PlaNet: Learning to Mitigate Atmospheric Turbulence in Planetary Images

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

Abstract

Obtaining planetary images with good visual quality is not an easy task since they are usually degenerated by atmospheric turbulence during the imaging procedure. Existing atmospheric turbulence mitigation methods designed for conventional images cannot be applied to planetary images, since the objects on the Earth have totally different degeneration patterns to planets. Besides, in planetary imaging, photographers often capture as many frames as possible to reduce the noise level of planetary images, which requires the method designed for planetary images to support an arbitrary number of input frames. In this paper, we propose a vertical distance-aware turbulence simulation pipeline to synthesize realistic planetary images in accordance with their unique degeneration patterns at a large scale with affordable computational cost, and design a neural network to mitigate the turbulence with flexible input frames by adopting an edge-based supervision strategy to handle the background scarcity issue. Experimental results show that our method achieves state-of-the-art performance on both synthetic and real-world images.

Introduction

Planetary imaging aims to capture the planets within our solar system on the Earth. Since the distribution of atmospheric refraction index is spatially-variant (Roggemann, Welsh, and Hunt 1996), the light waves passing through the atmosphere would be severely distorted, resulting in degenerated captured images. Such degeneration is called *atmospheric turbulence*, which can be regarded as an entanglement of pixel displacement (known as the tilt), blur, and noise (Fazlali et al. 2022; Zhang et al. 2022b; Mao et al. 2022). To obtain a turbulence-free planetary image, photographers usually record a large number of frames (usually ranging from hundreds to thousands) of the same scene, and fuse them using softwares (*e.g.*, AutoStakkert[1]). However, the fusing results are not yet satisfactory, since these softwares only keep the "lucky" frames (frames with relatively less distortion) by simply discarding other frames, while the "lucky" frames are not always available (Fried 1978) so that the information contained in the discarded frames are completely unexploited. Therefore, effectively mitigating atmospheric turbulence in planetary images is still an open problem.

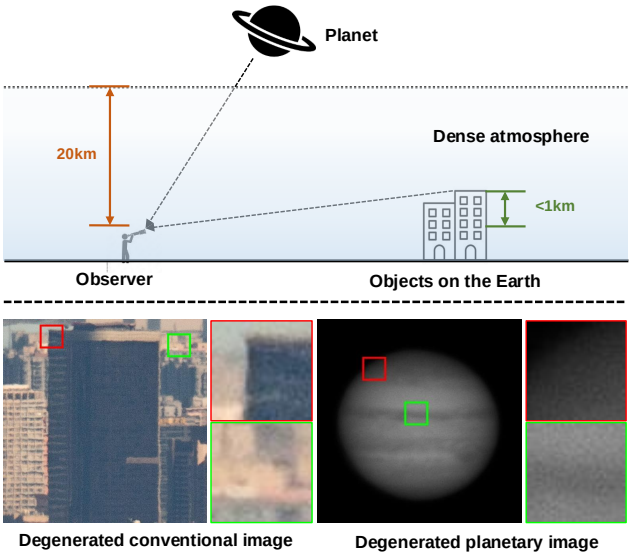

Figure 1: Top: An illustration to show the difference of vertical distance (marked using orange and green arrows) between planets and the objects on the Earth. Bottom: A real-world example to demonstrate the domain gap between conventional images and planetary images.

Recently, neural networks have been introduced to mitigate atmospheric turbulence in conventional images, *i.e.*, images only containing objects on the Earth (Gao, Anantrasirichai, and Bull 2019; Mao et al. 2022; Rai and Jawahar 2022; Chak, Lau, and Lui 2018; Jin et al. 2021; Li et al. 2021; Fazlali et al. 2022; Zhang et al. 2022b). They aim to reconstruct the corresponding clean image from a single degenerated image (Gao, Anantrasirichai, and Bull 2019; Mao et al. 2022; Rai and Jawahar 2022) or a fixed number of degenerated frames (Chak, Lau, and Lui 2018; Jin et al. 2021; Li et al. 2021; Fazlali et al. 2022; Zhang et al. 2022b), by extracting semantic and physical priors from a

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

[*]Corresponding author.

[1]https://www.autostakkert.com/

large amount of training data. However, as shown in Figure 1 (top), since the vertical distance[2] of a planet (around 20 km) is much larger than the one of an object on the Earth (typically less than 1 km), and the variance of turbulence magnitude monotonically increases with the vertical distance, (Vyhnalek 2017), planetary images and conventional images have totally different degeneration patterns. Besides, as shown in the green box of Figure 1 (bottom), the noise level in planetary images is usually much higher than in conventional images since the lighting condition in outer space is not as good as the one on the Earth (Li et al. 2020). Furthermore, as shown in the red box of Figure 1 (bottom), the contrast between the foreground and background of planetary images is much higher than conventional images (*i.e.*, the background of planetary images is extremely dark). These unique properties of planetary images pose the following major difficulties in the turbulence mitigation process:

1. Existing turbulence simulation methods designed for conventional images (Schwartzman et al. 2017; Chimitt and Chan 2020) cannot generate realistic planetary images due to the differences in degeneration patterns.
2. In planetary imaging, photographers often capture as many frames as possible to reduce the noise level (Li et al. 2020), which requires the method to support an arbitrary number of input frames.
3. The extremely dark background of planetary images provides little semantic information or physical priors (we call it background scarcity issue), making the problem highly ill-posed.

In this paper, to synthesize realistic planetary images in accordance with their unique degeneration patterns at a large scale, we propose a new turbulence simulation pipeline specially designed for planetary images to simulate the light propagation procedure with affordable computational cost in a vertical distance-aware manner. To support an arbitrary number of input frames for reducing the noise level in planetary imaging, we propose **PlaNet** which solves the atmospheric turbulence mitigation problem for **Pla**netary images with specially designed neural **Net**works, capable of making full use of the information contained in all input frames. To handle the background scarcity issue, we explicitly utilize the edge map of the average frame as an external prior to provide supervision in the latent space with multi-scale observations, based on the fact that the variation of the turbulence field in planetary images over time follows a zero-mean Gaussian distribution (Zhu and Milanfar 2011) so that the average frame often has a similar boundary to the corresponding clean image.

In summary, this paper makes contributions to overcome the difficulties in the turbulence mitigation process of planetary images by demonstrating:

1. a vertical distance-aware turbulence simulation pipeline for planetary images;
2. a specially designed network with flexible input frames to reduce the noise level; and
3. an edge-based supervision strategy to handle the background scarcity issue.

Experimental results show that our method achieves state-of-the-art performance on both synthetic and real-world planetary images.

[2]The vertical distance is defined as the altitude difference between the observer and the object or the layer where most of the turbulence that affects image quality occurs.

Related work

Atmospheric turbulence simulation. Previous work on turbulence simulation has predominantly concentrated on modeling the degradation patterns of conventional images. Various methods have introduced hand-crafted image priors and manually constructed tilt, blur, and noise models to generate degraded images from their pristine versions (Zhu and Milanfar 2012; Chak, Lau, and Lui 2018; Gao, Anantrasirichai, and Bull 2019). Although visually appealing, these methods lack physical validity, creating a domain gap between synthesized and real-world images. To address this, some techniques used physical priors based on turbulence formation statistics for image synthesis (Hardie et al. 2017; Schwartzman et al. 2017; Chimitt and Chan 2020; Li et al. 2020), or induced turbulence effects through air heating along the imaging path (Jin et al. 2021; Mao et al. 2022). However, these approaches often incur significant computational expenses due to complex numerical optimization algorithms. To mitigate this, approximation strategies that utilize phase-to-space (P2S) transformations and neural networks have been proposed to replace these algorithms (Mao, Chimitt, and Chan 2021; Chimitt et al. 2022). Regarding planetary images, photon simulators (Peterson et al. 2015; Dobke et al. 2010) generate images by sampling photons from astronomical models and simulating their interactions with the atmosphere, telescope, and camera; however, this approach operates at the photon level, leading to prolonged processing times.

