Peterson, J.; Jernigan, J.; Kahn, S.; Rasmussen, A.; Peng, E.; Ahmad, Z.; Bankert, J.; Chang, C.; Claver, C.; Gilmore, D.; Grace, E.; Hannel, M.; Hodge, M.; Lorenz, S.; Lupu, A.; Meert, A.; Nagarajan, S.; Todd, N.; Winans, A.; and Young, M. 2015. Simulation of astronomical images from optical survey telescopes using a comprehensive photon Monte Carlo approach. *The Astrophysical Journal Supplement Series*, 218(1): 14.

Zhang, X.; Mao, Z.; Chimitt, N.; and Chan, S. H. 2022. Imaging through the atmosphere using turbulence mitigation transformer. *arXiv preprint arXiv:2207.06465*.

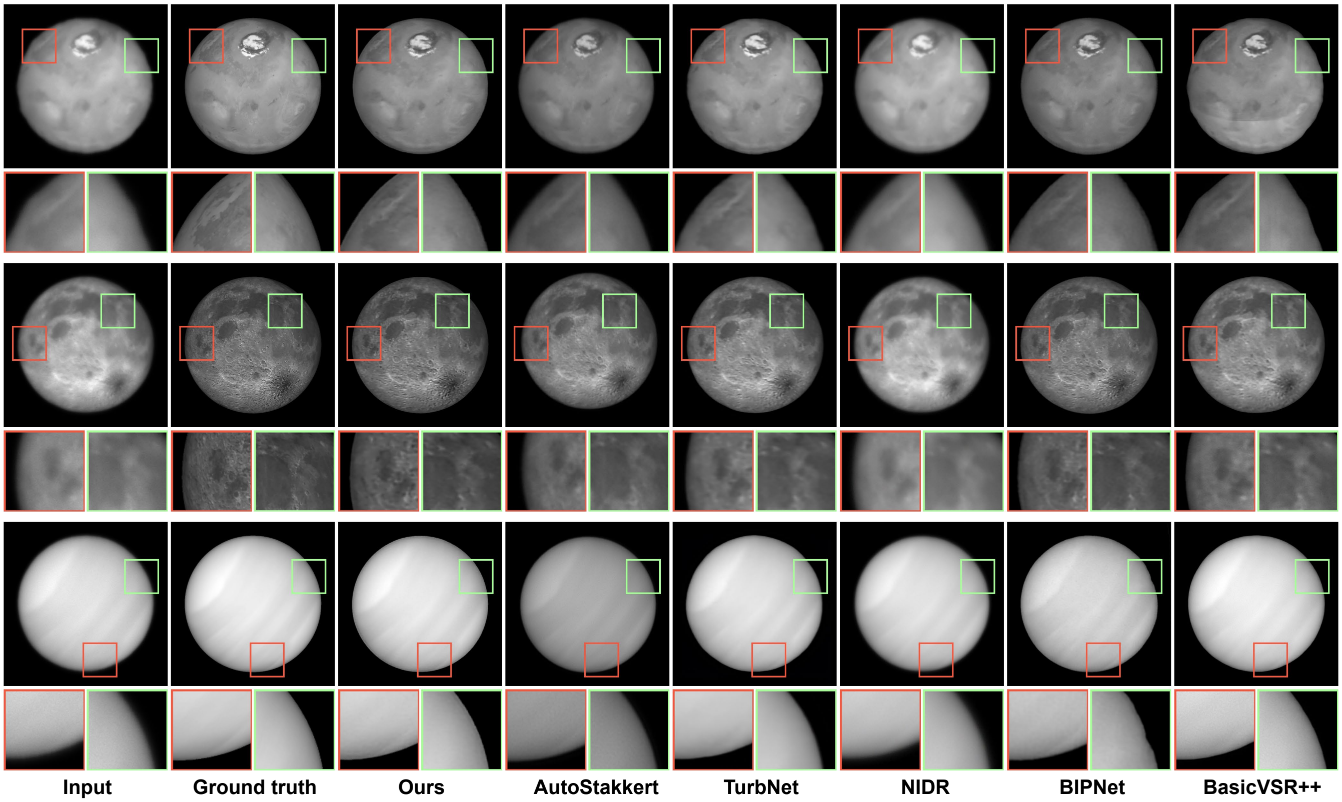

