

# Automatic Detection of Stationary Gravity Waves in the Venus' Atmosphere Using Deep Generative Models

\*Minori Narita<sup>1</sup>, Daiki Kimura<sup>2</sup>, Atsushi Yamazaki<sup>3</sup>, Takeshi Imamura<sup>1</sup>

1. The University of Tokyo, 2. IBM Research AI, 3. ISAS/JAXA

Venus is covered with thick clouds of sulfuric acid, and the entire upper atmosphere rotates much faster than the planet itself. Recently, interhemispheric large bow-shaped structures, remaining fixed above the highland in contrast to background wind fields, were observed in long-infrared and ultraviolet images by Venus Orbiter Akatsuki. These bow-shaped structures are said to be the result of atmospheric gravity waves generated in the lower atmosphere and propagate upwards to the cloud top, which has a significant influence on the Venus' atmospheric system. Not a few bow-shaped, stationary features indicative of topographic gravity waves have been reported so far. However, to unravel where and when these structures appear frequently, we need to analyze the phenomena comprehensively, using much more data. Detection of such characteristic cloud features in images has been done by human eye in Venus research, but to detect many more cases efficiently from massive amounts of image data, we need a framework to do it automatically and robustly. Moreover, most of the bow-shaped structures in ultraviolet images are small, which is almost impossible to find out every case by our eyes.

In this study, we propose a novel approach that detects these stationary features using a deep generative model that has deep neural networks (deep learning). There are various types of such deep models nowadays; we used a variational auto-encoder (VAE) because it has reported good results in dimensionality reduction fields and been a well-known method for image reconstruction. The input and the output of VAE are set to be the same, and the network is divided into two parts, which are the encoder that converts an image to latent representation, and the decoder that reconstructs an image from given latent variables. The detection of stationary features can be seen as an anomaly detection task, which treats images that have stationary features as the anomaly class and the normal cloud image as the normal class.

We used ultraviolet (283 nm) images for this analysis. Firstly, we performed photometric correction using Minnaert Law, and averaged several images taken successively in 24 hours to highlight structures that remains still. Then we extracted features that have scales smaller than 6 degrees by high-pass filtering. Lastly, we extracted image patches from these images. Each patch has the size of 24 degrees both in latitude and longitude. We used images taken in 25 days in 2016 and 2017.

The analysis procedure is as follows. First, we train this network using only normal cloud images that do not include stationary features to let it learn its latent features. When testing, we input test images composed of both normal and anomaly (stationary feature) images, and detect anomalies based on the reconstruction loss that is calculated by the sum of the pixel-wise distance between the original image and the reconstructed image that is output from the last layer in the network. The network has five convolutional layers in the encoder and the decoder, respectively. The latent space has 32 dimensions. We used 4,000 normal cloud images as the training data, and 1,000 normal and 255 anomaly images as the test data.

We evaluated our method by calculating the area under the curve (AUC) of the receiver operating characteristic (ROC) curve. As a result, the AUC was 0.93 (maximum is 1.00), which shows that the

proposed method successfully detects stationary features. The reconstruction error of a stationary structure image is large because the network cannot reconstruct such bow-shaped structure properly. Our method can be easily applied not only to Venus but also various planetary images to detect characteristic structures.

Keywords: Venus, Deep Learning, Generative Model, Anomaly Detection, Image