UNIVERSIDAD DE LAS AMÉRICAS PUEBLA

Escuela de Ciencias

Departamento de Actuaría, Física y Matemáticas

Mapping the Zoo of Rest-Frame Galaxy SEDs using Unsupervised Machine Learning

Tesis de Investigación

Paulina Contreras Miranda

164579

Supervisora:

Dra. Milagros Zeballos Rebaza

San Andrés Cholula, Puebla. Otoño 2023

Acknowledgements

I extend my deepest gratitude to those who have offered their support throughout this academic journey. The guidance provided by my professors has been invaluable, shaping my understanding and pushing me to overcome my own expectations. Their mentorship has not only enhanced my academic skills, but also inspired me to be a lifelong student. I am truly fortunate to have been surrounded by such dedicated educators who have invested their time and expertise in my intellectual growth. To my friends, who have been pillars of strength, who made me laugh during challenging times and cheered me on during moments of success and I will be forever thankful for their company. Special thanks to my best friend, Mónica, who will be my forever inspiration. To Edgard, whose love and support have guided me through life. To my parents, whose sacrifices and encouragement have been the bedrock of my achievements. To my brother, who has taught me not to give up. Their support has been the reason that has kept me grounded and motivated. Thank you for being the pillars of my life.

Contents

Abstract

Spectral Energy Distributions (SEDs) represent how the energy of an object is arranged considering different wavelengths. These are useful in astronomical research, as are known as the object's *fingerprints* that bring insights to the characteristics of, e.g. galaxies. This study focuses on building a self-organizing map (SOM) to bring order to a set of rest-frame galaxy SEDs. The aim is to find patterns within thie galaxy zoo, providing insights and te4chniques that are crucial for photometric redshifts. Morevoer, the SOM's outcomes are expected to contribute to the construction of a comprehensive model for the galaxy population, aiming to provide insights and techniques that will be useful for photometric redshifts and more generally to build a model of the galaxy population useful for future cosmological galaxy surveys. Through this work, we intent to offer alternative methods for interpreting and categorizing galaxy spectral profiles.

Keywords: Spectral Energy Distribution, Galaxy, Variarional Autoencoder, Self Organizing Map.

1 Introduction

The study of galaxies has long been a cornerstone of modern astrophysics and cosmology. As telescopes and observational techniques have advanced, astronomers have been able to gather vast amounts of data on the properties and distribution of galaxies across the cosmos. However, making sense of this data requires advanced computational tools and techniques. Machine learning has emerged as a powerful approach for analyzing complex astronomical datasets. In particular, self-organizing maps (SOMs) have shown great promise for identifying patterns and groupings in large datasets of galaxy spectral energy distributions (SEDs).

This work describes the development and implementation of a SOM-based analysis tool for galaxy SEDs aiming to build a more comprehensive and accurate model of the galaxy population, with applications ranging from photometric redshifts to predictions of large-scale structures. It begins with a review of the current state of the physical properties that involve the understanding of the results, as well as knowledge of galaxy formation and evolution. It also provides an overview of machine learning techniques, such as VAE and SOM and describes the development of the SOM-based analysis tool. The work then presents the results of applying these tools to a large dataset of galaxy SEDs, highlighting the insights and patterns that emerge from the analysis. It concludes with a discussion of the implications of the project for astrophysics, cosmology, and machine learning, as well as potential avenues for future research.

1.1 Photometry

Photometry is a branch of astronomy that works with measurements of the brightness and intensities of light emitted by objects in the cosmos, fundamentally studying stars and galaxies. It involves the use of specialized instruments such as photometers, which measure the amount of light received by a detector. These instruments can be used to measure the brightness of a star, or more generally, the distribution of light of an extended object, such as a galaxy or a cluster of stars.

1.1.1 Magnitudes, Flux, and Luminosity

The *surface brightness (I)* ($[I] = ergs^{-1}cm^{-1}sr^{-1}$) is defined as the photon energy received by a unit area from a frame of reference per unit time from a unit solid angle in a specific direction [4]. We can obtain the flux of the object f ($[f] = ergs^{-1}cm^{-2}$) by integrating I over the image. This result is defined as the total amount of energy of all wavelengths that cross a unit area oriented perpendicular to the direction of the light's travel path per unit time [2]. If we now integrate f over a sphere centered on the object with a radius equal to the distance r from the object to the observer, we can obtain the bolometric luminosity L ($[L] = ergs^{-1}$) of the object

$$L = 4\pi r^2 f^1 \ . \tag{1}$$

We can re-write equation 1 solving for the radiant flux

$$f = \frac{L}{4\pi r^2} \ . \tag{2}$$

Since L is independent of r, f is inversely proportional to the square of the distance from the star, a relation which is defined as the *inverse square law* [2]