Atmospheric turbulence mitigation. The mitigation of atmospheric turbulence presents a highly ill-posed challenge due to the intricate structures of natural images. Some methods have focused on turbulence reduction in human faces (Yasarla and Patel 2020, 2021; Mei and Patel 2023). For terrestrial objects, earlier approaches used prior-based numerical optimization to restore clean images (Zhu and Milanfar 2012; Hirsch et al. 2010; Anantrasirichai et al. 2013; Lau, Lai, and Lui 2019; Mao, Chimitt, and Chan 2020). With advancements in deep learning, neural networks have been employed to reconstruct clean images from single degraded images (Gao, Anantrasirichai, and Bull 2019; Mao et al. 2022; Rai and Jawahar 2022) or a limited number of degraded frames (Chak, Lau, and Lui 2018; Jin et al. 2021; Li et al. 2021; Fazlali et al. 2022; Zhang et al. 2022b). However, these methods are not suitable for planetary images as they do not accommodate an arbitrary number of input frames. Although TurbuGAN (Feng, Xie, and Metzler 2022) supports a flexible number of input frames, it relies on iterative optimization, making it computationally expensive.

Image restoration. There are many other methods that have the partial ability to solve the atmospheric turbulence mitigation problem. For example, methods designed for video/burst image restoration/deblurring/enhancement

(Dudhane et al. 2022; Chan et al. 2022) can also take an arbitrary number of frames as input (more methods can be found in these surveys (Zhang et al. 2022a; Su, Xu, and Yin 2022)) and output images with enhanced visual quality. However, since they do not consider the unique properties of planetary images, their performance may not be optimized.

Image formation and turbulence simulation

Image formation model. For a better understanding of the atmospheric turbulence mitigation problem for planetary images, we first introduce the formation model of a degenerated planetary image. Denoting a clean image of a planet and its degenerated counterpart as $\mathbf{I}$ and $\mathbf{T}$ respectively, their relationship can be described as

$$\mathbf{T} = \mathcal{T}(\mathbf{I}) + \boldsymbol{\epsilon}, \tag{1}$$

where $\mathcal{T}$ denotes the total effects of the turbulence, including the blur and tilt artifact, and $\boldsymbol{\epsilon}$ is a noise term. Specifically, according to the physical-based imaging model of the atmospheric turbulence (Peterson et al. 2015), $\mathcal{T}$ can be decomposed into the effects of the turbulence induced when the light waves pass through the atmosphere and arrive at the observer. Denoting these two effects as $\mathcal{A}$ and $\mathcal{O}$ respectively, Equation (1) can be further formulated as

$$\mathcal{T}(\mathbf{I}) = \mathcal{O}\left(\mathcal{A}\left(\mathbf{I}; \mathcal{V}(C)\right); \frac{D}{r_0}\right). \tag{2}$$

We can see that $\mathcal{A}$ is dependent on $\mathcal{V}(C)$, which is the variance of the structural constant of atmospheric turbulence C (reflecting the turbulence magnitude of the atmosphere (Vyhnalek 2017))[3] during the light propagation procedure through the atmosphere, and $\mathcal{O}$ is dependent on $\frac{D}{r_0}$ (reflecting the turbulence level at the observer end (Mao, Chimitt, and Chan 2021)), where D is the aperture of the lens and r_0 is the Fried parameter (Fried 1978).

The image formation model adopted by existing turbulence simulation methods designed for conventional images (Schwartzman et al. 2017; Chimitt and Chan 2020; Mao, Chimitt, and Chan 2021) could be regarded as a simplified version of Equation (2). Specifically, they often ignore the influence of $\mathcal{V}(C)$, based on the following assumption:

$$\mathcal{A}\left(\mathbf{I}; \mathcal{V}(C)\right) \approx \mathbf{I}. \tag{3}$$

It is a fact that C varies along the path of light passing through the atmosphere from outer space to the ground (Vyhnalek 2017). Since the vertical distance of an object on the Earth is small enough (see Figure 1 (top)), *i.e.*, the influence of $\mathcal{V}(C)$ is ignorable, Equation (3) holds true for conventional images. However, it is not the case for planetary images, since the vertical distance of a planet in outer space is large, making the influence of $\mathcal{V}(C)$ not ignorable. Therefore, existing turbulence simulation methods designed for conventional images (Schwartzman et al. 2017; Chimitt and Chan 2020; Mao, Chimitt, and Chan 2021) cannot be applied to planetary images. Figure 2 (right) shows an example of using two frequently-used simulation methods designed for conventional images (proposed by Schwartzman *et al.*(Schwartzman et al. 2017) and Mao *et al.*(Mao, Chimitt, and Chan 2021)) to synthesize planetary images. We can see that they produce unreasonable degeneration patterns compared with the real planetary image.

[3]The structural constant of atmospheric turbulence is usually denoted using C_n^2 in relevant papers like (Vyhnalek 2017). For simplicity of notations, we use C to denote the same term in this paper.

Turbulence simulation pipeline. To train a network for mitigating atmospheric turbulence in planetary images, a large-scale dataset containing pairwise clean and degenerated images is required. In the field of astronomy, photon simulators (such as PhoSim (Peterson et al. 2015)) are common choices to generate planetary images with realistic atmospheric turbulence. However, they usually require an excessively long running time (more than half an hour for a planetary image with 512 $\times$ 512 pixels on an Intel Core i7-7800X CPU) since they work at the photon level (*i.e.*, taking photons as input), making them not suitable for generating a large-scale dataset.

To generate realistic degeneration patterns for planetary images with affordable computational cost, we propose a vertical distance-aware turbulence simulation pipeline working at the pixel level (*i.e.*, taking a single clean image as input, similar to the behavior of existing turbulence simulation methods designed for conventional images (Hardie et al. 2017; Schwartzman et al. 2017; Chimitt and Chan 2020; Mao, Chimitt, and Chan 2021)). As shown in Figure 2 (left), we divide the path of light passing through the atmosphere into three layers (upper/middle/bottom atmosphere) based on their different spans of vertical distance, and treat them as three cascaded "optical systems". Given a preset overall D and r_0, we first adopt a layer-wise parameter configurator, to configure the layer-wise turbulence level, by assigning a reasonable $\frac{D}{r_0}$ for each layer to ensure physical justifiability. Then, we adopt a inner-layer turbulence simulator to simulate the specific turbulence effect inside each layer, based on the assigned $\frac{D}{r_0}$ and the statistical distribution of the relationship between the span of vertical distance and $\mathcal{V}(C)$ (Vyhnalek 2017). According to Equations (1) and (2), the relationship between the input image $\mathbf{I}_{i_{\text{in}}}$ and output image $\mathbf{I}_{i_{\text{out}}}$ of the i-th layer ($i = 1, 2, 3$) can be described as (the noise term $\boldsymbol{\epsilon}$ is omitted for now):

$$\mathbf{I}_{i_{\text{out}}} = \mathcal{O}\left(\mathcal{A}\left(\mathbf{I}_{i_{\text{in}}}; \mathcal{V}_i(C)\right); \frac{D_i}{r_{0_i}}\right), \tag{4}$$

where we use the subscript i to mark the parameters in the i-th layer. Since in each layer, the influence of $\mathcal{V}(C)$ does not change too much, *i.e.*, Equation (3) holds, we choose to adopt the approximation strategy proposed by Mao *et al.*(Mao, Chimitt, and Chan 2021) (which is designed for conventional images, ignoring the influence of $\mathcal{V}(C)$ for reducing the running time) to speed up the simulation process inside each layer. Thanks to the vertical distance-aware design, our method can generate a planetary image with 512 $\times$ 512 pixels in 9 seconds on an NVIDIA 1080Ti GPU, *i.e.*, runs hundreds of times faster than photon simulators (Peterson et al. 2015), while presenting more realistic results.

