

# Lunar elemental analysis and mapping of Chandrayaan-2 Large Area Soft X-ray Spectrometer Data

**Abstract**—The Chandrayaan-2 mission's CLASS (Chandrayaan-2 Large Area Soft X-ray Spectrometer) instrument has provided a wealth of data for understanding the elemental composition of the lunar surface. In this study, we analyzed extensive CLASS instrument datasets to determine elemental abundances across various lunar regions using the XSPEC software. These abundances were mapped onto a lunar albedo map to correlate elemental distribution with surface reflectance properties. Additionally, we explored sub-pixel resolution enhancement techniques for these elemental abundance maps, leveraging machine learning methodologies to improve spatial resolution from 12.5 km to 2 km. Our approach highlights the potential of integrating traditional spectrometry analysis with modern machine learning techniques to derive finer spatial details, advancing our understanding of the Moon's compositional heterogeneity and geological evolution.

## I. INTRODUCTION

The Chandrayaan-2 mission, launched by the Indian Space Research Organisation (ISRO), represents a significant milestone in lunar exploration. Equipped with state-of-the-art instruments, the mission aims to expand our understanding of the Moon's composition, surface processes, and potential for future resource utilization. One of its key payloads, the Chandrayaan-2 Large Area Soft X-ray Spectrometer (CLASS), was designed to perform X-ray fluorescence (XRF) spectroscopy to determine the elemental composition of the lunar surface with unprecedented accuracy.

This report focuses on the analysis of extensive data collected by the CLASS instrument, highlighting the processes and methodologies employed to determine elemental abundances. Using advanced spectral analysis tools like XSPEC, these elemental abundances were mapped onto the lunar albedo map to explore surface correlations. Furthermore, to address the challenges posed by the instrument's native spatial resolution, this work leverages machine learning techniques to achieve a resolution enhancement to 2 km. The insights gained from this study not only enhance our understanding of lunar surface heterogeneity but also lay the groundwork for future advancements in planetary remote sensing and data analysis.

## II. THEORETICAL MODELING

Theoretical modeling allowed us to better understand the XRF data because it takes into account accurately the complex interaction between X-rays and the material. Incorporating factors like the scattered continuum, mass attenuation coefficients, and fluorescent X-ray intensity lets us account for the influences of the sample's composition and geometry, as

well as the incident X-ray energy. This modelling was done as below.

### A. Fluorescent X-Ray Intensity

The total fluorescent X-ray intensity is modelled as summation of primary fluorescent x-ray intensity ( $I_1(ip)$ ) and secondary fluorescent x-ray intensity ( $I_2(ip)$ ) [1]. Where ( $I_1(ip)$ ) and ( $I_2(ip)$ ) are given by,

$$I_1(ip) = \frac{1}{\sin \Psi} \int_{\lambda_m}^{\lambda_i} \frac{Q_{ip}(\lambda) I_0(\lambda)}{\frac{\mu(\lambda)}{\rho} / \sin \Phi + \frac{\mu(ip)}{\rho} / \sin \Psi} d\lambda \quad (1)$$

$$I_2(ip) = \frac{1}{2 \sin \Psi} \sum_{jq} \int_{\lambda_m}^{\lambda_e} \frac{Q_{jq}(\lambda) Q_{ip}(jq) I_0(\lambda)}{\frac{\mu(\lambda)}{\rho} / \sin \Phi + \frac{\mu(ip)}{\rho} / \sin \Psi} d\lambda \times \left\{ \frac{\sin \Psi}{\frac{\mu(ip)}{\rho}} \log \left( 1 + \frac{\frac{\mu(ip)}{\rho} / \sin \Psi}{\frac{\mu(jq)}{\rho}} \right) + \frac{\sin \Phi}{\frac{\mu(\lambda)}{\rho}} \log \left( 1 + \frac{\frac{\mu(\lambda)}{\rho} / \sin \Phi}{\frac{\mu(jq)}{\rho}} \right) \right\} \quad (2)$$

Emission probability of the fluorescent x-rays, p-line of the element i, ( $Q_{ip}(\lambda)$ ) is given by

$$Q_{ip}(\lambda) = \frac{\mu_i(\lambda)}{\rho_i} W_i K_i \omega_i R_p^i \quad (3)$$

$$Q_{ip}(jq) = \frac{\mu_i(jq)}{\rho_i} W_i \omega_i R_p^i K_i \quad (4)$$

and other terms in the expression are in the table

### B. Total Mass Attenuation Coefficient

The mass attenuation coefficient ( $\mu/\rho$ ) represents the extent to which a material attenuates (reduces the intensity of) radiation per unit mass. This property is intrinsic to the material and reflects how radiation, such as X-rays or gamma rays, is absorbed and scattered as it travels through the substance. It is a quantification of material's ability to interact with and weaken the radiation.

For a composite material ( $\mu/\rho$ ) in units of ( $\text{cm}^2/\text{g}$ ) it is given by a weighted average with weights  $W_j$  single elements as ( $(\frac{\mu}{\rho})_i$ ) [2]

$$\frac{\mu}{\rho} = \sum_j W_j \left( \frac{\mu}{\rho} \right)_j \quad (5)$$

Notation	Description
$\phi$	The angle made by the incident beam with the sample surface.
$\psi$	The angle made by the emergent beam with the sample surface.
$I_0(\lambda)$	The intensity of x-ray of wavelength $\lambda$ .
$I(ip)$	The intensity of the x-ray of $p$ -line emitted from element $i$ , where $i$ refers to the element and $p$ refers to the spectral line.
$\mu(\lambda)$	The linear absorption coefficient of the specimen for the x-ray of wavelength $\lambda$ .
$\mu(ip)$	The linear absorption coefficient of the specimen for the characteristic x-ray, $p$ -line of the element $i$ .
$\mu_j(\lambda)$	The linear absorption coefficient of the element $j$ for the x-ray of wavelength $\lambda$ .
$\mu_j(ip)$	The linear absorption coefficient of element $j$ for the characteristic x-ray, $p$ -line of element $i$ .
$\rho$	The density of the specimen.
$\rho_i$	The density of element $i$ .
$W_i$	The weight fraction of element $i$ in the specimen.
$\omega_i$	Fluorescent yield of element $i$ .
$R_p^i$	The intensity fraction of the $ip$ -line in the characteristic x-ray series which $ip$ -line belongs to.
$K_i$	The absorption jump of element $i$ .
$\lambda_m$	The minimum wavelength of exciting fluorescent radiation.
$\lambda_e$	The wavelength of the absorption edge of element $i$ .

TABLE I: Notations used in equations

where

$$\sum_j W_j = 1 \quad (6)$$

### C. Scatter Continuum

The total scattering cross section ( $\sigma_{\text{total}}$ ) is Defined as summation of scattering cross sections due to Rayleigh scattering ( $\sigma_{\text{Rayleigh}}$ ) and Crompton Scattering ( $\sigma_{\text{Crompton}}$ )

$$\sigma_{\text{total}} = \sigma_{\text{Rayleigh}} + \sigma_{\text{Crompton}} \quad (7)$$

where the value of total Rayleigh (elastic) scattering cross-section ( $\sigma_{\text{Rayleigh}} = \sigma_R$ ) is given by [3]:

$$\sigma_R = \frac{r_0^2}{2} \int_{\theta_{\min}}^{\theta_{\max}} (1 + \cos^2 \theta) |F(x, Z)|^2 2\pi \sin \theta d\theta \quad (8)$$

Where:

- $\sigma_R$  is the elastic scattering cross section,
- $F(x, Z)$  is the atomic form factor,
- $r_0$  is the radius of electron, and
- $\theta$  is the scattering angle

and the intensity of elastically scattered spectrum ( $I_R$ ) from an infinitely thick sample can be computed by

$$I_R = \int \frac{I_0(E) \sigma_R(E)}{\mu(E) [\text{cosec } \phi + \text{cosec } (\psi)]} dE \quad (9)$$

Where:

- $\phi$  is the incident angle,
- $\psi$  is the emission angle,
- $E$  is Energy,
- $\sigma_R(E)$  is the elastic scattering cross section as a function of energy, and
- $\mu(E)$  is the linear attenuation coefficient as a function of energy

## III. ALGORITHM

### A. Data

CLASS is operated continuously on the lit side of the moon to record the XRF photons excited by the lunar surface when the solar flares hit the surface of the moon. While night side observations( 5min) every orbit are used to measure the background in the lunar orbit. CLASS instrument is designed to detect this X-ray emission in the 0.8-15 keV energy range. Since it is important to identify the spectrum solar flare-produced Xrays that trigger the generation of XRF lines, a companion payload on chandrayaan-2, the Xray solar monitor(XSM) observes the sun simultaneously to measure solar flares X-ray spectrum. These XRF spectrum are composed of a scattered continuum and X-ray fluorescence lines along with a background continuum that is taken to be steady over many months.

The lunar surface is constantly exposed to Galactic Cosmic Rays (GCRs), solar wind, and occasional Solar Energetic Particles (SEPs), ranging from a few keV to GeV. Low-energy particles deposit energy in SCDs as a continuum, while high-energy particles pass through. CLASS efficiently distinguishes particle events (spanning hundreds of pixels) from X-ray events (limited to 2–3 pixels). Counts above  $\approx 7$  keV predominantly arise from particles, with notable X-ray peaks, such as Cu (8.05 keV) and the Al filter (1.47 keV).

The 7–16 keV light curve over nine months shows monthly background enhancements due to geotail passage. Sunlit spectra contain XRF lines (e.g., O-K $\alpha$ , Mg, Al, Si), while night-side observations reveal background variations. The background spectra for XRF analysis are selected based on low levels of particle flux and remain stable over months.