One of the most common measurements in photometry is the *Apparent Magnitude (m)*, which is defined as the numerical scale that describes the brightness of the stars as seen from our frame of reference: the Earth, by assigning an apparent magnitude $m = 1$ to the brightest star and the dimmest $m = 6$. Now, the scale is extended from $m = -26.83$ (the Sun) to $m = 30$ (the faintest detectable object) with a total range of 57 magnitudes. The difference of 1 magnitude between two stars implies a constant ratio between their brightnesses; then these 57 magnitudes correspond to a brightness ratio of $100^{-57/5} \simeq 10^{23}$ [2].

Another measurement used in photometry is the *Absolute Magnitude (M)*, which refers to the apparent magnitude a star would have if it were located at the distance of 10pc [2]. If we now relate the flux ratio between two stars, we obtain that

$$\frac{f_2}{f_1} = 100^{(m_1 - m_2)/5} \tag{3}$$

[1]For objects at cosmological distances r, it has to be replaced by the luminosity distance

The connection between m, M and the distance of a star is given combining equations 2 and 3:

$$100^{(m-M)/5} = \frac{f_{10}}{f} = (\frac{d}{10pc})^2 \ . \tag{4}$$

Solving equation 4 for d, we obtain that

$$d = 100^{(m-M)/5} pc \ . \tag{5}$$

Another relation between m and M is the distance modulus $(m - M)$, which is defined as the measurement of the distance to a star

$$m - M = 5log_{10}(\frac{d}{10pc}) \ . \tag{6}$$

If we know both values, equation 6 can be used to obtain its distance. If we know the distance to an object, a measurement of its apparent magnitude (or flux) can be used to obtain its absolute magnitude (or luminosity). Now, in order to obtain the total (apparent) magnitude from an image, one must first convert magnitude per unit area into flux per unit area, integrate the flux over the entire image, and then convert the total flux back to a total magnitude. If observations are made for an object in more than one waveband, then the difference between the magnitudes in any two different bands defines a color index (which corresponds to the slope of the SED between the two wavebands).

Let's analyze now two stars that are located at the same distance. The ratio of their radiant fluxes will be equal to the ratio of their luminosities. Let's now assume one of the stars is the Sun, then the relation between a star's absolute magnitude and luminosity is given by

$$M = M_{Sun} - 2.5log_{10}(\frac{L}{L'}) \ , \tag{7}$$

where $M_{Sun} = +4.74$ and $L' = 3.893x10^{26}W$. The star's apparent magnitude is related to f received from the star by

$$m = M_{Sun} - 2.5log_{10}(\frac{f}{f'_{10}}) \ . \tag{8}$$

1.1.2 Wavebands and Bandwidths

In the field of astronomy, different types of EM radiation are associated with different wavebands. Generally speaking, a waveband refers to a range of wavelengths or frequencies

that are associated with a particular type of electromagnetic radiation. Bandwidth, on the other hand, refers to the range of frequencies or wavelengths that can be transmitted over a communication channel.

Let $F_X(\lambda)$ the transmission of the filter that defines the waveband X, $T(\lambda)$ the atmospheric transmission, and $R(\lambda)$ the efficiency with which the telescope plus instrument detects photons, then the observed flux from an object is related to its SED $f\lambda$ by

$$f_X = \int f_\lambda F_X(\lambda) R(\lambda) T(\lambda) d(\lambda) \ . \tag{9}$$

Assuming that F_X has been corrected for atmospheric absorption and telescope efficiency by calibrating the data using standard objects with known F_λ, the observed flux will only depend on the spectral energy distribution (this will be explained later on chapter 1.4) and the chosen filter. Thus, a filter function can be characterized by an effective wavelength, λ_{eff}. A characteristic bandwidth FWHM (full width at half maximum) is then defined as $|\lambda_1 - \lambda 2|$, such that $F_X(\lambda_1) = F_X(\lambda_2) =$ half the peak value of $F_X(\lambda)$ [4].