Our turbulence simulation pipeline takes a 3D planet

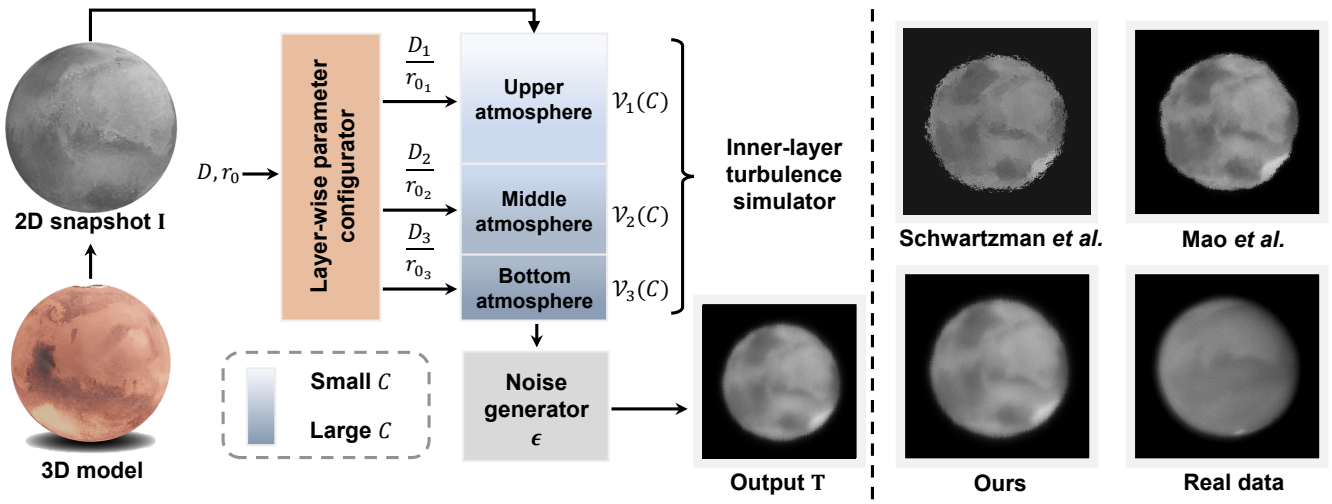

Figure 2: Left: The proposed turbulence simulation pipeline for planetary images working in a vertical distance-aware manner. Right: Visual comparisons of the simulation results from existing simulation methods designed for conventional images (proposed by Schwartzman *et al.*(Schwartzman et al. 2017) and Mao *et al.*(Mao, Chimitt, and Chan 2021)) and our method along with a real planetary image (an observation sample taken from a different angle).

model (obtained from the NASA website[4]) as input and outputs a degenerated image (or outputs an arbitrary number of degenerated frames, by running the pipeline multiple times) along with the corresponding clean image (as the ground truth), as shown in Figure 2 (left). Specifically, we first sample 2D snapshots from the 3D planet model in a uniform manner to obtain the clean images, then feed them into the inner-layer turbulence simulator with the parameters produced from the layer-wise parameter configurator to obtain the degenerated image, and add the noise term ϵ in the end. From Figure 2 (right) we can see that our method can synthesize planetary images with the degeneration patterns resembling the real planetary images more closely[5].

PlaNet architecture design

Given the high-quality synthetic data, we propose PlaNet, a neural network for mitigating the atmospheric turbulence in planetary images with simple but effective module designs. Due to the poor lighting condition in outer space, the noise level in planetary images is usually much higher than in conventional images, which often requires capturing as many frames as possible to reduce the noise level (Li et al. 2020). Besides, since the background of planetary images is extremely dark, providing little semantic information or physical priors, mitigating the turbulence in planetary images could be more challenging than conventional images. Therefore, unlike existing methods designed for conventional images that only consider reconstructing the corresponding clean image from a single degenerated image (Gao, Anantrasirichai, and Bull 2019; Mao et al. 2022; Rai and Jawahar 2022) or a fixed number of degenerated frames (Chak, Lau, and Lui 2018; Jin et al. 2021; Li et al. 2021; Fazlali et al. 2022; Zhang et al. 2022b), PlaNet should support an arbitrary number of input frames and can fully exploit their information. Besides, an effective strategy to handle the background scarcity issue should be integrated into PlaNet. As shown in Figure 3, denoting PlaNet as f, it can be written as

$$\mathbf{I} = f(\mathbf{T}_1, \cdots, \mathbf{T}_N), \tag{5}$$

where $\mathbf{T}_i (i = 1, \ldots, N)$ denotes a degenerated input frame, $\mathbf{I}$ denotes the output clean image, and N is the number of input frames (can be an arbitrary positive integer). And we will introduce the module design of PlaNet in the following paragraphs.

Permutation-invariant feature aggregation. Due to the atmospheric time constant, typically under 10 milliseconds (Tokovinin 2002), adjacent frames in an input sequence exhibit low correlation. Consequently, we design PlaNet to ensure permutation invariance. For each input frame $\mathbf{T}_i$, we first process it through a backbone network to extract its feature map, followed by the OBG (output by group) module, which aggregates these feature maps in a permutation-invariant manner. This design allows PlaNet to accommodate any number of input frames while fully utilizing the information from all frames. The backbone network employs a U-Net architecture (Ronneberger, Fischer, and Brox 2015), known for its effective multi-scale information extraction in computational photography (Zhou et al. 2020, 2021), with an SE (squeeze-and-excitation) block (Hu, Shen, and Sun 2018) added in the coarsest layer for enhanced detail. All input frames share the same backbone weights to minimize network size. The OBG module integrates a CBAM (convolutional block attention module) block (Woo et al. 2018) that recalibrates feature maps using attention for adaptive refinement, along with two ConvBlocks that decode the aggregated feature maps into a single output image.

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

[5]Additional evaluation of the simulator can be found in the supplementary material.