The spectral analysis involves several key steps. First, the background spectrum appropriate for the observation,  $B(E)$ .  $B(E)$ , is constructed. Next, the incident solar X-ray spectrum corresponding to CLASS observations  $I_0(E)$ , is derived using XSM data. This is followed by determining the scattered spectrum of the solar flare reflected from the lunar surface  $I_R(E)$ . From the spectral analysis, X-ray line fluxes of various elements are extracted. Finally, these XRF line fluxes are

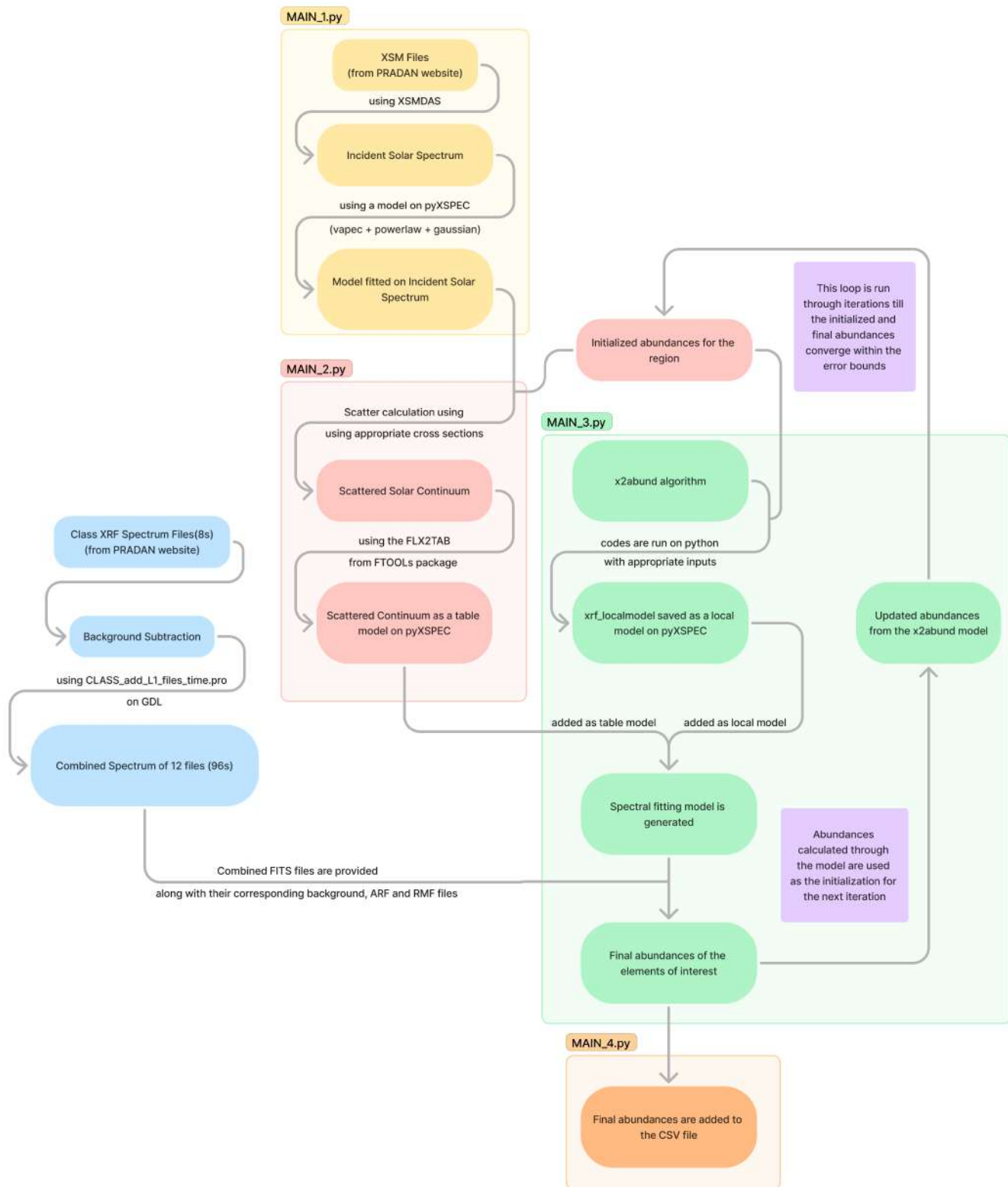


Fig. 1: Flowchart of the algorithm used for deriving elemental abundances from XRF spectra.

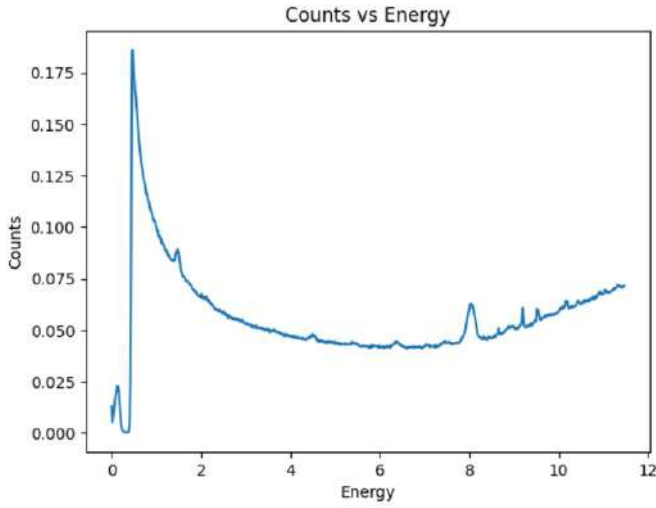


Fig. 2: Intensity of background radiation

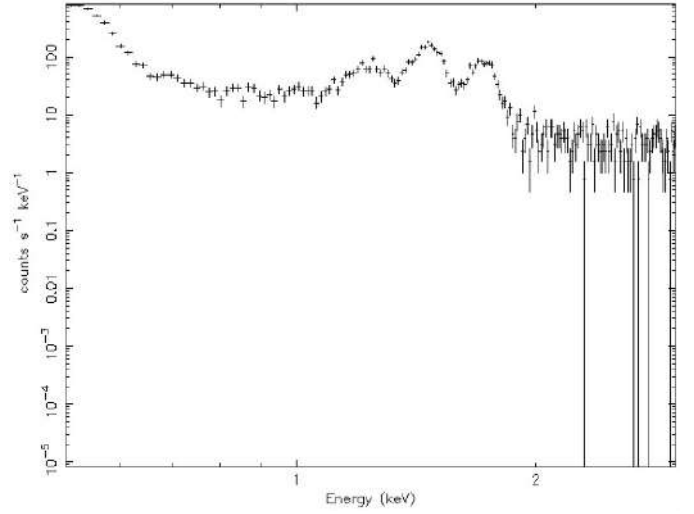


Fig. 4: Combined XRF Spectrum Data plotted against Energy on XSPEC

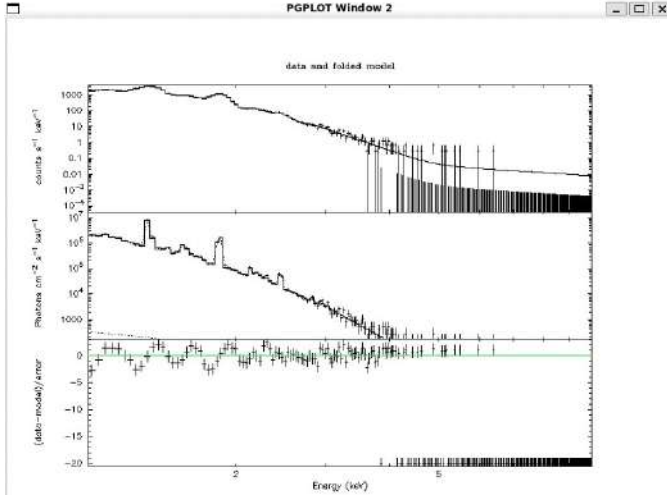


Fig. 3: Solar spectrum data fitted with the Solar Continuum model

converted into elemental abundances using the `x2abundance` model. Background emission comes from cosmic X-rays, secondary particles, and X-rays produced by high-energy particles interacting with the instrument. Energetic particles can also leave partial signals in the detector, copying X-ray-like patterns. The changing particle environment causes variations in the background, which must be accurately understood in order to analyze the XRF line fluxes.

The Moon experiences two main particle environments in a lunar month:

- 1) Solar Wind and Cosmic Rays ( $\approx 24$  Days): Steady background caused by solar wind and cosmic rays with energies from keV to GeV.
- 2) Earth's Geotail ( $\approx 6$  Days): Background affected by energetic electrons (1–several keV) as the Moon passes through the geotail near the full moon phase.

Sudden particle bursts from solar events can cause sharp

increases in CLASS counts, unrelated to solar X-rays, leading to temporary changes in the background.

### B. Solar Spectrum

The incident solar spectra  $I_0(E)$  for the observed CLASS timings were derived from XSM data, providing key inputs for the analysis of lunar XRF spectra. Spectral analysis was carried out using the XSMIDAS software developed by ISRO. This software allowed us to create the appropriate spectrum files for 96-second intervals corresponding to the XRF spectrum being analyzed more easily, using the `xsmgenspec` command.

The fitting of solar spectra was performed in XSPEC on the energy from 1.2 to 7.2 KeV interval using `vapec` with an additional powerlaw component as the model, with initial parameters set to coronal abundances based on Feldman [?]. This approach ensures a robust representation of the observed solar X-ray intensity. Other models, such as using only `vapec` or `vvapec`, were also tested but resulted in similar or worse fits.

The solar spectral model obtained from this fitting process was then used in `x2abundance`. This model was essential for computing the lunar XRF line intensities, which are critical for determining elemental abundances on the lunar surface. Additionally, the scattered component of solar X-rays, reflected from the lunar surface, was calculated using the derived solar spectrum model, ensuring accurate interpretation of the lunar XRF data.

### C. Calculation of Scatter Continuum ( $I_R(E)$ )

We used XSM data to obtain the solar spectrum and spacecraft telemetry for the observing angles. Starting with an apriori composition based on the lunar highland and mare regions, we iteratively modeled the scattered component.

We adopted the feldspathic terrane composition from Koro-tev et al. [2003] [?] for the apriori model. The scattered

spectrum was then used as a component in spectral fitting to derive XRF line fluxes. The determined abundances were used in further iterations to refine the scattered spectrum, which was incorporated into the spectral fitting process. Convergence was generally achieved within two iterations, with line fluxes stabilizing within error bounds.

We used the FFAST dataset derived from the NIST database to find the  $\frac{\mu}{\rho}$  values and the  $\frac{\sigma}{\rho}$  values corresponding to each element.

Weighted sum of  $\frac{\sigma}{\rho}$  values using all elements is determined to calculate the equivalent  $\frac{\sigma}{\rho}$  values and similarly a weighted sum of  $\frac{\mu}{\rho}$  values are determined to calculate the equivalent  $\frac{\mu}{\rho}$  values for a given energy bin. Finally, we calculate the scatter continuum corresponding to each energy bin, generating the scatter continuum from the solar continuum.

This scatter continuum is a function of the  $\frac{\mu}{\rho}$  values and  $\frac{\sigma}{\rho}$  values corresponding to the elements on the surface of the moon which majorly include Fe, Ti, Ca, Na, O, Mg, Al, Si. The values we have used are included in the following table to calculate the scatter continuum. These are the major elements found on the surface of the moon and are also detected by xray spectroscopy and thus these elements are considered.

Scatter continuum also depends on the weight fractions that are mentioned in 5. These weight fractions correspond to the abundances of particular elements on the surface of the moon.

#### D. x2abundance

We now describe the formulation of the XRF inversion algorithm, X2Abundance, which converts the observed XRF line intensities into elemental abundances. The algorithm employs analytical expressions based on the Fundamental Parameters (FP) method, assuming a thick, homogeneous, and flat sample. The expressions by Shiraiwa and Fujino [1966] [1], outlined in 1 and 2 are utilized to compute XRF line intensities from the sample. The key steps in x2abundance for determining elemental abundances are outlined below, with the sequence of execution of major steps summarized in Fig. 5.