1.1.3 Magnitude and Color

In photometry, magnitude and color are two important parameters used to describe the brightness and spectral properties of astronomical objects. The flux of an astronomical object in the optical band (and also the near-infrared and near-ultraviolet bands) is usually denoted in terms of apparent magnitude

$$m_x = -2.5 log(f_X / f_{X,0}) \ , \tag{10}$$

where $f_{X,0}$ is the flux in the X band of the star Vega. Now it is more common to use "AB-magnitudes". Letting ν represent the frecuency and c the speed of light, we can write the flux as

$$f_{X,0} = 3.6308 x 10^{-20} ergs^{-1} cm^{-2} Hz^{-1} \int F_X(c/\nu) d\nu \ . \tag{11}$$

The luminosities of objects in waveband X are often denoted as an absolute magnitude: $M_X = \hat{a} 2.5 log(L_X) + C_X$, where C_X is a zero point and L_X is written in units of the solar luminosity in the same band $L_{\odot\ X}$. The Vega absolute magnitudes of the Sun in the UBVRI photometric system have the values in 1.

Band:	*U*	*B*	*V*	*R*	*I*	*J*	*H*	*K*	*L*	*M*
$\lambda_{\rm eff}$ (nm):	365	445	551	658	806	1220	1630	2190	3450	4750
FWHM (nm):	66	94	88	138	149	213	307	390	472	460
$\mathscr{M}_{\odot}$:	5.61	5.48	4.83	4.42	4.08	3.64	3.32	3.28	3.25	–
$L_{\odot}(10^{32}\,{\rm erg/s})$:	1.86	4.67	4.64	6.94	4.71	2.49	1.81	0.82	0.17	–

Figure 1: Filter characteristics of the UBVRI photometric system [4]

Another important concept is the one-dimensional surface brightness profile $I(R)$, which is defined as the surface brightness as a function of the isophotal semimajor axis length R [4]. An isophote twisting happens when the position angle of the semimajor axis changes with radius, then I(R) traces the surface brightness along a curve that connects the intersections of each isophote with its own major axis.

1.2 Spectroscopy

Another branch of astrophysics is spectroscopy, which involves the study of the interaction between light and matter. It is the analysis of the properties of light that is emitted, absorbed, or scattered by matter, allowing astronomers to obtain information about the composition, temperature, density, and motion of astronomical objects, such as stars, galaxies, and interstellar gas. By studying the properties of spectral lines from an object, it is possible to infer the kinematics of the emitting (or absorbing) material. The spectra for objects are obtained from their SEDs, where $f\lambda d\lambda$ and $f\nu d\nu$ are the fluxes received in the elemental wavelength and frequency ranges $d\lambda$ at λ and $d\nu$ at ν. From $\lambda = c/\nu$, it follows that

$$f_\nu = \lambda^2 f_\lambda / c \tag{12}$$

$$f_\lambda = \nu^2 f_\nu / c\,. \tag{13}$$

At optical wavelengths, spectroscopy is performed by guiding the light from an object to a spectrograph where it is dispersed according to wavelength. For the case of long-slit spectroscopy, the object is imaged directly onto the spectrograph slit, resulting in a separate spectrum from each point of the object falling on the slit. Finally, in an integral field unit (IFU) the light from each point within the image of an extended object is led to a different

point on the slit, resulting in a three-dimensional data cube with two spatial dimensions and one dimension for the wavelength. For the case of infrared and radio wavelengths, the incoming signal from a source may be Fourier analyzed in time in order to obtain the power at each frequency. At X-ray wavelengths, the energy of each incoming photon can be recorded and the energies of different photons can be binned to obtain the spectrum.

A galaxy spectrum usually contains a slowly varying component called *the continuum*, with localized features produced by emission and absorption lines. This is defined as the superposition of the spectra of all the individual stars in the galaxy, modified by emission and absorption from the gas and dust lying between the stars [4]. The continuum has different behaviours depending on the spectra:

- From the ultraviolet through the near-infrared, the continuum is due primarily to boundâfree transitions in the photospheres of the stars.
- In the mid- and far-infrared it is dominated by thermal emission from dust grains.
- In the radio it is produced by diffuse relativistic and thermal electrons within the galaxy
- In the X-ray it comes mainly from accretion of gas onto compact stellar remnants or a central black hole.

The emission and absorption lines are produced by boundâbound transitions within atoms, ions and molecules, both in the outer photospheres of stars and in the interstellar gas. The emission lines can be used to measure temperature, density and chemical composition of interstellar gas and its strength depends on the abundance of the excited state that produces it, which depends on the abundance of the corresponding element, as well as on the temperature and ionization state of the gas. Absorption lines, on the other hand, arises in the atmospheres of stars and contain information about the age and metallicity of the galaxyâs stellar population. In addition, interstellar dust gives rise to continuum absorption with characteristic features. Since dust extinction happens more often at shorter wavelengths, it causes reddening, i.e. a change of the slope's continuum emission.