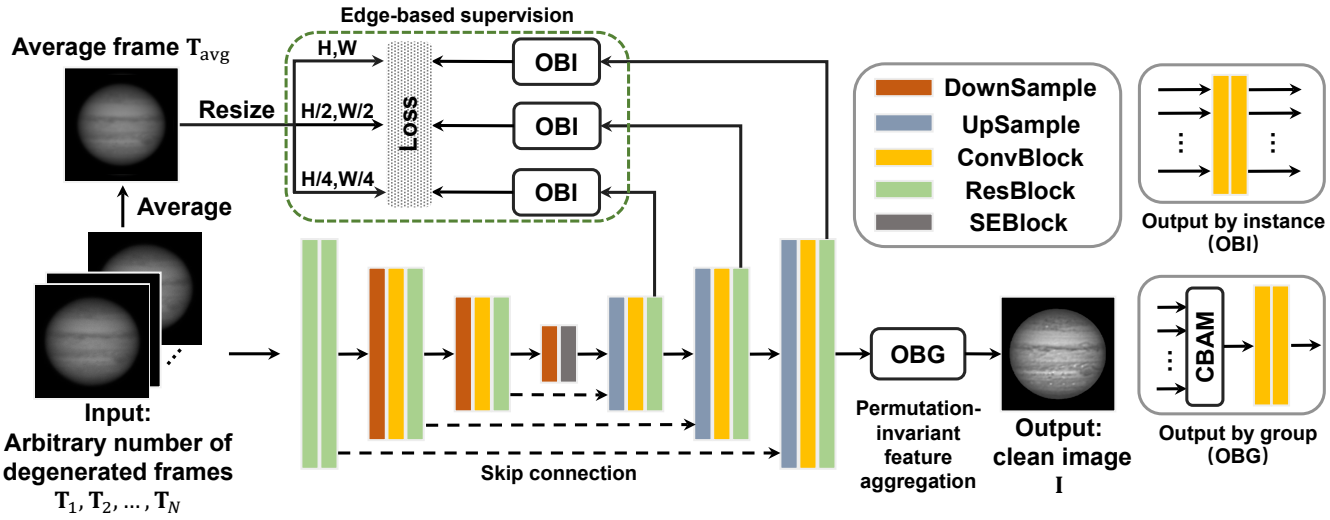

Figure 3: Architecture of the proposed PlaNet designed for mitigating the atmospheric turbulence in planetary images. It takes an arbitrary number of degenerated frames $\mathbf{T}_1, \mathbf{T}_2, \cdots, \mathbf{T}_N$ (N is the number of input frames, which can be an arbitrary positive integer) as input, and outputs a single clean image $\mathbf{I}$.

Edge-based supervision. Due to the significant distance between planets and Earth, both the observer and the planets can be considered relatively stationary. In this context, the turbulence in planetary images over time adheres to a zero-mean Gaussian distribution (Zhu and Milanfar 2011). Consequently, while the average frame $\mathbf{T}_{\text{avg}}$ may appear blurrier, it is less affected by tilt artifacts, resulting in boundaries that closely resemble those of the corresponding clean image. Leveraging this characteristic, we propose an edge-based supervision strategy to address background scarcity and reduce blurriness from averaging. This approach employs the Laplacian edge map of $\mathbf{T}_{\text{avg}}$ as an external prior for supervision in the latent space using multi-scale observations. We introduce three OBI (output by instance) modules at different scales within the decoder of the backbone network to generate multi-scale feature maps. A Laplacian edge extractor is then used to obtain and resize the edge map of $\mathbf{T}_{\text{avg}}$ to supervise these feature maps through loss computation. Each OBI module consists of two ConvBlocks without feature aggregation, with shared weights across feature maps from different frames.

Implementation details

Dataset preparation. We first collect 27 different 3D planet models in our solar system, including 9 planets (with two different models for Venus), 4 dwarf planets, 3 asteroids, and 11 satellites. Then, we split them into two parts that contain 22 and 5 different 3D models for making the training and test sets respectively. For each 3D model, we first position virtual cameras in the direction of the vertices of a regular dodecahedron that shared a center point with the 3D model to uniformly sample 12 2D snapshots using orthographic projection as the ground truth clean image, then use the proposed vertical distance-aware turbulence simulation pipeline to generate a sequence of degenerated frames for each clean image. We further perform data augmentation (such as random flipping and rotating) on the images for training, so that the training (testing) set contains 1056 (60) different image sequences finally. Note that the images are resized to 256×256 (512×512) pixels in the training (test) set.

Loss function. The overall loss function L is defined as

$$L = L_{\text{out}} + \lambda L_{\text{dec}}, \tag{6}$$

where L_{out} and L_{dec} denote the output loss term and decoder loss term respectively, and λ is a weighting coefficient set to be 10.0. The output loss term L_{out} computes the loss between the output image $\mathbf{I}$ and the corresponding ground truth clean image $\mathbf{I}_{\text{gt}}$, which can be written as

$$L_{\text{out}} = L_2(\mathbf{I}, \mathbf{I}_{\text{gt}}), \tag{7}$$

where L_2 denotes the ℓ_2 loss. The decoder loss term L_{dec} computes the loss between the feature map $\mathbf{F}_i$ of each input frame $\mathbf{T}_i$ ($\mathbf{F}_i$ is obtained from the decoder part of the backbone network using OBI modules) and the edge map $\mathbf{E}_i$ of the average frame $\mathbf{T}_{\text{avg}}$ ($\mathbf{E}_i$ is extracted using a Laplacian edge extractor) in three different scales, which can be written as

$$L_{\text{dec}} = \frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{3} \alpha_j L_1(\mathbf{F}_i^{(j)}, \mathbf{E}_i^{(j)}), \tag{8}$$

where N is the number of input frames, L_1 denotes the ℓ_1 loss, the superscript j denotes the j-th scale, and $\alpha_{1,2,3}$ are weighting coefficients set to be 9.0, 3.0, and 1.0 respectively.

Training strategy. We implement our method using PyTorch on a computer with an Intel Xeon Platinum 8358P CPU and two NVIDIA A100 GPUs. The network is trained for 100 epochs with a batch size of 8. For optimization, we use Adam optimizer (Kingma and Ba 2014) ($\beta_1 = 0.9$,