Since the lunar surface's elemental composition is dominated by elements with XRF line energies in the range of 0.9–4.2 keV, the computation focuses on elements with K-shell atomic binding energies below 10 keV. Using fundamental atomic data, such as mass absorption coefficients and fluorescence yields, the code analytically calculates the XRF line intensities for a given matrix ( $M_j$ ), incident spectrum ( $I_0(E)$ ), and observational geometry.

Qualitative analysis of the XRF spectrum identifies the elements present in a sample. The set of elements and their corresponding weight fractions is referred to as the "matrix." Different combinations of weight fractions are generated for these elements, ensuring the total always sums to 1. For a given matrix  $M_j$ , the condition  $\sum W_i = 1$  is maintained, where  $W_i$  represents the weight fraction of each element.

To optimize computations for samples with many elements, constraints based on prior knowledge can limit the range of weight fractions. For example, if an element's abundance is known to exceed 0.6 in a sample containing four elements,

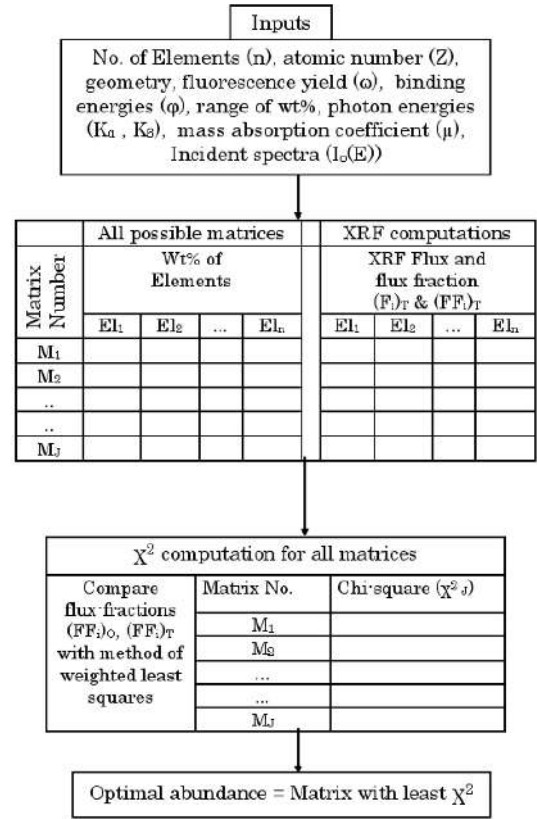


Fig. 5: Important steps involved in execution of x2abundance

its weight fraction is restricted to the range 0.6–0.99. The remaining three elements collectively vary between 0.01 and 0.4, significantly reducing the number of possible combinations.

The program calculates the line flux ( $F_i$ ) and flux fraction  $FF_i$ ) for each element in every matrix  $M_j$ .

Using the tools that have been discussed the methodology that we used to finally find the abundance corresponding to a single 96 seconds XRF spectrum fits file combined in a sliding window fashion has been discussed below. We explored various approaches to combine the 8s XRF spectrum FITS files provided by the CLASS instrument, which are available for download from the Pradan website. Three primary strategies were considered: the sliding window approach, the running check approach, and a combined approach (sliding + running). These methods are described below:

##### 1) Sliding Window Approach:

The sliding window approach involves combining adjacent FITS files in a sequential manner. For example, files 1-12 are combined together, then files 2-13, and so on. This approach ensures significant overlap between adjacent tracks, which is crucial as the satellite moves across different positions. This method optimizes the spatial coverage by ensuring smooth transitions between tracks.

##### 2) Running Check Approach:

In the running check approach, we sequentially combine

files and perform spectral fitting using three Gaussian functions and a continuum. The goal is to minimize the chi-squared value of the fit, ideally between 0.8 and 2. Files are combined until a satisfactory fit is achieved, and once a set of files has been selected, the next combination starts from the next available file. This approach helps in reducing processing time, but it may come at the cost of lower accuracy, particularly when subpixel resolution is critical.

### 3) Combined Approach (Sliding + Running):

The third approach merges the sliding window and running check approaches. Initially, files are combined using the sliding window method until a combination of three Gaussian functions and a continuum provides an optimal fit. Once a satisfactory fit is identified, the process continues with the next set of files, starting from the second file of the previous set. Although this approach incorporates the benefits of both previous methods, it did not lead to significant improvements in the fitting results and thus was not implemented for our final analysis.

The sliding window approach was chosen for our analysis as it ensures sufficient overlap between tracks, which is crucial for capturing accurate spectral features, particularly given the motion of the satellite. Although it is more time-consuming, this approach is essential for achieving high subpixel resolution. The running check approach, while faster, sacrifices some accuracy in the subpixel resolution, which is a key factor for our study. The combined approach did not show substantial improvements and was not pursued further. Further research can explore alternative strategies for combining FITS files to optimize processing time while maintaining the accuracy needed for high-resolution analysis.

The time interval of 96s was taken and the XSM spectrum file which are available on a day wise basis. Now we find the solar spectrum for that interval of 96 seconds using the XSMDAS software by ISRO. Only those files were considered that belong to the time intervals which were a subset of the GTI and GOES solar flare data in which we included solar flares of class B,C,M because A is a low intensity solar flare and is not enough for characteristic x-ray peaks to be differentiable from noise. X class solar flares were rejected because the instrument does not generate reliable data in these solar flares. The standard GTI include all times when the unocculted Sun is within the full FOV ( $\pm 38^\circ$ ) of XSM and also with the instrument health parameters within recommended ranges.

Now we use the discussion above for calculation of the scattered continuum from the spectrum using 9. Once the scattered continuum is determined. The scattered continuum is stored as a table model in XSPEC and a xrf\_localmodel derived from the NIST database and input abundances is used to calculate the XRF line component of the XRF spectrum. We used PyXspec(Python interface to xspec) for all the purposes. To derive the abundances we assumed an initial abundance that was a weighted sum of the fraction of area under highland regions and area under mare regions multiplied

by the corresponding average abundances in those regions for every element. Initialization using data from [?] and validated from [?] gave us good results. The background-subtracted XRF spectrum is fit using the scattered continuum and XRF lines with abundances as parameters for training. The inversion algorithm is used for training and fitting the abundance parameters. We also experimented with the number of iterations required to calculate the abundances, scattered continuum, and XRF line intensities. We found that one iteration was enough to obtain a low margin of error in the abundance values. Although performing two iterations was ideal, three iterations led to overfitting and overcomputation. We continued this process and analyzed files corresponding to approximately 75 days. We optimized our code And the results obtained have been discussed in the following section.

## IV. SUBPIXEL RESOLUTION

The task of sub-pixel resolution using overlapping tracks is modeled as semi-supervised deep spatial interpolation using a sparse dataset. This problem is approached using a hybrid architecture, consisting of a Graph Neural Network (GNN) and a Convolutional Neural Network (CNN), to capture local and global relationships inherent in the elemental abundance distribution. The discussion of the use of overlapping tracks for subpixel resolution and ultimately arriving at a  $2\text{ km} \times 2\text{ km}$  resolution map has been discussed in the following three subsections. Subsection A focuses on the introduction and problem formulation, it also talks about the difficulties in learning the mapping function. Subsection B covers the theoretical perspective of the implementation covering the theory behind the three-part methodology in detail. Subsection C focuses on the experimentation and the results obtained.

### A. Introduction and Problem Formulation

A specific point  $s$  on the lunar surface is defined by its 2D coordinate value  $(\text{lat}_s, \text{lon}_s) \in \mathcal{S}$ , where  $\mathcal{S}$  represents the entire (latitude, longitude) space, i.e.

$$\mathcal{S} = [90^\circ, -90^\circ] \times [-180^\circ, 180^\circ].$$

Each location  $s$  is associated with two components  $\mathbf{x}_s, \mathbf{y}_s$  belonging to  $\mathcal{X}, \mathcal{Y}$ .  $\mathcal{X}$  denotes the input feature space that is designed to capture both the spatial characteristics such as latitude, longitude, and surface characteristics, and  $\mathcal{Y}$  represents the output space of elemental abundances.

The goal is to learn the predictive mapping function

$$f : \mathcal{X} \rightarrow \mathcal{Y},$$

which captures the relationship between the input feature space  $\mathcal{X}$  and elemental abundances  $\mathcal{Y}$ , s.t.

$$f(\mathbf{x}_s) = \mathbf{y}_s \quad \forall s \in \mathcal{S}.$$

Learning the exact mapping function is challenging due to spatial heterogeneity across the lunar surface. Formally, this variability due to spatial heterogeneity can be expressed as

$$f(x_s) \neq f(x_{s'})$$

for  $s, s' \ni \|\mathbf{x}_s - \mathbf{x}_{s'}\|_2 > \epsilon$  for some  $\epsilon > 0$ . This task of learning  $f$  is further complicated by the sparsity and large spatial span of elemental abundances calculated from XRF spectrum.

Thus we try to model this problem of finding  $f$  as a non-convex optimization problem, and try to approximate  $f$  from a hypothesis class  $\mathcal{H}$  that is rich enough to capture both the local variations while maintaining consistency in the elemental distribution across the global domain. We hypothesize that a deep hybrid model composed of a Graph Neural Network (GNN) and a Convolutional Neural Network (CNN) can approximate this  $\mathcal{H}$  because of their abilities to capture both large scale spatial relationships as well as localized patterns. Specifically, the GNN is adept at modelling non-Euclidean spatial dependencies between different regions, while the CNN can identify local patterns within individual regions. The hybrid model can thus effectively account for the spatial heterogeneity and sparsity inherent in the problem.

Any such approximate function  $f_{\mathcal{R}}$  over any spatial region  $\mathcal{R}$ , must account for variations in spatial features and surface characteristics throughout  $\mathcal{R}$ . Thus instead of having a single  $f$  over the entire  $\mathcal{S}$ , which would have led to an under-constrained, highly non-convex optimisation problem, we partitioned the lunar surface into 64 equisized segments. This allows us to give a more precise and better region-specific approximation of  $f_{\mathcal{R}}$  for each of these regions. The models can predict elemental abundances at high resolution across the entire lunar surface by learning these adaptive relationships.

Thus, we can formulate our problem as follows: our objective is to predict elemental abundances at a finer spatial resolution of (approximately)  $2\text{ km} \times 2\text{ km}$  using the abundances measured over larger regions provided by the CLASS instrument, i.e., given a set of  $N$  observed data samples  $\{L_{1,i}, L_{2,i}, \mathbf{A}_i\}_{i=1}^N$ , where  $L_{1,i} \in [90^\circ, -90^\circ]^4$ ,  $L_{2,i} \in [-180^\circ, 180^\circ]^4$  are the latitude and longitudes of the 4 corner points of any observation, and  $\mathbf{A}_i \in \mathbb{R}^8$  are the observed elemental abundances of the 8 elements ('Fe', 'Ti', 'Ca', 'Si', 'Al', 'Mg', 'Na', 'O') in that observation, we need to approximate the region specific mapping function  $f_{\mathcal{R}}$  for all the regions. This region-specific learned mapping  $f_{\mathcal{R}} : \mathcal{X}_{\mathcal{R}} \rightarrow \mathcal{Y}$ , where  $\mathcal{X}_{\mathcal{R}}$  is the region constrained input feature space, should generalize to both seen locations (where information about elemental abundances is available) and unseen locations.