The intrinsic frequency of photons produced by electron transitions between the energy levels $E1$ and $E2$: $E_2 > E_1$ $\nu_{12} = \frac{E2\hat{a}E1}{h_P}$ and supposing these photons are produced by atoms moving with velocity v relative to the observer. Consider the case for $v < c$, because of the Doppler effect, the observed photon frequency is given as

$$\nu_{obs} = (1 - \frac{\mathbf{v}\hat{\mathbf{r}}}{c})\nu_{12} \ , \tag{14}$$

where $\hat{\mathbf{r}}$ is defined as the unit vector of the emitting source relative to the observer. If the source is receding from the observer, the observed frequency is **redshifted**, $\nu_{obs} < \nu_{12}$, whereas if the source is approaching the observer, the observed frequency is blueshifted, $\nu_{obs} > \nu_{12}$. Let's now consider the Doppler effect and define the redshift as

$$z = \frac{\mathbf{v}}{c} \ . \tag{15}$$

The redishdt is the define as the difference between the wavelength of the star and the wavelength at redshift 0. By increasing redshift (distance), the lines are shifted to longer wavelength

$$z = \frac{\lambda - \lambda_0}{\lambda_0} = \frac{\delta\lambda}{\lambda_0} \tag{16}$$

$$1 + z_{pec} = \sqrt{\frac{1 + v/c}{1 - v/c}} \tag{17}$$

$$1 + z_{obs} = (1 + z_{pec})(1 + z_H) \tag{18}$$

$$v_r = H_0 d + v_{pec} \ . \tag{19}$$

We can also define the redshift parameter to characterize the change in frequency as

$$z \equiv \frac{\nu_{12}}{\nu_{obs}} - 1 \ . \tag{20}$$

Spectroscopy can be studied in different manners. Supposing the emitting gas atoms in an object have random motions along the line-of-sight drawn from a velocity distribution $f(v)dv$. The frequency distribution of the observed photons is given as

$$F(\nu_{obs})d\nu_{obs} = f(v)(c/\nu_{12})d\nu_{obs} \ , \tag{21}$$

where $v = c(1 - \nu_{obs}/\nu_{12})$. By observing the emission line profile in frequency space $F(\nu_{obs})$, we can infer $f(v)$. If the random motion is caused by thermal effects, we can infer the temperature of the gas from the observed line profile.

For the case of a stellar system, the observed spectral line is the convolution of the original stellar line profile $S(\nu)$ with the line-of-sight velocity distribution of all the stars in the observational aperture

$$F(\nu_{obs}) = \int S[nu_{obs}(1 + v/c)]f(v)dv \ . \tag{22}$$

Each spectral line is broadend by a line-of-sight velocity disperson of stars that contribute to that line, as we can see in figure

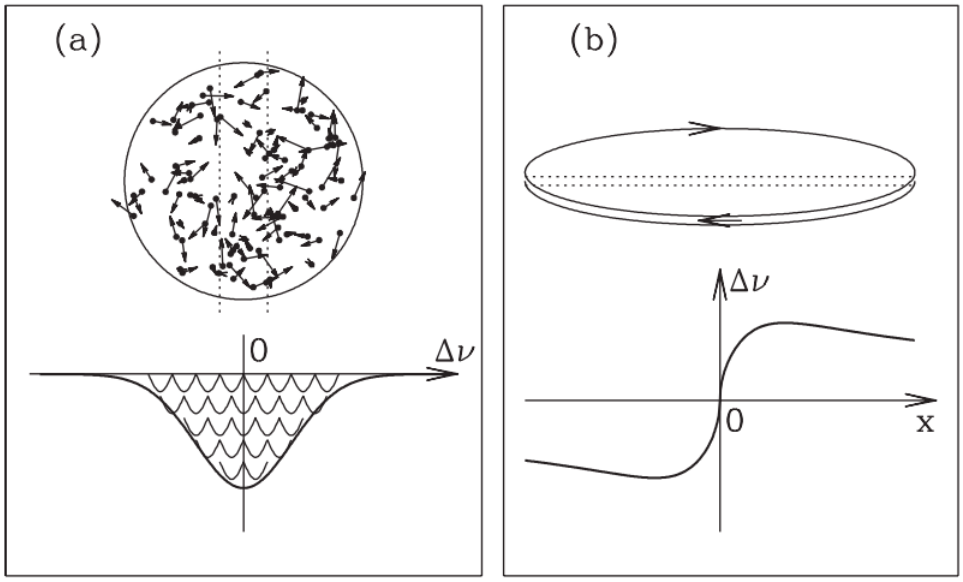


Figure 2: (a) Broadening of a spectral line by the velocity dispersion of stars in a stellar system. (b) long-slit spectroscopy of a thin rotating disk along the major axis of the image [4].