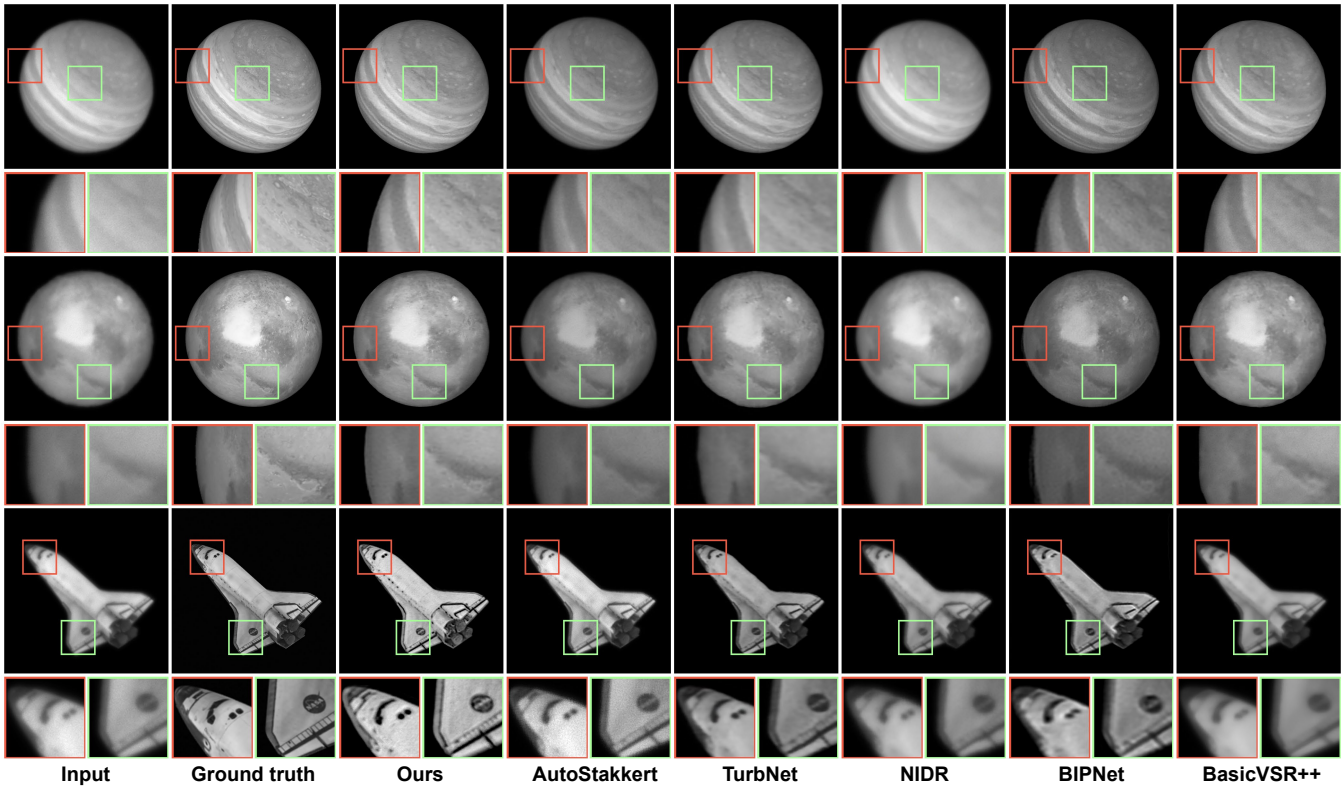

Figure 4: Qualitative comparisons on synthetic data among our method, a representative planetary imaging software AutoStakkert, and several state-of-the-art learning-based methods that solve the closest problems including TurbNet (Mao et al. 2022), NIDR (Li et al. 2021), BIPNet (Dudhane et al. 2022), and BasicVSR++ (Chan et al. 2022).

$\beta_2 = 0.999$) with a constant learning rate of 10^{-4} during training. We add an instance normalization (Ulyanov, Vedaldi, and Lempitsky 2016) layer and a `ReLU` activation function after each convolution layer.

Experiments

Evaluation on synthetic data. We compare our method to a representative planetary imaging software AutoStakkert, along with several state-of-the-art learning-based methods that solve the closest problems, including TurbNet, NIDR (Li et al. 2021), BIPNet (Dudhane et al. 2022), and (Chan et al. 2022). Note that for a fair comparison, we fix the number of input frames to 12, and retrain all learning-based methods (Mao et al. 2022; Li et al. 2021; Dudhane et al. 2022; Chan et al. 2022) using our dataset. Besides, for TurbNet (Mao et al. 2022), we use it to process the input frames in a one-by-one manner to get multiple output frames, and select the best frame as its result; and for BasicVSR++ (Chan et al. 2022), since it outputs multiple frames, we also choose the best frame as its result. In this way, the performance of TurbNet (Mao et al. 2022) and BasicVSR++ (Chan et al. 2022) is optimized. For AutoStakkert, we utilize its built-in tools for post-processing to recover high-spatial-frequency details.

Visual quality comparisons are shown in Figure 4[6]. It can be seen that our method exhibits the closest approximation to the ground truth. Besides, we can see that our method can not only handle the planets, but also perform well on other objects in the outer space (*e.g.*, the spacecraft shown in the bottom row of Figure 4), which shows that our method has broad applicability. To evaluate the results quantitatively, we adopt two frequently-used metrics including PSNR and SSIM. Results are shown in Table 1. Our model consistently outperforms the compared methods on all metrics.

Evaluation on real data. In order to demonstrate the robustness and generalization capabilities of our method, we capture some planetary images using a Celestron C11 telescope and a ZWO ASI 290 camera, and collect some planetary images from the Internet[7]. For each scenario, we prepare 12 frames as the input of PlaNet and the compared methods (AutoStakkert, TurbNet (Mao et al. 2022), NIDR (Li et al. 2021), BIPNet (Dudhane et al. 2022), and BasicVSR++ (Chan et al. 2022)). As shown in Figure 5[8], our method achieves superior image quality across various noise levels and turbulence magnitudes, producing clean backgrounds and sharp edges. In contrast, the other methods fail

[6]Additional results can be found in the supplementary material.
[7]http://www.skyimaging.com/astronomy-videos.php
[8]Additional results are provided in the supplementary material.

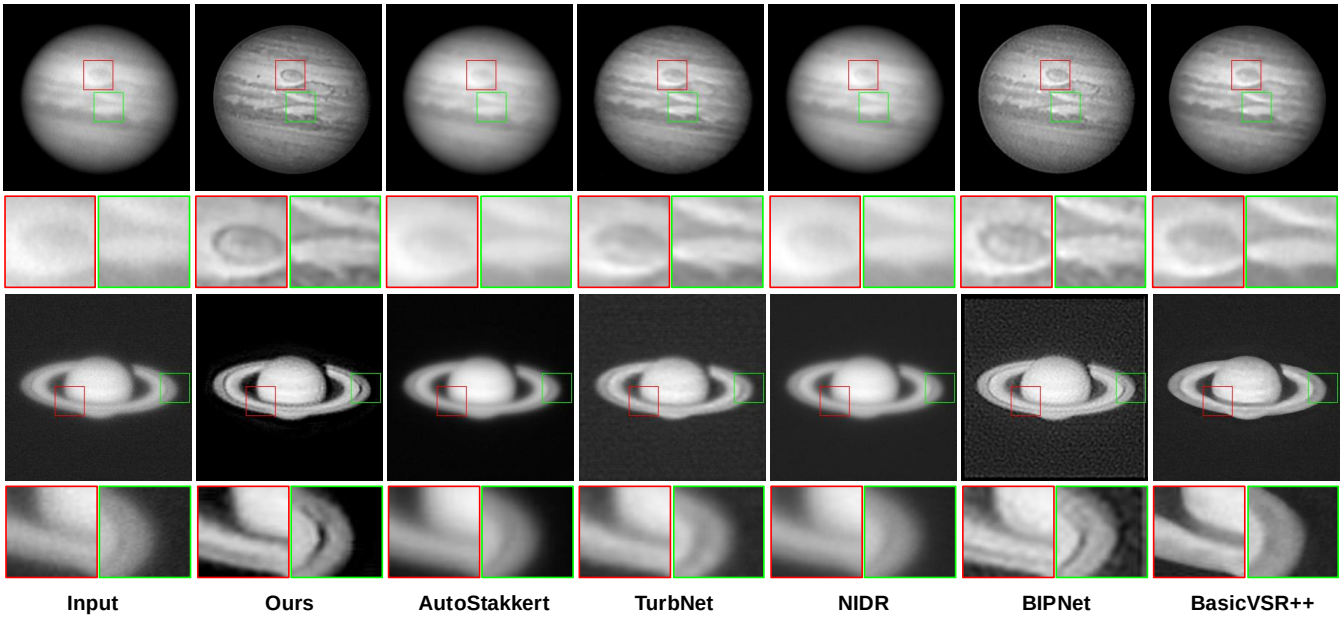

Figure 5: Qualitative comparisons on real data among our method, a representative planetary imaging software AutoStakkert, and several state-of-the-art learning-based methods that solve the closest problems including TurbNet (Mao et al. 2022), NIDR (Li et al. 2021), BIPNet (Dudhane et al. 2022), and BasicVSR++ (Chan et al. 2022).