We solve this problem by breaking it into 3 steps:

- 1) Information Initialisation
- 2) Abundance Population
- 3) Abundance Generalization

In Step 1, we initialize and define various data structures and constructs to help with the remaining steps. In Step 2, i.e. Abundance Population, we map the processed data onto the data structures initialised above, and in Step 3, i.e. Abundance Generalization, we use the information at the known grid locations to approximate  $f_{\mathcal{R}}$  and predict elemental abundances at all points in that region  $\mathcal{R}$ . We also utilize overlapping tracks to increase the density of known data grid locations.

*1) Related works:* Recent advancements in Spatial interpolation and resolution enhancement highlight innovative methods that uses neural networks, attention mechanisms, and hybrid architectures to overcome limitations of traditional approaches. Viana and Barbosa [4] introduced an attention-based spatial interpolation technique to enhance house price prediction accuracy. Their method avoids reliance on costly data sources such as satellite imagery or census data. By combining a Euclidean-based attention mechanism for structural feature similarity with Geo Attention for geographic proximity, the model generates house embeddings that capture both structural and spatial contexts. Similarly, Zhou et al. [5] developed a frost prediction model tailored for agricultural applications that eliminates the need for on-site historical data or sensors. This model uses geographic and environmental factors, such as elevation, vegetation, and distance, to aggregate predictions through weighted averages. Appleby et al. [6] extended Kriging principles by integrating them with Graph Neural Networks (GNNs) to create Kriging Convolutional Networks (KCN). In this innovative approach, data points are modeled as graph nodes, with edge weights representing spatial proximity. By utilizing attention mechanisms, the KCN dynamically adjusts the influence of neighboring nodes, offering significant flexibility and avoiding retraining for new datasets.

In the domain of X-ray fluorescence (XRF) imaging, super-resolution has been employed, particularly in Earth-based applications. For instance, methods such as "Deep Learning Enhanced Super-Resolution XRF Microscopy" [7] and "XRF Image Super-Resolution Using Dictionary Learning" [8] employ high-resolution structural or RGB data to complement low-resolution XRF measurements. Other approaches, like "Super-Resolution for Macro XRF Data Collected from Old Master Paintings" [9] and "Resolution-Enhanced XRF Microscopy via Deep Learning" [10], utilize multimodal data fusion with complementary imaging modalities such as ptychography or RGB images to achieve resolution enhancement. These strategies have proven effective in controlled environments, such as laboratory settings and artwork analysis.

However, such methods face challenges when applied to lunar XRF data due to unique constraints, including extreme data sparsity, spatial heterogeneity, and the absence of high-resolution complementary imaging. To address these challenges, our methodology employs a hybrid architecture combining Graph Neural Networks [4] and Convolutional Neural Networks [11]. Our methodology exclusively relies on lunar XRF data, effectively overcoming the limitations of external dependencies.

## B. Proposed Methodology

The entire lunar surface is divided into 64 equisized regions based on latitude, and longitude. Each region has a latitude span of  $22.5^\circ$  and longitude span of  $45^\circ$ . This can be visualised in Figure 6. We construct a Global Information Bank (GIB)



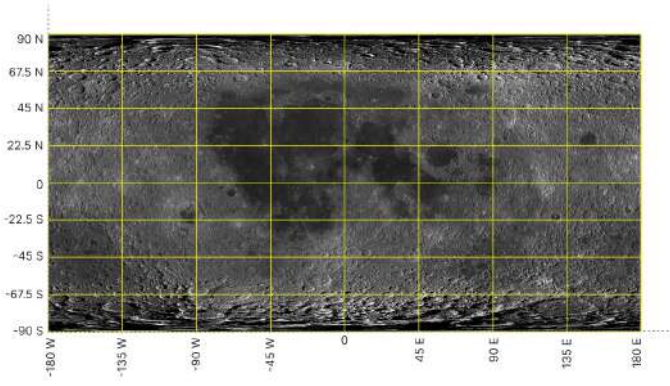


Fig. 6: Partitioning of the Lunar Surface in 64 divisions

to store all computation information, defined as follows:

$$GIB = \bigcup_{(i,j) \in [1 \dots 8]^2} (GIB_{i,j}),$$

where  $GIB_{i,j}$  is the region specific  $GIB$ , defined as,

$$GIB_{i,j} = (L_1, L_2, \mathfrak{A}, \mathfrak{M}, \mathfrak{M}),$$

where  $L_1$  is the vector of latitudes of center of super-resolved pixels and  $L_2$  is the vector of longitudes,  $\mathfrak{A}$  is the matrix of elemental abundances at every pixel,  $\mathfrak{M}$  is a matrix of surface characteristics vector  $\mathbf{W}_k$  of every pixel  $k$ , and  $\mathfrak{M}$  is a 2 class vector, which is either 1 or 2, signifying if the center of the pixel lies in a Mare region or a Highland region on the lunar surface.

The final hybrid model aims to learn the mapping from  $\mathfrak{M}, \mathfrak{M}, L_1, L_2$  to  $\mathfrak{A}$  for known points, and tries to generalize this mapping to unknown points by using the similarity between the surface characteristics (or  $\mathbf{W}_k$ 's) and the spatial distance as captured by  $L_1$  and  $L_2$ . The GNN tries to incorporate global context by considering pixels that are spatially far but feature-wise close, while the CNN tries to incorporate local context and patterns from spatially close nodes.

The complete methodology for achieving sub-pixel resolution consists of 3 main steps:

1) *GIB Initialization*: For each of the 64 regions, we initialize  $GIB_{i,j}$  by finding the centers of each of the superresolved pixels. Each  $22.5^\circ \times 45^\circ$  region is divided into a  $342 \times 682$  grid, s.t. the latitude or longitude difference between any 2 super-resolved pixels corresponds to approximately 2 km at the equator.

For each super-resolved pixel's,  $k$ , latitude and longitude, we calculate  $\mathbf{W}_k$ , to denote its surface characteristics. To ensure feature continuity and overlap between 2 adjacent pixels, instead of taking a  $2\text{ km} \times 2\text{ km}$  surface region, we consider a  $12.5\text{ km} \times 12.5\text{ km}$  surface region centered at that pixel and calculate its surface characteristics by employing a ResNet50 model [12]. The model outputs surface characteristic vectors of length 2048. To speed up computation, we selected the top 300 features greedily based on relevance and redundancy using the Max-Relevance Min-Redundancy (mRMR) feature

selection method. This method greedily selects the feature that is the most informative (quantified by maximum variance), while minizing overlap (correlation) with already selected features [13]. This 300 length vector  $\mathbf{W}_k$  forms the surface characteristic vector at super-resolved pixel  $k$ .

2) *Abundance Population*: The processed elemental abundances, given by,  $\{L_{1,i}, L_{2,i}, \mathbf{A}_i\}_{i=1}^N$ , represents a data region of size approximately equal to  $12.5\text{ km} \times 100\text{ km}$ . In order to map these abundances into the  $GIB$ , we make the following assumptions:

- The  $12.5\text{ km} \times 100\text{ km}$  region is composed of 8 adjacent  $12.5\text{ km} \times 12.5\text{ km}$  regions, each of which contribute equally to the final abundance calculated over the entire  $12.5\text{ km} \times 100\text{ km}$  region.
- The abundance of each of the  $12.5\text{ km} \times 12.5\text{ km}$  region is assumed to be concentrated at its center coordinate.

These 2 assumptions allow us to map every  $12.5\text{ km} \times 100\text{ km}$  observation onto 8 different points in the  $GIB$ . We made these assumptions to balance maximum population and ability to deviate from the observed data.

In order to find the transformation vector  $B \in \mathbb{R}^{1 \times 8}$ , s.t. the calculated  $12.5\text{ km} \times 12.5\text{ km}$  abundances  $C_i$  for each  $i \in [1 \dots 8]$ , can best approximate the abundance over the entire  $12.5\text{ km} \times 100\text{ km}$  region  $A \in \mathbb{R}^{8 \times 1}$ , we phrase and solve a Lagrangian optimization problem.

Define a matrix  $C \in \mathbb{R}^{8 \times 8}$ , whose columns are given by

$$C_i = A \cdot b_i$$

, where  $b_i$  is the  $i$ -th element of the transformation row vector  $B \in \mathbb{R}^{1 \times 8}$ ,  $A \in \mathbb{R}^{8 \times 1}$  are the abundances of the 8 elements over 1 data  $12.5\text{ km} \times 100\text{ km}$  region, and the columns of the matrix  $C \in \mathbb{R}^{8 \times 8}$  represents the elemental abundances of the 8 smaller parts. In order to get a constrained optimization problem, we make the following assumptions: The elemental assumptions at each of the center pixels, of each of the  $12.5\text{ km} \times 12.5\text{ km}$  do not deviate much from the expected elemental abundance at that location, assuming its either a Mare or Highland region.

This assumption gives rise to the following loss function:

$$\text{Loss}(B) = \sum_{i=1}^8 \left\| A \cdot b_i - \tilde{A} \right\|_2^2,$$

where  $\tilde{A} \in \mathbb{R}^{8 \times 1}$  either equals to  $A_h$  or  $A_m$ , which represents the average elemental abundances of Highland and Mare respectively.

This loss function can be written as

$$\text{Loss}(B) = \|A \cdot B - D\|_F^2,$$

where  $\|\cdot\|_F$  represents the Frobenius norm of the matrix, and  $D \in \mathbb{R}^{8 \times 8}$  is a matrix such that its  $i$ -th column  $D_i = \tilde{A}$ .