The stellar spectral line is broadened by the line-of-sight velocity dispersion of the stars that contribute to that line. If we know the type of stars that dominate the spectral lines in consideration, we can estimate $S(\nu)$ and use the above relation to infer the properties of $f(v)$.

When analyzing the spectra of galaxies, it is possible to analyze their composition, redshift, velocity dispersion, and therefore dynamical mass, metalicity, and age. Considering these parameters, it is possible to obtain the star formation history.

1.3 Galaxy Population

Galaxies are the basic building blocks of the universe. Their study is an important area of astronomy because they provide clues to the evolution of the universe. By studying their properties, astrophysicists can gain insights into the formation and evolution of stars, the structure of the universe, and the nature of dark matter and dark energy.

The main classifications of galaxies based on morphology are the Hubble sequence and the Vaucouleurs clasification. Hubble's diagram classifies the galaxies into four classes, as seen in figure 3. The first ones are **elliptical galaxies**. They have smooth, almost elliptical isophotes, are divided into subtypes $E0, E1, ..., E7$,where the integer is the one closest to $10(1\hat{a}b/a)$, with a and b the lengths of the semimajor and semi-minor axes. Another type are **spiral galaxies**, which have thin disks with spiral arm structures. These are divided into barred spirals and normal spirals. They are further divided into three classes, a, b, and c, according to the fraction of the light in the central bulge, the tightness with which the spiral arms are wound and the degree to which the spiral arms are resolved into stars, HII regions and ordered dust lanes. [2] The third clasification are **lenticular or S0 galaxies**, which are an itermediate class between ellipticals and spirals. These have smooth light distribution with no spiral arms or HII regions and a thin disk and a bulge, but the bulge is more dominant than that in a spiral galaxy. [3] Finally, **irregular galaxies** have neither a dominating bulge nor a rotationally symmetric disk and lack any obvious symmetry. Irregulars are usually included as an extension to the spiral galaxies.

[2]Spirals with a pronounced bulge component usually also have tightly wound spiral arms with relatively faint HII regions, and are classified as Sa. Spirals with weak or absent bulges usually have open arms and bright HII regions and are classified as Sc. When the three criteria give conflicting indications, Hubble put most emphasis on the openness of the spiral arms.

[3]They may also have a central bar, in which case they are classified as SB0.

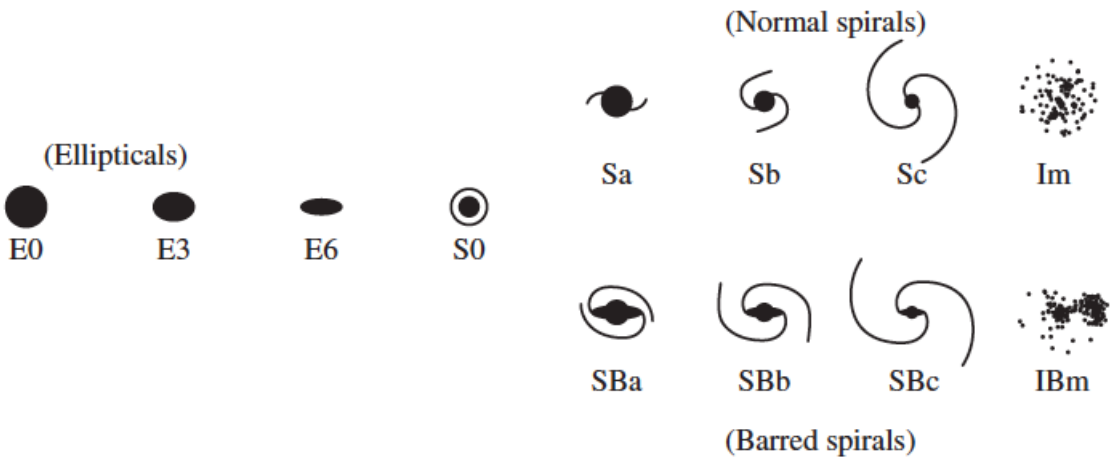


Figure 3: Hubble sequence of galaxy morphologies [4]

The de Vaucouleurs adds new sub-classifications of the spirals in the Hubble sequence, such as S0a, Sab, and Sbc (and the corresponding barred types). It also extended spirals to irregulars and classifies them from their decreasing regularity: Scd, Sd, Sdm, Sm, Im, and I0 and represent morphological types denoting them from â6 to 10. Moreover, galaxies can also be classified into bright and faint according to luminosity, into high and low surface brightness according to surface brightness, red and blue according to color, gas-rich, and gas-poor according to gas content, into quiescent and starburst according to their current level of star formation, and into normal and active according to the presence of an active nucleus.