Table 1: Quantitative evaluation results on synthetic data among our method, a representative planetary imaging software AutoStakkert, and several state-of-the-art learning-based methods.

Method	PSNR	SSIM
AutoStakkert	24.41	0.836
TurbNet (Mao et al. 2022)	26.48	0.887
NIDR (Li et al. 2021)	23.86	0.833
BIPNet (Dudhane et al. 2022)	23.11	0.898
BasicVSR++ (Chan et al. 2022)	26.07	0.835
Ours	**27.78**	**0.901**

to address real atmospheric turbulence effectively, yielding blurry results, or introduce artifacts and background noise.

Ablation study. We validate our design choices through ablation studies in Table 2. First, we assess edge-based supervision by excluding the edge map of the average frame (W/o edge-based supervision), underscoring the edge map's utility. Next, we evaluate the OBI module by comparing it to a model that applies edge-based supervision after feature map aggregation (W/o OBI) and the OBG module against one using average pooling (W/o OBG), confirming the soundness of both designs. We also test multi-scale edge supervision by calculating L_{dec} only at level $j = 1$ (W/o multi-scale L_{dec}), which results in a significant performance decline. Furthermore, moving the OBG module ahead of the backbone network (Frame instead of feature aggr.) demonstrates that frame aggregation fails to utilize information fully, leading to reduced performance. Overall, our complete model outperforms all variations, validating our design choices.

Table 2: Quantitative evaluation of ablation study.

	PSNR	SSIM
W/o edge-based supervision	27.25	0.893
W/o OBI	27.21	0.898
W/o OBG	25.90	0.895
W/o multi-scale L_{dec}	26.40	0.897
Frame instead of feature aggr.	23.37	0.891
Our complete model	**27.78**	**0.901**

Conclusion

We present PlaNet, a learning-based method designed to reduce atmospheric turbulence in planetary images, encompassing not only planets but also other celestial objects like spacecraft, utilizing flexible input frames. Our vertical distance-aware turbulence simulation pipeline enables the large-scale synthesis of realistic planetary images while maintaining cost-effectiveness. To address the challenge of limited background data, we incorporate an edge-based supervision strategy into PlaNet. Experimental results demonstrate that our approach outperforms existing methods on both synthetic and real-world planetary images.

Limitations. Due to the challenges in data acquisition, the real data presented in this paper only demonstrates validation results on planets, not including other objects in outer space. Besides, the edge-based supervision strategy might also be less effective in situations where the background is not scarce or the edges are not clear.

Acknowledgements

This work is supported by the National Natural Science Foundation of China under Grant Nos. 62136001, 62088102, and 62276007. We extend our gratitude to Junda Liu and Qi Liu for providing real-world planetary images.

References

Anantrasirichai, N.; Achim, A.; Kingsbury, N. G.; and Bull, D. R. 2013. Atmospheric turbulence mitigation using complex wavelet-based fusion. *IEEE Transactions on Image Processing*, 22(6): 2398–2408.

Chak, W. H.; Lau, C. P.; and Lui, L. M. 2018. Subsampled turbulence removal network. *arXiv preprint arXiv:1807.04418*.

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*.

Chimitt, N.; and Chan, S. H. 2020. Simulating anisoplanatic turbulence by sampling intermodal and spatially correlated Zernike coefficients. *Optical Engineering*, 59(8): 083101–083101.

Chimitt, N.; Zhang, X.; Mao, Z.; and Chan, S. H. 2022. Real-time dense field phase-to-space simulation of imaging through atmospheric turbulence. *IEEE Transactions on Computational Imaging*, 8: 1159–1169.

Dobke, B. M.; Johnston, D. E.; Massey, R.; High, F. W.; Ferry, M.; Rhodes, J.; and Vanderveld, R. A. 2010. Astronomical image simulation for telescope and survey development. *Publications of the Astronomical Society of the Pacific*, 122(894): 947.

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*.

Fazlali, H.; Shirani, S.; BradforSd, M.; and Kirubarajan, T. 2022. Atmospheric Turbulence Removal in Long-Range Imaging Using a Data-Driven-Based Approach. *International Journal of Computer Vision*, 130(4): 1031–1049.

Feng, B. Y.; Xie, M.; and Metzler, C. A. 2022. Turbugan: An adversarial learning approach to spatially-varying multiframe blind deconvolution with applications to imaging through turbulence. *IEEE Journal on Selected Areas in Information Theory*, 3(3): 543–556.

Fried, D. L. 1978. Probability of getting a lucky short-exposure image through turbulence. *JOSA*, 68(12): 1651–1658.

Gao, J.; Anantrasirichai, N.; and Bull, D. 2019. Atmospheric turbulence removal using convolutional neural network. *arXiv preprint arXiv:1912.11350*.

Hardie, R. C.; Power, J. D.; LeMaster, D. A.; Droege, D. R.; Gladysz, S.; and Bose-Pillai, S. 2017. Simulation of anisoplanatic imaging through optical turbulence using numerical wave propagation with new validation analysis. *Optical Engineering*, 56(7): 071502–071502.

Hirsch, M.; Sra, S.; Schölkopf, B.; and Harmeling, S. 2010. Efficient filter flow for space-variant multiframe blind deconvolution. In *Proc. of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 607–614.

Hu, J.; Shen, L.; and Sun, G. 2018. Squeeze-and-excitation networks. In *Proc. of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7132–7141.

Jin, D.; Chen, Y.; Lu, Y.; Chen, J.; Wang, P.; Liu, Z.; Guo, S.; and Bai, X. 2021. Neutralizing the impact of atmospheric turbulence on complex scene imaging via deep learning. *Nature Machine Intelligence*, 3(10): 876–884.

Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Lau, C. P.; Lai, Y. H.; and Lui, L. M. 2019. Restoration of atmospheric turbulence-distorted images via RPCA and quasiconformal maps. *Inverse Problems*, 35(7): 074002.

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. 2020. Image reconstruction of static and dynamic scenes through anisoplanatic turbulence. *IEEE Transactions on Computational Imaging*, 6: 1415–1428.

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.

Mei, K.; and Patel, V. M. 2023. Ltt-gan: Looking through turbulence by inverting gans. *IEEE Journal of Selected Topics in Signal Processing*.

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.

Rai, S. N.; and Jawahar, C. 2022. Removing atmospheric turbulence via deep adversarial learning. *IEEE Transactions on Image Processing*, 31: 2633–2646.

Roggemann, M. C.; Welsh, B. M.; and Hunt, B. R. 1996. *Imaging through turbulence*. CRC press.

Ronneberger, O.; Fischer, P.; and Brox, T. 2015. U-net: Convolutional networks for biomedical image segmentation. In *Proc. of Medical Image Computing and Computer*, 234–241.

Schwartzman, A.; Alterman, M.; Zamir, R.; and Schechner, Y. Y. 2017. Turbulence-induced 2d correlated image distortion. In *Proc. of International Conference on Computational Photography*.

Su, J.; Xu, B.; and Yin, H. 2022. A survey of deep learning approaches to image restoration. *Neurocomputing*, 487: 46–65.