Figure 10: Qualitative comparisons on synthetic data among our method, a representative planetary imaging software AutoStakkert, and several state-of-the-art learning-based methods that solve the closest problems including TurbNet (Mao et al. 2022), NIDR (Li et al. 2021), BIPNet (Dudhane et al. 2022), and BasicVSR++ (Chan et al. 2022).

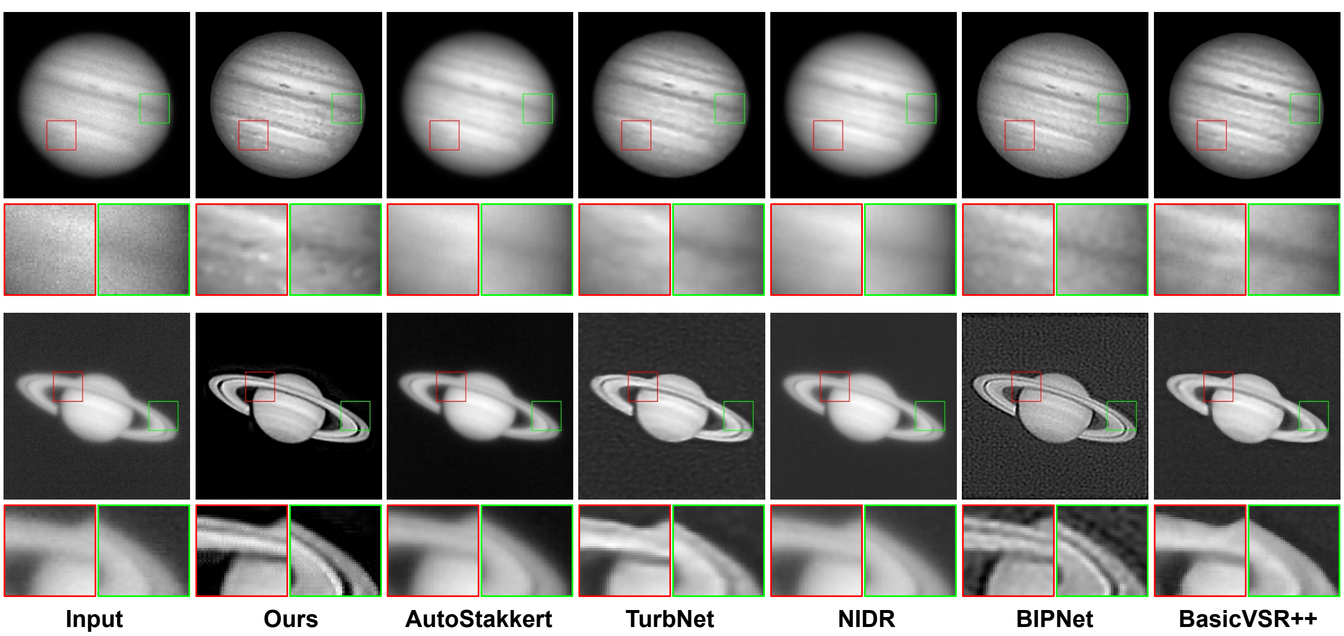

Figure 11: Qualitative comparisons on real data among our method, a representative planetary imaging software AutoStakkert, and several state-of-the-art learning-based methods that solve the closest problems including TurbNet (Mao et al. 2022), NIDR (Li et al. 2021), BIPNet (Dudhane et al. 2022), and BasicVSR++ (Chan et al. 2022).