The first set of assumptions also leads to the following global consistency constraint:

$$\frac{1}{8} \sum_{i=1}^8 C_i = A,$$



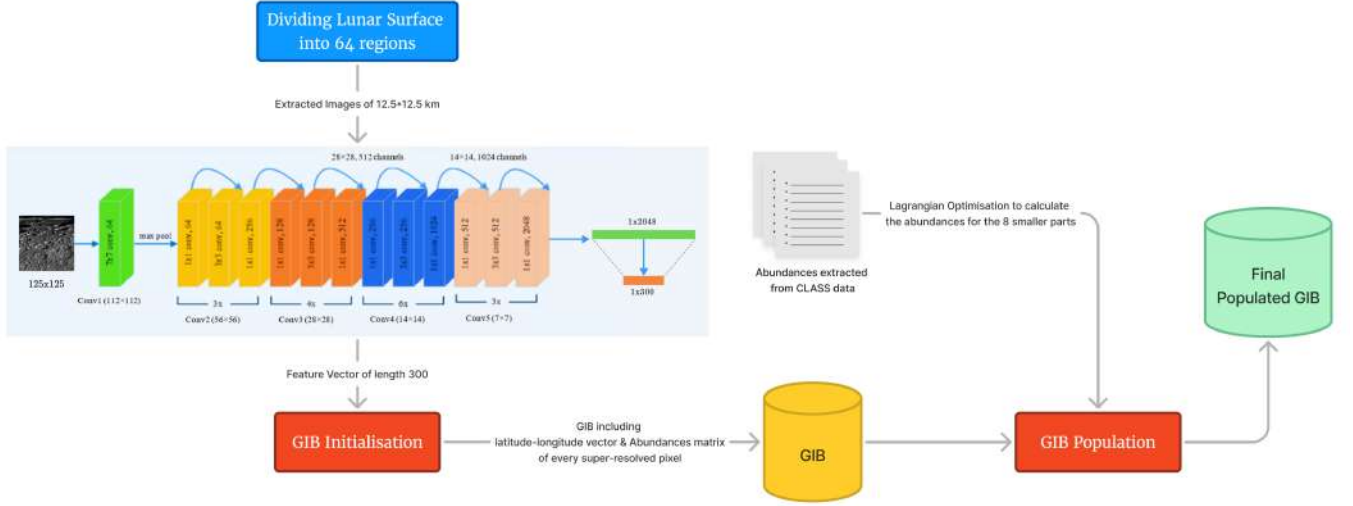


Fig. 7: GIB initialisation with latitude longitude and the abundances vector along with the top 300 features extracted using the ResNet50 network to obtain the initialised GIB. Further populating them with the elemental abundances using Optimisation to obtain the final populated GIB .

which can be rewritten as:

$$\frac{1}{8} A \cdot (B \cdot \mathbf{1}_8) = A,$$

where  $\mathbf{1}_8$  is an  $8 \times 1$  column vector of ones.

In addition to the global consistency constraint, we impose the non-negativity constraint on the elemental abundances. The elemental abundances for the  $i$ -th location,  $C_i = [c_{i,1}, c_{i,2}, \dots, c_{i,8}]^T$ , must satisfy:

$$c_{i,j} \geq 0, \quad \text{for } j = 1, 2, \dots, 8.$$

Thus the optimization problem subject to the constraints mentioned above can be formulated as:

$$\min_B \|A \cdot B - D\|_F^2,$$

To solve this constrained optimization problem, we introduce two Lagrange multipliers  $\lambda$  &  $\mu$ . Hence the Lagrangian function for the optimization problem becomes

$$L(B, \lambda, \mu) = \|A \cdot B - D\|_F^2 + \lambda^T (A \cdot (B \cdot \mathbf{1}_8) - 8A) + \mu^T \cdot \max(0, -B).$$

Here  $\mu$  is the Lagrange multiplier for the non-negativity constraint, and  $\max(0, -B)$  introduces a penalty for negative values in  $B$ .

To solve the optimization problem with the added non-negativity constraint, we take the gradients [14] of the Lagrangian with respect to  $B$ ,  $\lambda$ , and  $\mu$ :

$$\frac{\partial L}{\partial B} = 2A^T (A \cdot B - D) + \lambda^T A + \mu^T \cdot \text{sign}(-B),$$

$$\frac{\partial L}{\partial \lambda} = A \cdot (B \cdot \mathbf{1}_8) - 8A,$$

$$\frac{\partial L}{\partial \mu} = \max(0, -B).$$

Setting the gradients to zero, we solve for  $B$ ,  $\lambda$ , and  $\mu$ :

$$2A^T (A \cdot B - D) + \lambda^T A + \mu^T \cdot \text{sign}(-B) = 0,$$

$$A \cdot (B \cdot \mathbf{1}_8) - 8A = 0,$$

$$\max(0, -B) = 0.$$

We solve the optimization problem iteratively by updating  $B$ ,  $\lambda$ , and  $\mu$  using gradient descent:

$$B_{\text{new}} = B_{\text{old}} - \eta \frac{\partial L}{\partial B},$$

$$\lambda_{\text{new}} = \lambda_{\text{old}} - \eta \frac{\partial L}{\partial \lambda},$$

$$\mu_{\text{new}} = \mu_{\text{old}} - \eta \frac{\partial L}{\partial \mu},$$

where  $\eta$  is the learning rate.

This optimization problem is solved for all  $\{L_{1,i}, L_{2,i}, \mathbf{A}_i\}_{i=1}^N$  and the columns of the calculated matrix  $C$  gives the elemental abundances at the  $8 \times 12.5 \text{ km} \times 12.5 \text{ km}$  region's center super-resolved pixel, which is updated in the  $GIB$ .

3) *Abundance Generalization*: Abundance generalization consists of learning the mapping  $f_{\mathcal{R}} : \mathcal{X}_{\mathcal{R}} \rightarrow \mathcal{Y}$  for each of the 64 well-define regions  $\mathcal{R}$ . We learn this mapping using the sparsely populated  $GIB$  created in Step 2.

To capture long-range relationships we employ a Graph Neural Network. To represent our lunar region as a graph, we consider each super-resolved pixel as a Node [15]. Since each lunar region consists of  $(342 \times 682 =) 233244$  such nodes, it would be infeasible to load the entire region as a single graph. It is required to break a single region spanning  $22.5^\circ \times 45^\circ$ , into multiple subregions, each of which will have a subgraph. To ensure easier learning of the mapping function on this region,

the subgraphs have to be overlapping, otherwise, it would be impossible to go from global minimum of 1 subregion to the global minimum of another subregion. This requirement of overlapping subregions also has the benefit of Data Diffusion, i.e. once the elemental abundances are predicted for 1 sub-region, this allows for that new information to propagate to all overlapping subregions. Such a data diffusion is extremely important considering the sparsity of the data.

We divide each region consisting of  $342 \times 682$  super-resolved pixels into 78 subregions, by taking each subregion to have 100 pixel length and 100 pixel breadth and a stride of 50 pixels in each direction, thus giving us 6 grid cells along the latitude and 13 grid cells along the longitude. Let  $\mathcal{H} = \{H_1, H_2, \dots, H_{78}\}$  represent the set of these overlapping graphs where each graph  $H_k$  corresponds to a subregion within a region in the  $8 \times 8$  grid and overlaps with its adjacent subregions.

For each region  $\mathfrak{R}$  we construct  $H_l$  for  $l = [1...78]$ , the overlapping subgraph's of that region. Each  $H_l = (\mathcal{N}_l, \mathcal{E}_l)$ , since we construct the overlapping subgraph with a fixed size,  $|\mathcal{N}_l| = 100 \times 100 = 10,000 \forall l \in [1...78]$ . Each node  $n \in \mathcal{N}_l$  contains the following information:

- $\text{lat}_n, \text{lon}_n$ : The latitude and longitude of the center of the super-resolved pixel that node  $n$  represents.
- $\mathbf{W}_n$ : A column vector of length 300 denoting the surface characteristics of the  $12.5 \text{ km} \times 12.5 \text{ km}$  centered at the center of super-resolved pixel that node  $n$  represents.
- $A_n$ : A column vector of length 8 representing the Elemental abundances of the 8 elements if known at that node  $n$ , otherwise 0.

Since we want to capture long range dependencies and relationships in this graph, the set of edges in the graph must connect nodes even if they are spatially far but close w.r.t. surface characteristics. Thus we define an edge-weight function

$$w : (\mathcal{N}_l \times \mathcal{N}_l) \rightarrow \mathbb{R}$$

as follows:

$$w(n_1, n_2) = \exp \left( -\alpha \cdot \text{haversine}(G_{n_1}, G_{n_2}) - \beta \cdot \|\mathbf{W}_{n_1} - \mathbf{W}_{n_2}\|_2^2 \right)$$

such that the scaling factor  $\beta > \alpha$  implying the requirement that surface feature similarity matters more than spatial similarity. Here,  $G_{n_1}$  and  $G_{n_2}$  represents the latitude and longitude at each of  $n_1$  and  $n_2$  respectively. Thus, connections between nodes are determined by an edge weight function that decreases exponentially with spatial distance and feature-based dissimilarities. Here,

- $\text{haversine}(G_i, G_j)$  is the haversine distance between the geographic coordinates of nodes  $i$  and  $j$  [16].

$$\text{haversine}(G_i, G_j) = 2r \arcsin \left( \sin^2 \left( \frac{\text{lat}_j - \text{lat}_i}{2} \right) + \cos(\text{lat}_i) \cos(\text{lat}_j) \sin^2 \left( \frac{\text{long}_j - \text{long}_i}{2} \right) \right)^{1/2}$$

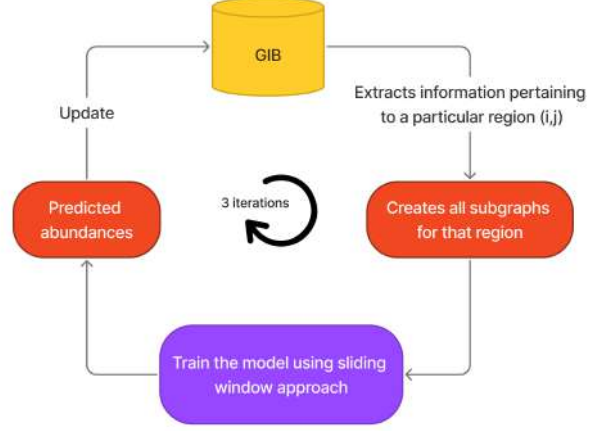


Fig. 8: Model pipeline for abundance generalisation. Pictorial representation that illustrates our workflow starting from extraction of information from GIB all the way to training the sub-graphs and updating the predicted abundances into the csv files.

where  $r$  is the radius of the moon.

- $\|\mathbf{W}_i - \mathbf{W}_j\|_2^2$  is the squared Euclidean distance between the feature vectors of the nodes  $i$  and  $j$ .

For each node  $n \in \mathcal{N}_l$ , we select the top 100 neighbors with the largest edge weight, signifying their closeness either spatially or feature wise. The mapping function we aim to learn must account for both local and global relationships. We hypothesize that the following described hybrid model composed of GNN + CNN followed by a fusion deep layers gives a rich enough Hypothesis class to approximate the mapping function. The complete model architecture has been shown in Figure 9.

The GNN component employs 2 GATv2Conv layers in succession. Each GATv2Conv layer computes node embeddings by dynamically attending to neighbors in a graph [17].