Tokovinin, A. 2002. Measurement of seeing and the atmospheric time constant by differential scintillations. *Applied Optics*, 41(6): 957–964.

Ulyanov, D.; Vedaldi, A.; and Lempitsky, V. 2016. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*.

Vyhnalek, B. E. 2017. Path profiles of Cn2 derived from radiometer temperature measurements and geometrical ray tracing. In *Free-Space Laser Communication and Atmospheric Propagation XXIX*, volume 10096, 386–396.

Woo, S.; Park, J.; Lee, J.-Y.; and Kweon, I. S. 2018. Cbam: Convolutional block attention module. In *Proc. of European Conference on Computer Vision*.

Yasarla, R.; and Patel, V. M. 2020. Learning to restore a single face image degraded by atmospheric turbulence using CNNs. *arXiv preprint arXiv:2007.08404*.

Yasarla, R.; and Patel, V. M. 2021. Learning to restore images degraded by atmospheric turbulence using uncertainty. In *Proc. of International Conference on Image Processing*, 1694–1698. IEEE.

Zhang, K.; Ren, W.; Luo, W.; Lai, W.-S.; Stenger, B.; Yang, M.-H.; and Li, H. 2022a. Deep image deblurring: A survey. *International Journal of Computer Vision*, 130(9): 2103–2130.

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

Zhou, C.; Teng, M.; Han, Y.; Xu, C.; and Shi, B. 2021. Learning to dehaze with polarization. *Advances in Neural Information Processing Systems*, 34: 11487–11500.

Zhou, C.; Zhao, H.; Han, J.; Xu, C.; Xu, C.; Huang, T.; and Shi, B. 2020. Unmodnet: Learning to unwrap a modulo image for high dynamic range imaging. *Advances in Neural Information Processing Systems*, 33: 1559–1570.

Zhu, X.; and Milanfar, P. 2011. Stabilizing and deblurring atmospheric turbulence. In *Proc. of International Conference on Computational Photography*, 1–8.

Zhu, X.; and Milanfar, P. 2012. Removing atmospheric turbulence via space-invariant deconvolution. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(1): 157–170.

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 4 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

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

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/

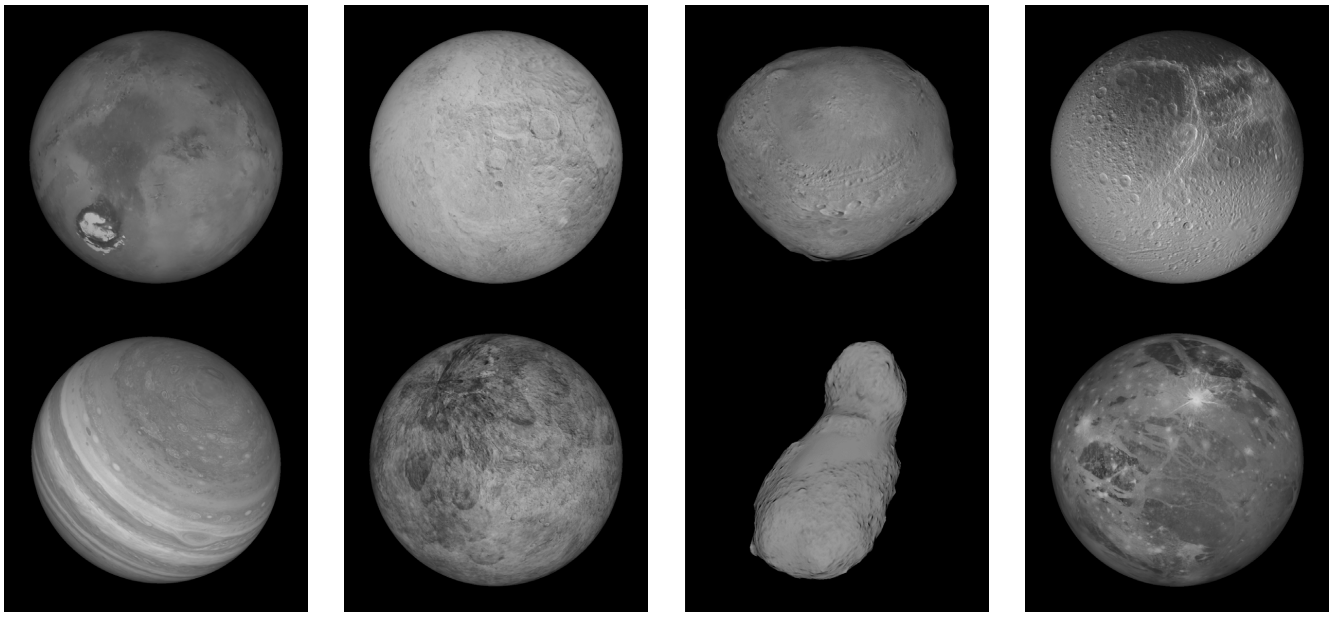

Figure 6: Visual representation of four types of celestial objects included in our synthetic dataset. From left to right, the images show two examples for each type: planets (Mars and Jupiter), dwarf planets (Eris and Makemake), asteroids (Bennu and Itokawa), and satellites (Dione and Ganymede).

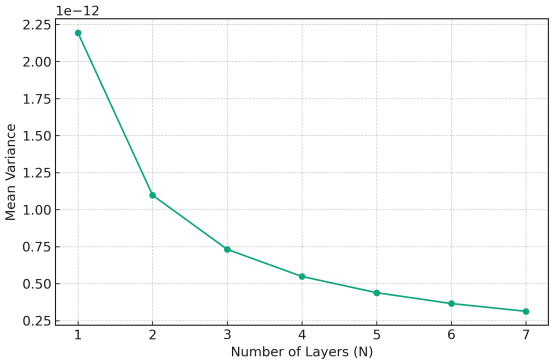

Figure 7: Mean variance of atmospheric turbulence magnitude as a function of the number of discrete atmospheric layers (N) used in the simulation. Each layer is configured with distinct values of D_i and r_{0_i}, contributing to the overall turbulence profile captured by our pipeline. Our choice $N = 3$ 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.

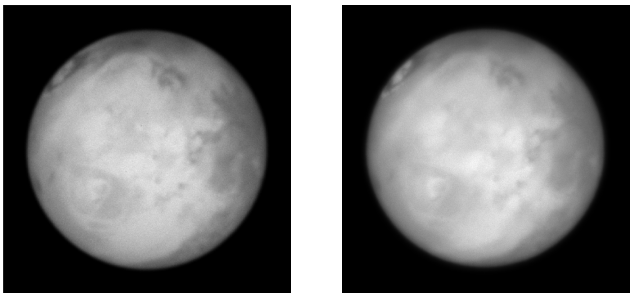

Figure 8: Comparison between PhoSim (Peterson et al. 2015) and our simulation method, demonstrating that our simulation method is a close approximation to the physically accurate photon simulator PhoSim (Peterson et al. 2015).

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	$0.6844\ \alpha$
2	3860	$0.3941\ \alpha$
3	1263	$0.0895\ \alpha$

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 as you mentioned. 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, corresponding to Footnote 6 of the main paper.

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.