1.4 Star-formation in galaxies

In general, the observation and definition of a galaxy revolve around its stellar content. As a result, any comprehensive theory of galaxy formation must tackle the fundamental question of how stars come into existence. Understanding the process of star formation is essential for elucidating the formation and evolution of galaxies as a whole. By studying the mechanisms and conditions under which stars form, we can gain insights into the intricate interplay between the formation of individual stars and the emergence of galaxies as coherent systems.

Baryonic gas within galaxy-sized halos has the ability to cool down in a relatively short

time compared to the age of the halo itself. This cooling process causes the gas to lose its pressure support, leading it to migrate toward the center of the halo's potential well. Consequently, the density of the gas increases as it continues to flow inwards. Eventually, the gas reaches a point where its density surpasses that of the surrounding dark matter in the central region of the halo. At this stage, the cooling gas becomes self-gravitating and collapses under its own gravitational pull. Due to this, self-gravitating gas, especially in the presence of efficient cooling, can undergo a catastrophic collapse. This collapse can eventually result in the formation of dense, cold gas clouds within which the process of star formation takes place.

Indeed, achieving a comprehensive understanding of star formation within a cosmological framework is an immensely challenging task, but we can study the star formation rates in different types of galaxies. In actively star-forming galaxies, such as spiral galaxies, star formation is often concentrated in regions of high gas density, such as spiral arms or central starburst regions. These galaxies contain abundant reservoirs of molecular gas and experience ongoing star formation. In contrast, elliptical galaxies, which are generally older and have lower levels of gas, exhibit lower star formation rates. However, some elliptical galaxies can still undergo episodic bursts of star formation triggered by mergers or interactions with other galaxies.

1.5 Spectral energy distributions

The radiation from a source can be characterized by the total energy of emitted photons with wavelengths from λ to $\lambda + \mathrm{d}\lambda$. This is called the spectral energy distribution (SED); the photometric observations are generally carried out in some chosen waveband [4]. The dominant sources of radiation at energies below hard gamma-ray (except the CMB) are related to the evolution, clustering, and nuclei of galaxies. The galactic starlight is present in the near-UV, optical, and near-infrared, and dust emission from star-forming galaxies is responsible for (most of them) far-infrared emission. The hot gas in galaxy clusters emits a significant but non-dominant fraction of the total X-ray background and is the only major source of emission from scales larger than an individual galaxy. These structures can be seen

in absorption.

To characterize the overall shape of the SED of an object with high efficiency, a "broad-band photometry", for example, λ_{eff} and FWHM for the filters of the standard UBVRI photometric system, in which the FWHM are all of order 10 % or larger of the corresponding λ_{eff}. Also, one can use "narrow-band photometry" with much narrower filters to image objects in a particular emission line or to study its detailed SED properties.

1.6 Stellar population synthesis models

The light emanating from a specific galaxy originates from a diverse population of stars, each potentially possessing distinct characteristics such as varying masses, ages, and metallicities. To comprehensively interpret the observed spectral energy distribution (SED), it becomes necessary to estimate the individual contributions of these stars toward the overall spectrum [4].

Given an understanding of the theory of stellar evolution, it is possible to synthesize the spectrum of a galaxy at any specific moment by combining the spectra of individual stars, which evolved at that particular time. This synthesis can be accomplished by incorporating the star-formation history (i.e., the rate at which stars are formed as a function of time) and the initial mass function (IMF) of the galaxy. By accounting for these factors, we can calculate the combined spectrum that arises from the contribution of all stars within the galaxy.

The majority of the energy emitted by a stellar population is concentrated in the optical range. In the case of very young stellar populations (with an age of less than approximately 10 million years), a significant portion of the energy is emitted in the ultraviolet range. However, if a galaxy contains a substantial amount of dust, a considerable fraction of the optical and ultraviolet light can be absorbed by the dust particles and re-emitted in the infrared range. Predicting the resulting emergent spectrum in the presence of dust is a highly intricate task. It not only relies on the amount of radiation absorbed but also heavily depends on various properties of the dust, including its geometry, chemical composition, and

the distribution of dust grain sizes.