Let the input graph to any of the GATv2Conv layers be represented by a set of nodes  $\mathcal{N}$  and edges  $\mathcal{E}$ , with node features  $\mathbf{X} \in \mathbb{R}^{|\mathcal{N}| \times F_{in}}$  and edge features  $\mathbf{E} \in \mathbb{R}^{|\mathcal{E}| \times D}$ . In our case,  $F_{in} = 303$  (300 length surface characteristic vector, 1 scalar representing whether the super-resolved pixel that node  $n$  represents is either Mare or Highland, and 2 scalar values representing the latitude and longitude of that pixel). At any GATv2Conv layer, the updated representation of a node  $i$  is computed as:

$$\mathbf{h}'_i = \sum_{j \in \mathcal{N}(i) \cup \{i\}} \alpha_{ij} \Theta_t \mathbf{h}_j,$$

- $\mathbf{h}'_i$  is the updated feature vector of node  $i$ .
- $\alpha_{ij}$  represents the attention coefficient between nodes  $i$  and  $j$ , defined as:

$$z_{ij} = \mathbf{a}^\top \text{LeakyReLU}(\Theta_s \mathbf{h}_i + \Theta_t \mathbf{h}_j + \Theta_e \mathbf{e}_{ij}),$$

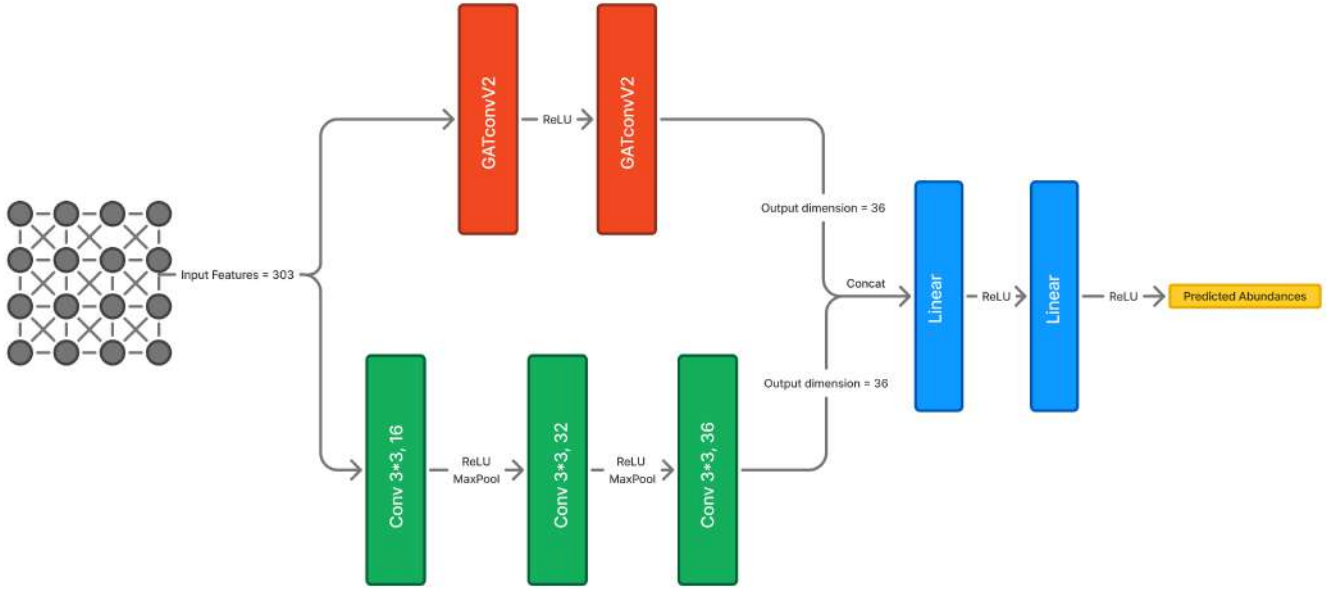


Fig. 9: The Hybrid Model architecture involving two GNN (GATv2conv) layers and three convolutional layers. The results of both the networks are concatenated and passed through fusion layers to obtain the final predicted elemental abundances.

$$\alpha_{ij} = \frac{\exp(z_{ij})}{\sum_{k \in \mathcal{N}(i) \cup \{i\}} \exp(z_{ik})}.$$

- $\Theta_s, \Theta_t, \Theta_e$  are learnable weight matrices for query nodes, key nodes, and edge features, respectively.
- $\mathbf{a}$  is a learnable vector that computes attention scores.

The proposed GNN network allows conditioning the attention coefficients on the query node, this allows for learning richer node representations, compared to older methods based on static attention.

The CNN component processes spatial data represented as a grid, where node features are mapped to pixels using latitude (lat) and longitude (lon) indices. The mapping is computed as:

$$\text{lat\_index}_i = \left\lfloor \frac{\text{lat}_i - \text{lat}_{\max}}{\text{lat}_{\min} - \text{lat}_{\max}} \cdot (n_{\text{grid}} - 1) \right\rfloor,$$

$$\text{lon\_index}_i = \left\lfloor \frac{\text{lon}_i - \text{lon}_{\min}}{\text{lon}_{\max} - \text{lon}_{\min}} \cdot (n_{\text{grid}} - 1) \right\rfloor,$$

where  $n_{\text{grid}}$  is the number of grid points along each dimension. This grid structure is then passed through the CNN layers to get a final grid of extracted features. The CNN extracted features for each node are then extracted from the grid based on the indexing equations mentioned above.

The CNN comprises three of convolutional layers interspersed with max-pooling operations. Each convolution is defined as:

$$F_{ij}^{(l)} = \sigma \left( \sum_{p=-k}^k \sum_{q=-k}^k w_{pq}^{(l)} F_{i+p, j+q}^{(l-1)} + b^{(l)} \right),$$

where:

- $F_{ij}^{(l)}$  is the output at pixel  $(i, j)$  in layer  $l$ .
- $w_{pq}^{(l)}$  are the filter weights.
- $b^{(l)}$  is the bias term.
- $\sigma$  is the activation function (ReLU used here).
- $k$  is half the size of the filter used in each dimension.

Forwarding a subregion through the GNN and CNN backbones results in node embeddings that encompass both the global context and local context for each node. These high dimensional embeddings can then be used to predict the elemental abundances by learning a mapping via deep neural network layers.

So, the outputs from the GNN and CNN are concatenated to form a unified representation:

$$\mathbf{h}_{\text{fusion}} = \text{ReLU}(\mathbf{W}_f [\mathbf{h}_{\text{GNN}}, \mathbf{h}_{\text{CNN}}] + \mathbf{b}_f),$$

where:

- $[\mathbf{h}_{\text{GNN}}, \mathbf{h}_{\text{CNN}}]$  is the concatenation of GNN and CNN outputs.
- $\mathbf{W}_f$  and  $\mathbf{b}_f$  are the learnable weights and bias of the fusion layer.

The final prediction is computed using a fully connected layer:

$$\mathbf{y} = \mathbf{W}_o \mathbf{h}_{\text{fusion}} + \mathbf{b}_o,$$

where  $\mathbf{W}_o$  and  $\mathbf{b}_o$  map the fused representation to the output space of elemental abundances. This entire procedure has been illustrated in Figure 8.

Finding the correct mapping function/hypothesis in this expansive hypothesis class  $\mathcal{H}$  requires us to select a loss function that incorporates all the constraints inherent in the problem. These losses can be expressed as follows:

**Weighted Masked Mean Squared Error (MSE) Loss:** A weighted masked MSE loss is incorporated to enforce correct predictions of abundances at known data points. Formally, it is defined as:

$$\mathcal{L}_{\text{masked\_mse}} = \frac{\sum_{i=1}^N \mathbb{1}_{\text{mask}_i} \cdot \sum_{j=1}^8 w_j \cdot (\hat{y}_{i,j} - y_{i,j})^2}{\sum_{i=1}^N \mathbb{1}_{\text{mask}_i}}$$

where  $\mathbb{1}_{\text{mask}_i}$  is an indicator function specifying if the abundances for node  $i$  are entirely known,  $w_j$  represents element-specific weights, and  $\hat{y}_{i,j}$  and  $y_{i,j}$  denote predicted and true abundances, respectively. This loss ensures the model focuses on accurate predictions for available data.

**Weighted masked Logarithmic Loss:** To address the wide dynamic range in elemental abundances, a logarithmic loss is added [18]. It emphasizes relative differences rather than absolute magnitudes, which is more indicative in case of elements with small abundance. The loss function is given by:

$$\mathcal{L}_{\log} = \frac{\sum_{i=1}^N \mathbb{1}_{\text{mask}_i} \sum_{j=1}^8 w_j (\log(\hat{y}_{i,j} + \epsilon) - \log(y_{i,j} + \epsilon))^2}{\sum_{i=1}^N \mathbb{1}_{\text{mask}_i}}$$

where  $\epsilon$  is a small constant added for numerical stability, and the rest of the variables are as described above. This log loss allows every element to have an equal contribution in the loss.

**Feature Similarity Loss:** The Feature Similarity Loss is used to enforce consistency in elemental abundances prediction for nodes with similar surface characteristics. For a given subgraph  $H_l$  This loss is defined as:

$$\mathcal{L}_{\text{feature\_sim}} = \frac{1}{|\mathcal{E}_l|} \sum_{(s,t) \in \mathcal{E}_l} w_{st} \cdot \|\hat{y}_s - \hat{y}_t\|_2^2$$

where  $w_{st} = \exp(-\|\mathbf{W}_s - \mathbf{W}_t\|_2)$  is a weight reflecting the similarity between feature vectors  $\mathbf{W}_s$  and  $\mathbf{W}_t$  of nodes  $s$  and  $t$ . By penalizing large abundance differences for nodes with similar features, this loss accounts for the alignment of predictions of nodes with known physical relationships.

**Spatial Similarity Loss:** The Spatial Similarity Loss enforces consistency in predictions between spatially close nodes. This constraint is based on the assumption that elemental abundances should vary smoothly. The loss is defined as:

$$\mathcal{L}_{\text{spatial\_sim}} = \frac{1}{|\mathcal{E}_l|} \sum_{(s,t) \in \mathcal{E}_l} w_{st} \cdot \|\hat{y}_s - \hat{y}_t\|_2^2$$

with:

$$w_{st} = \exp\left(-\frac{\|\text{loc}_s - \text{loc}_t\|_2 - d_{\min}}{d_{\max} - d_{\min} + \epsilon}\right)$$

where  $\text{loc}_s$  and  $\text{loc}_t$  are the spatial coordinates of nodes  $s$  and  $t$ , and  $d_{\min}, d_{\max}$  denote the minimum and maximum pairwise distances.

**Deviation Loss:** This loss enforces that the sum of predicted abundances is close to 100%. The deviation loss is given by:

$$\mathcal{L}_{\text{deviation}} = \frac{1}{N} \sum_{i=1}^N \left| \sum_{j=1}^8 \hat{y}_{i,j} - 100 \right|$$

The Weighted Masked MSE Loss and the Weighted Masked Logarithmic Loss are responsible for learning the mapping from node-level embeddings to predicted abundances for  $2\text{km} \times 2\text{km}$  pixels where the data is already populated. The Feature Similarity Loss and Spatial Similarity Loss are responsible for smoothening the predicted abundances across nodes that are spatially and feature-wise close. These 2 loss functions are responsible for the generalization of the mapping function for nodes where abundances are not known.