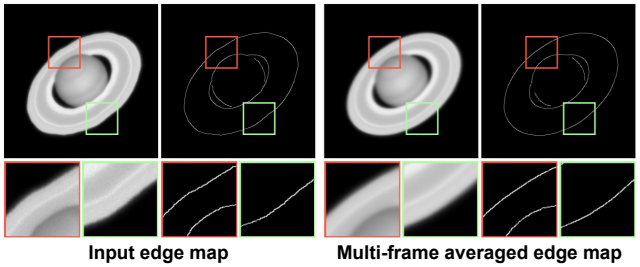

Figure 9: Visualization of the edge map showing the impact of atmospheric turbulence on individual frames and the efficacy of the averaging operation in mitigating distortion. Each frame shows its own unique distortions, which are significantly reduced in the stabilized image because of the zero-mean 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 7 in the main paper. Similarly, for real data, Figure 11 is aligned with Footnote 9 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*.

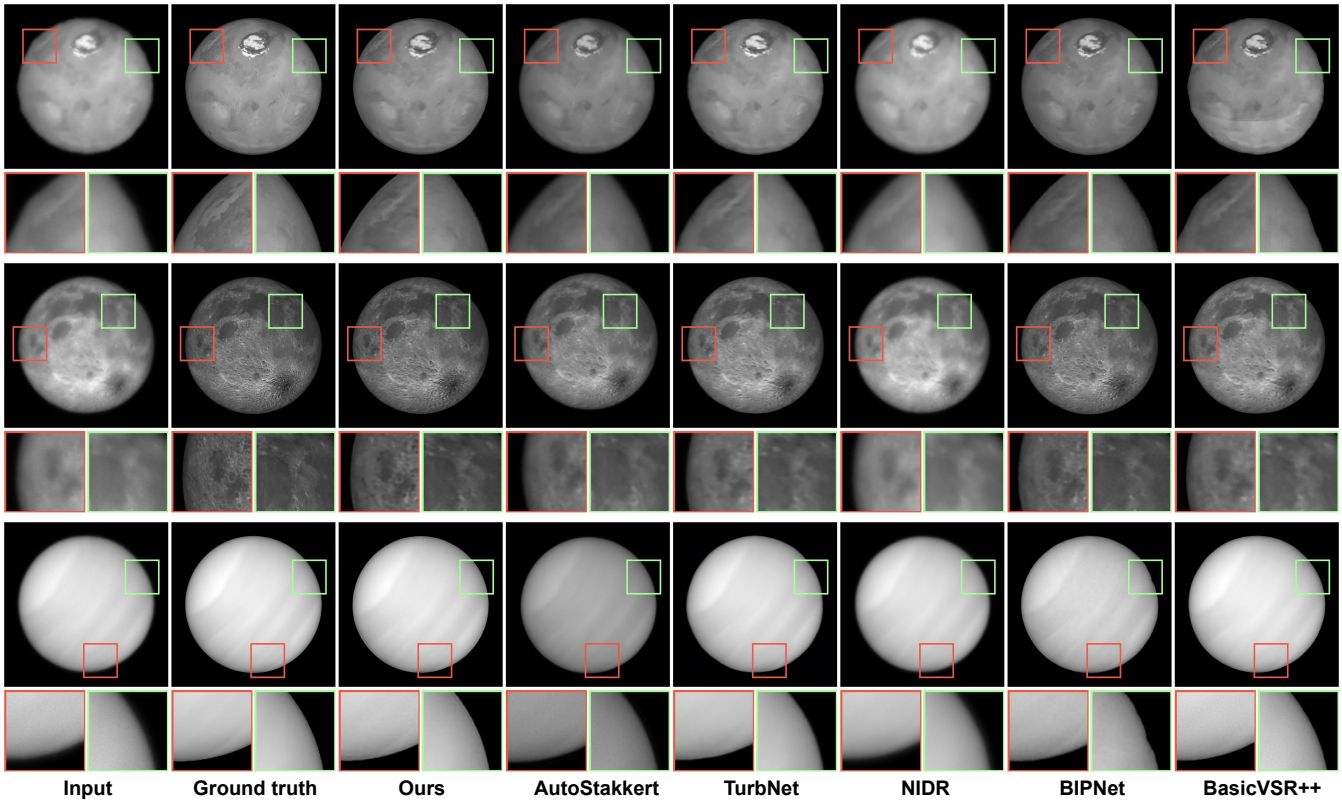

Figure 10: Qualitative comparisons on synthetic data among our method, a representative planetary imaging software AutoStakkert, and several state-of-the-art learning-based methods that solve the closest problems including TurbNet (Mao et al. 2022), NIDR (Li et al. 2021), BIPNet (Dudhane et al. 2022),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

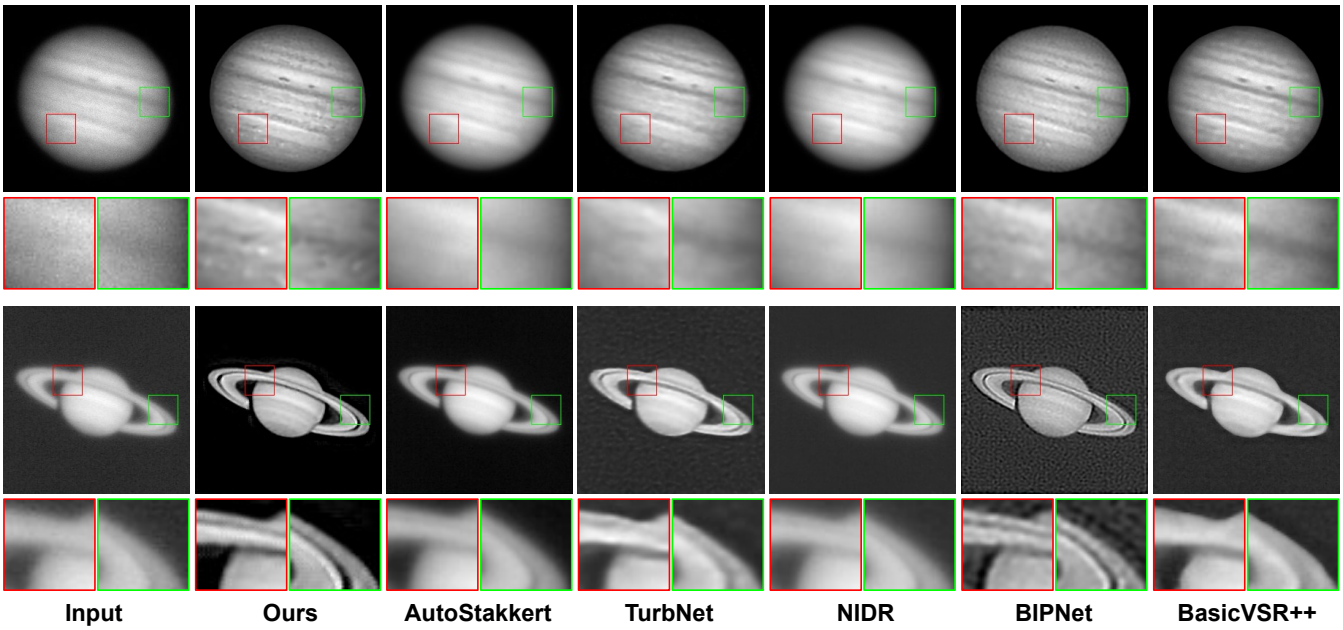

Figure 11: Qualitative comparisons on real data among our method, a representative planetary imaging software AutoStakkert, and several state-of-the-art learning-based methods that solve the closest problems including TurbNet (Mao et al. 2022), NIDR (Li et al. 2021), BIPNet (Dudhane et al. 2022), and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