1.7 Dust in galaxies

Dust plays a significant role in the interstellar medium (ISM) of galaxies, influencing their appearance, chemistry, and evolution. Interstellar dust consists of tiny solid particles, typically composed of carbon, silicates, and other heavy elements, which are distributed throughout the galaxy

The evolution of stars can be viewed as a cyclic process that involves the interplay between stars and the interstellar medium (ISM). Stars are born from the gas and dust present in the ISM, which is the material that exists between the stars in a galaxy. Throughout a star's lifetime, depending on its mass, a significant portion of its material can be returned to the ISM through processes such as stellar winds and explosive events like supernovae [2].

The recycled material from previous generations of stars becomes mixed with the existing ISM, enriching it with heavier elements and altering its composition. This processed material then serves as the building blocks for the formation of subsequent generations of stars. Therefore, understanding the nature and properties of the ISM is crucial for comprehending the evolution of stars. However, the significance of the interstellar medium extends beyond its role in stellar evolution. It plays a vital role in describing the structure, dynamics, and evolution of not only our own Milky Way Galaxy but also galaxies throughout the universe. The properties of the ISM impact our observations of various astronomical objects, ranging from nearby stars to the most distant galaxies and quasars. The ISM affects the propagation of light, absorption and emission processes, and the overall appearance of celestial objects.

1.8 Photometric redshifts

Spectroscopy, which disperses light according to wavelength, is the preferred method for obtaining accurate redshift measurements. However, it is limited to relatively bright objects due to the need for sufficient signal-to-noise in individual emission and absorption lines. For fainter objects, an alternative technique involves measuring broad-band photometry, which

provides a rough sampling of the object's spectral energy distribution (SED).

Galaxies exhibit broad spectral features in their SEDs, such as the 4000Ã break caused by a change in opacity in the atmospheres of low-mass stars. These features can be used to estimate the redshift of a galaxy. By comparing the expected fluxes in different wavebands based on a template spectrum (either observed or computed), with the observed fluxes, the best-fit redshift and template spectrum can be determined [4].

Photometric redshifts have the advantage of being faster to measure compared to spectroscopic redshifts and can be extended to much fainter magnitudes. However, they are less reliable. Photometric errors typically range from 3% to 10%, depending on the number and selection of wavebands used. The error is also correlated with the spectral type of the galaxy, being larger for star-forming galaxies with less pronounced spectral features compared to galaxies with an older stellar population.

1.9 Justification

The justification for this project lies in the increasing importance of large astronomical surveys in advancing our understanding of the universe. These surveys provide a wealth of data on galaxies across the cosmos, from their distribution and properties to their evolution over time. However, making sense of this data requires advanced computational techniques that can handle the complexity and scale of the datasets involved. Machine learning approaches, in particular, have shown great promise in tackling these challenges, allowing researchers to identify patterns and trends in the data that would be difficult to discern using traditional methods. The development of a self-organizing map to analyze galaxy SEDs is an example of how machine learning can be used to advance our understanding of the universe. By identifying common patterns and groupings in the SEDs, this approach can help us to build a more comprehensive and accurate model of the galaxy population. By participating in this project, the ACAI group at LMU is at the forefront of this exciting and rapidly evolving field. The project not only provides valuable insights into the properties and evolution of galaxies but also contributes to the broader goal of advancing our understanding of the universe using cutting-edge computational techniques.

2 Models and numerical tools

The input data for this project consists of a spectral energy distribution (SED) library of single stellar populations (SSPs) generated by the Flexible Stellar Population Synthesis (FSPS) model, which provides information about the distribution of energy across various wavelengths for different types of stellar populations. This serves as a valuable resource for investigating the diverse range of stellar populations and their associated SEDs.

2.1 FSPS

FSPS (Flexible Stellar Population Synthesis) is a software package designed to generate synthetic stellar populations for astrophysical research [3]. Its main purpose is to generate synthetic stellar populations that can be used to model the spectral energy distribution (SED) of real astronomical objects, such as galaxies, stars, and quasars. FSPS uses a library of pre-computed stellar spectra and a set of physical models to generate these synthetic populations.

One of the key features of FSPS is its flexibility. It allows users to vary a wide range of parameters, including the age, metallicity, and initial mass function (IMF) of the stellar population, as well as the dust content and other physical properties. This makes it a powerful tool for exploring different astrophysical scenarios and testing hypotheses about the nature and evolution of galaxies and other astronomical objects.