All 5 of these loss functions guide the search process of the mapping function in the hypothesis class  $\mathcal{H}$ , to ultimately arrive at the best mapping function for that region.

The problem for finding the best mapping function in the hypothesis class  $\mathcal{H}$  involves optimization in a highly non-convex landscape. The non-convexity arises from the multiple non-convex loss functions, along with the non-convexity inherent in the model architecture. This makes the search for an optimal mapping function within the hypothesis class  $\mathcal{H}$  more challenging.

However, with empirical analysis we found that there does exist a global minimum basin, and thus the search problem can be formulated similar to a Spiked Matrix-Tensor model [19]. Thus using a learning rate scheduler that follows a search-convergence pattern, we can probabilistically guarantee convergence to the global minimum [19]. We used a Cosine Annealing Learning Rate scheduler, which follows a search-convergence pattern, along with various other heuristics, thus increasing our chances of convergence. The following methods were used to increase our chances of convergence:

**Gradient Clipping** was employed to prevent the issue of exploding gradients, which is common in regions of steep curvature in the loss landscape. For each parameter  $\theta$ , the gradient  $\nabla\theta$  is rescaled when its norm exceeds a predefined threshold  $\tau$  (1.0 in our case):

$$\nabla\theta \leftarrow \nabla\theta \cdot \min\left(1, \frac{\tau}{\|\nabla\theta\|_2}\right),$$

where  $\|\nabla\theta\|_2$  is the L2 norm of the gradients. Imposing this constraint on the magnitude of the gradient can help prevent erratic training behaviour.

**YOGI Optimizer** is an adaptive gradient method, that is used to stabilize training in sparse and noisy gradient scenarios [20]. YOGI is different from traditional optimisers such as ADAM [21], since it introduces a corrective mechanism that adapts the second-moment estimate of the gradients  $v_t$  as follows:

$$v_t = v_{t-1} - \text{sign}(v_{t-1} - g_t^2) \cdot \eta,$$

where  $g_t$  is the gradient at time  $t$  and  $\eta$  is the learning rate. The parameter updates are given by:

$$\theta_{t+1} = \theta_t - \frac{\alpha}{\sqrt{v_t} + \epsilon} \cdot g_t,$$

where  $\alpha$  is the step size, and  $\epsilon$  is a small constant for numerical stability. YOGI is very effective for non-convex optimization tasks because of its ability to adjust learning rates adaptively.

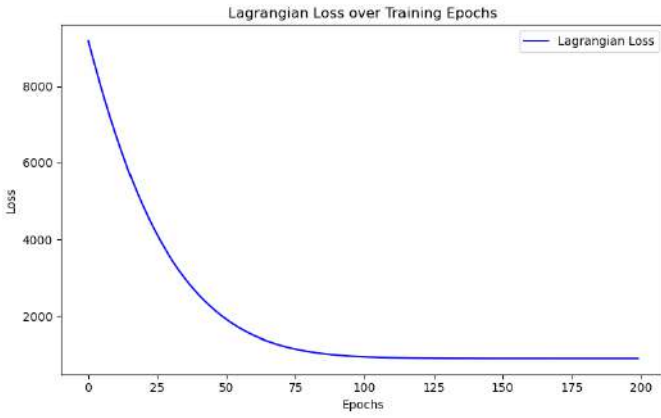


Fig. 10: Lagrangian minimization curve over epochs. This is calculated to distribute the abundances across 8 subregions of size 12.5 km x 12.5 km from abundances extracted from a region of size 12.5 km x 100 km.

**Cosine Annealing Learning Rate Scheduler** was employed to adjust the learning rate during training in search-convergence fashion [22]. The learning rate  $\eta_t$  at step  $t$  is computed as:

$$\eta_t = \eta_{\min} + \frac{1}{2}(\eta_{\max} - \eta_{\min}) \left( 1 + \cos \left( \frac{t}{T_{\max}} \cdot \pi \right) \right),$$

where  $\eta_{\min}$  and  $\eta_{\max}$  are the minimum and maximum learning rates, and  $T_{\max}$  is the total number of steps. The search-convergence exploration allows for effective navigation of this non-convex landscape.

A **Two-Phase Learning Scheme** was followed because of the presence of multiple loss functions. In the initial optimisation phase, the model was trained using only the weighted masked mean squared error ( $\mathcal{L}_{\text{masked\_mse}}$ ) and deviation loss ( $\mathcal{L}_{\text{deviation}}$ ):

$$\mathcal{L}_1 = \mathcal{L}_{\text{masked\_mse}} + \mathcal{L}_{\text{deviation}}.$$

This initial training allowed the model to find the global minimum of the convex MSE loss function and learn the mapping from features to abundances for known points. In the second phase of optimization, all the loss functions were used,

$$\mathcal{L}_2 = \mathcal{L}_{\text{masked\_mse}} + \lambda_{\text{reg}} \cdot \mathcal{L}_{\log} + \mathcal{L}_{\text{feature\_sim}} + \mathcal{L}_{\text{spatial\_sim}} + \lambda_{\text{reg}} \cdot \mathcal{L}_{\text{deviation}}.$$

This phase focuses on generalizing the mapping from known nodes to unknown nodes.

### C. Experimental Results and Conclusion

This section covers the experimental details, training details, mapping output and the dynamic nature of superresolution.

1) **Abundance Population:** During the abundance population, we minimize the Lagrangian for 80 epochs with  $\lambda = 1$  &  $\mu = 2$ . The minimization of the Lagrangian can be seen in Figure 10.

2) **Abundance Generalisation:** The model is trained for a region  $\mathfrak{R}$  by iteratively generating all overlapping subgraphs from the *GIB* and then training the model on each of the subgraphs in a sliding window fashion, to ensure there is no sudden domain shift and the model can remain in the global minimum. This iteration happens 3 times to ensure proper data diffusion of the new samples.

Anytime new data comes, we update the *GIB* by calculating the new abundance by solving the Lagrangian for every new sample (Takes around 0.167 seconds for 80 epochs). Following this we update the *GIB* and retrain the model for the updated regions. Thus our model and proposed training algorithm is inherently dynamic.

3) **Mapping Output:** The mapped output for super-resolved Mg abundance in the Lunar Region  $i = 0$   $j = 2$  which corresponds to the Lunar Region (90N,67.5N) and (90W,45W) has been shown in Figure 11

## V. MAPPING AND VISUALIZATION

### A. Introduction

Since the very beginning of lunar exploration with the Luna 1 [?], the mapping of moon has remained a subject of utmost importance. Accurate and reliable information about lunar geography and geology can allow mankind to take huge leaps forward in the field of space exploration as well as resource extraction from places beyond our planet. Precise mapping and visualization of any geological terrain or structure is vital to understanding its composition and evolutionary history [?]. Previous knowledge of elemental abundances of lunar surface can prove crucial to the planning of future lunar probe missions. The earliest attempts at mapping the moon were done by the Lunar orbiter program of NASA (1966-67) [?]paper) which not only mapped the lunar surface but also helped identify appropriate sites for the successive Apollo missions that followed. With advancement in technology, impetus was given to also studying the geology and mineralogy of the lunar surface as well. The Lunar Reconnaissance Orbiter (LRO) (2009–present) has proved to be a landmark with its extensive mapping data of lunar topography as well as mineral composition. Chandrayaan 1 sent by India also helped map the moon while confirming the presence of water near polar ice caps [?].

However, the primary focus of our research was Chandrayaan-2 or more specifically the data from Chandrayaan-2 Large Area Soft X-ray Spectrometer (CLASS). As mentioned earlier, it is an X-ray fluorescence experiment that aims to obtain high resolution global elemental maps of the lunar surface. The elemental abundances or more specifically the elemental ratios that we obtained from CLASS after thoroughly processing the fits files and enhancing the sub-pixel resolution showed the distribution and variation of different elements across the lunar surface. However since the data at the edges of the the range of CLASS (0.5 keV to 15 keV) was very noisy and difficult to process within reasonable number of iterations, it was decided to only select the peaks in the range of 0.9 keV to 4.2



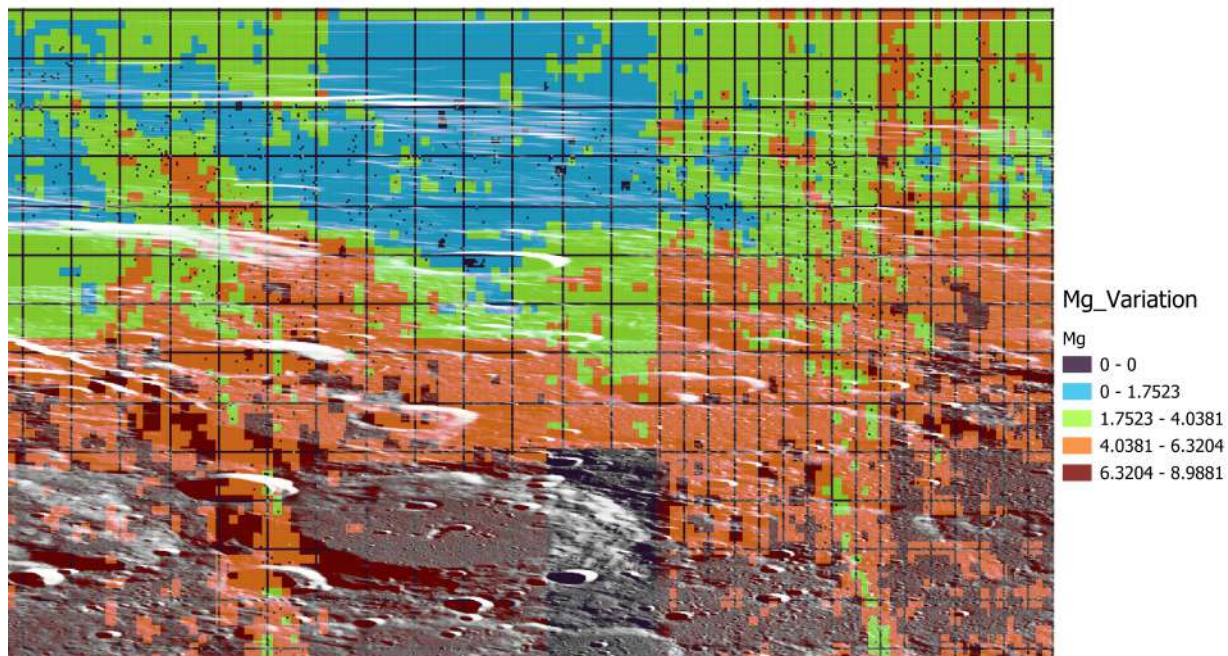


Fig. 11: Superresolved Mg Abundance in the lunar region (0, 2) corresponds to [(90N, 67.5N), (90W, 45W)]

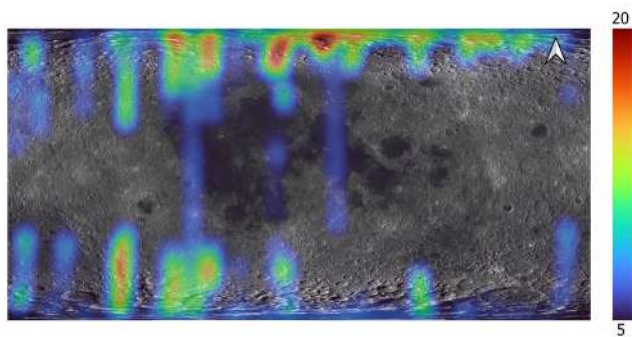


Fig. 12: Al % wt abundance on Lunar Albedo Map

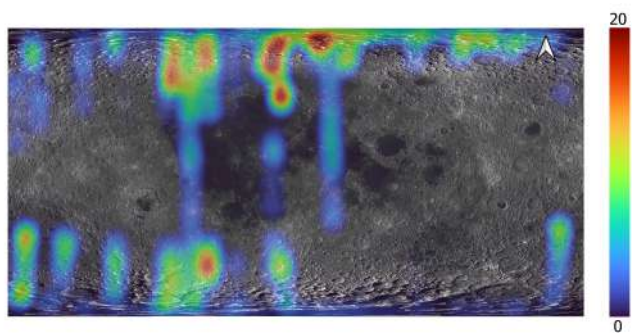
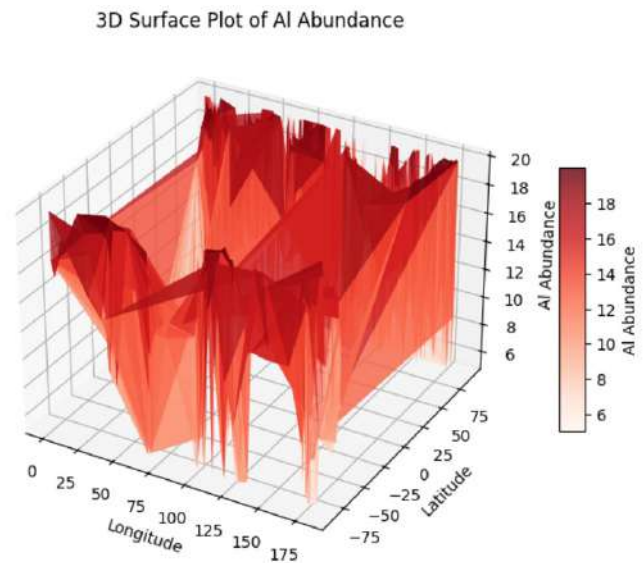


Fig. 13: Mg % wt abundance on Lunar Albedo Map



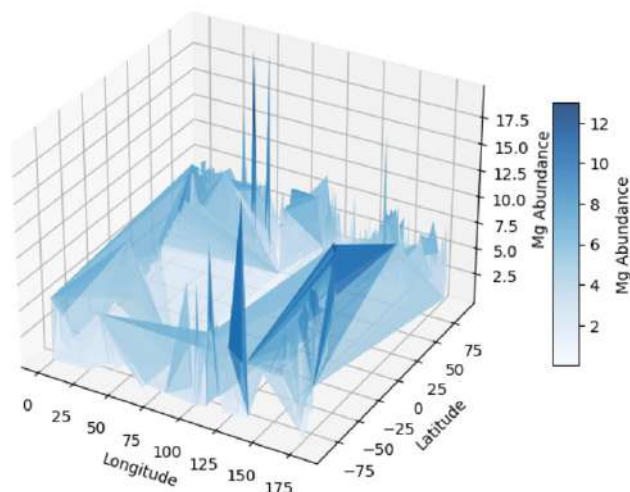
## B. Results

In this study, we successfully processed and visualized the spatial distribution of elemental abundances on the lunar surface through generation of heatmaps and graduated maps. We were also successful in PyQGIS scripting to automate our mapping tasks for importing Basemaps, formatting CSV files, creating heatmaps and graduated maps and exporting our final result in .tif (Geotiff) files and .shp (shape) files. Overall, the methods employed allowed for effective visualization and analysis of elemental abundances on the lunar surface, providing valuable insights for future lunar exploration and

keV. However, this means that the data for certain elements like O, Na, Ca, Fe and Ti can be somewhat inaccurate.



3D Surface Plot of Mg Abundance



resource mapping efforts.

## VI. SUMMARY

This study leverages the extensive CLASS instrument data from Chandrayaan-2 to provide a detailed analysis of lunar elemental abundances, offering significant insights into the Moon's surface composition. By employing XSPEC for spectral analysis, elemental abundances were precisely mapped onto the lunar albedo map, uncovering correlations between elemental distribution and surface reflectance properties. Additionally, the innovative application of machine learning methodologies enabled the enhancement of spatial resolution to sub-pixel levels, achieving a detailed 2 km resolution for elemental maps.

To further enhance data analysis, integrating complementary datasets such as high-resolution topographic maps or additional spectral data could provide a more comprehensive understanding of lunar surface processes. Improvements in detector technology and calibration techniques could also refine the accuracy of elemental abundance measurements. Exploring advanced machine learning models, such as transformer-based architectures for spatial data, may offer even better resolution and predictive accuracy.

These advancements underscore the value of combining advanced spectrometry with modern computational techniques, paving the way for deeper investigations into lunar geology and the Moon's evolutionary history. The methodologies and findings from this work not only enhance the scientific utility of Chandrayaan-2 data but also provide a framework for future planetary exploration and remote sensing missions.

## REFERENCES

[1] T. Shiraiwa and N. Fujino, "Theoretical calculation of fluorescent x-ray intensities in fluorescent x-ray spectrochemical analysis," *Japanese Journal of Applied Physics*, vol. 5, no. 10, p. 886, 1966.

[2] J. Fernández, "Rayleigh and compton scattering contributions to x-ray fluorescence intensity," *X-Ray Spectrometry*, vol. 21, pp. 57–68, 1992.

[3] P. S. Athiray, "Study of lunar surface chemistry using swept charge devices," 2015, available: <http://prints.iap.res.in/handle/2248/7440>.

[4] D. Viana and L. Barbosa, "Attention-based spatial interpolation for house price prediction," in *ACM Symposium on Neural Gaze Detection*, 2018, pp. 1–10.

[5] I. Zhou, J. Lipman, M. Abolhasan, and N. Shariati, "Intelligent spatial interpolation-based frost prediction methodology using artificial neural networks with limited local data," *Preprint submitted to Journal of LATEX Templates*, 2023, available: <https://arxiv.org/abs/2204.08465v2>.

[6] G. Appleby, L. Liu, and L.-P. Liu, "Kriging convolutional networks," in *Proceedings of the 37th AAAI Conference on Artificial Intelligence*, 2023, pp. 1–15, available: <https://arxiv.org/abs/2306.09463>.

[7] A. Smith, B. Johnson, and C. Lee, "Deep learning enhanced super-resolution xrf microscopy," *Optica*, available: <https://www.optica.org>.

[8] D. Brown, E. Taylor, and F. Garcia, "Xrf image super-resolution using dictionary learning," *Science for Art*, available: <https://scienceforart.org>.

[9] G. White, H. Clark, and J. Adams, "Super-resolution for macro xrf data collected from old master paintings," *ResearchGate*, available: <https://www.researchgate.net>.

[10] I. Hall, J. Green, and K. Martin, "Resolution-enhanced xrf microscopy via deep learning," *Nature*, available: <https://www.nature.com>.

[11] T. Ayodele, B. J. Afolabi, F. O. Folorunso, and O. K. Adekunle, "Survey of big data frameworks: Application perspectives," *Journal of Big Data*, vol. 8, no. 31, pp. 1–25, 2021.

[12] S. A. Agrawal, V. D. Rewaskar, R. A. Agrawal, S. S. Chaudhari, Y. Patil, , and N. S. Agrawal, "Advancements in nsfw content detection: A comprehensive review of resnet-50 based approaches," *International Journal of Intelligent Systems and Applications in Engineering (IJISAE)*, vol. 11, no. 4, pp. 41–45, 2023, available: <https://www.researchgate.net>.

[13] M. Shirzad and M. Keyvanpour, "A feature selection method based on minimum redundancy maximum relevance for learning to rank," in *Proc. RIOS*, Apr. 2015, pp. 1–5.

[14] J. Smith, "Title of the article," *Journal of Big Data*, vol. 10, no. 1, pp. 1–12, 2023, available: <https://doi.org/10.1186/s40537-023-00839-9>.

[15] J. S. Badwaik, "Collaborative activities," NAAC Criterion-3: 3.7, n.d., available: <https://www.acu.edu.in/naac/naac/Criterion-3/3.7/COLLABORATIVE%20ACTIVITIES/688.pdf>.

[16] A. R. Azdy and F. Darnis, "Use of haversine formula in finding distance between temporary shelter and waste end processing sites," *ResearchGate*, 2020, available: [https://www.researchgate.net/publication/341727897\\_Use\\_of\\_Haversine\\_Formula\\_in\\_Finding\\_Distance\\_Between\\_Temporary\\_Shelter\\_and\\_Waste\\_End\\_Processing\\_Sites](https://www.researchgate.net/publication/341727897_Use_of_Haversine_Formula_in_Finding_Distance_Between_Temporary_Shelter_and_Waste_End_Processing_Sites).

[17] D. Viana and L. Barbosa, "Attention-based spatial interpolation for house price prediction," *arXiv preprint*, 2021, available: <https://arxiv.org/abs/2105.14491>.

[18] N. P. Sukumar and H. Choo, "On numerical differentiation and its applications in finite element analysis," *arXiv preprint*, 2022, available: <https://arxiv.org/pdf/2201.05748>.

[19] S. d'Ascoli, M. Reinetti, and G. Biroli, "A deep learning model for classifying human facial expressions from infrared thermal images," *arXiv preprint arXiv:2202.04509*, 2022, available: <https://arxiv.org/abs/2202.04509>.

[20] A. G. Schmidt, A. L. Y. Lau, and M. A. Gabel, "An empirical study of optimizations in yogi," *ResearchGate*, 2012, available: [https://www.researchgate.net/publication/221554048\\_An\\_empirical\\_study\\_of\\_optimizations\\_in\\_YOGI](https://www.researchgate.net/publication/221554048_An_empirical_study_of_optimizations_in_YOGI).

[21] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1412.6980*, 2014, available: <https://arxiv.org/abs/1412.6980>.

[22] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *arXiv preprint arXiv:1608.03983v5*, 2016, available: <https://arxiv.org/abs/1608.03983v5>.