For the analysis of our galaxy SEDs, we encounter spectra and wavelengths that are defined by numerous parameters. Consequently, it is crucial to perform pre-processing steps before constructing the final outcome. Our objective is to visualize this information effectively within a Self-Organizing Map (SOM). To achieve this, we must reduce the dimensionality of the parameters that characterize the spectra and wavelengths of each galaxy. To accomplish this step, we will employ unsupervised machine learning techniques, specifically variational autoencoders. These models will enable us to capture and represent the essential features of the data, facilitating the visualization and interpretation of the galaxy catalog within the SOM framework.

2.2 Auto Encoders

An autoencoder is a specific type of neural network, which is mainly designed to encode the input into a compressed representation, and then decode it back, such that the reconstructed input is as similar as possible to the original one [1]. The network consists of two main parts: the encoder, which takes the input data to compress it into a so-called latent space, and a decoder, which takes the latent space as the input, producing a decompressed representation of the original data.

Variational Autoencoders (VAEs) is then a type of generative model based on this type of architecture. VAEs are a method of compressing the learning of an identity map in order to find a lower-dimensional representation of a dataset considering how the identity map is learned. These constraints lead to a lower-dimensional space of the original data, called the **latent space**. The resource that will be applied to build the VAE is mainly *keras*, a python library that is used for artificial neural networks [5].

A VAE consists of an encoder and a decoder; the encoder takes an input sample and converts its information into some vector. The decoder, on the other hand, takes this vector and expands it out to reconstruct the input sample. In the VAEs the goal is not to reconstruct the information, but rather the latent space. It's important because it is a representation of the input information [7]; this can be seen in Figure 4.

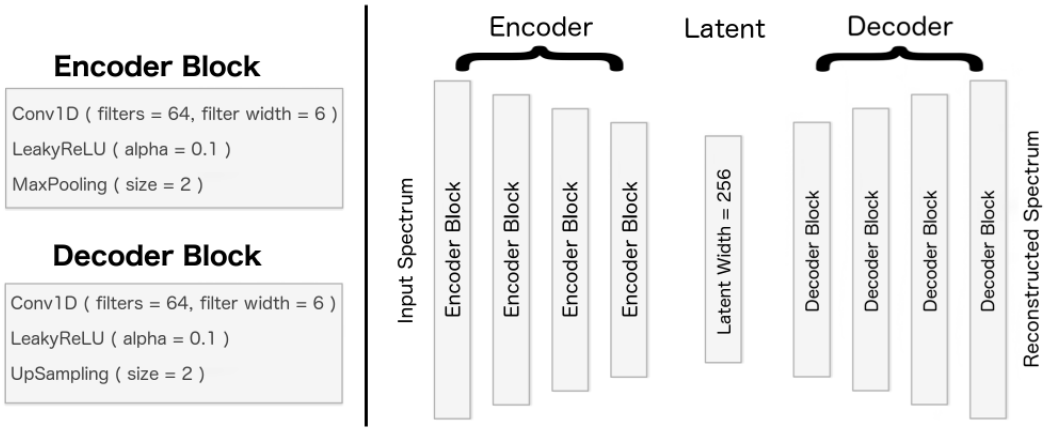


Figure 4: The architecture of the autoencoder component uses convolutional blocks. The encoder embeds data to the latent layer. The decoder takes the latent layer and attempts to reconstruct the original input [6].

This algorithm is mainly characterized by the following properties:

- Data-specific: Autoencoders are only able to meaningfully compress data similar to what they have been trained on. Since they learn features specific to the given training data, they are different than a standard data compression algorithm.
- Lossy: The output of the autoencoder will not be exactly the same as the input, it will be a close but degraded representation.
- Unsupervised: they are considered an unsupervised learning technique since they donât need explicit labels to train on. But to be more precise they are self-supervised because they generate their own labels from the training data. Before trainig the model, we should consider the folliwing:
- Latent space: number of nodes in the middle layer. Smaller size results in more compression (in our case: 256)
- Number of layers: the autoencoder can be as deep as we like. In figure 4 we have 4 layers in both the encoder and decoder.

- Number of nodes per layer: the autoencoder architecture weâre working on is called a stacked autoencoder since the layers are stacked one after another. Usually stacked autoencoders look like a âsandwichâ. The number of nodes per layer decreases with each subsequent layer of the encoder and increases back in the decoder. The decoder is symmetric to the encoder in terms of layer structure.

- Loss function: we either use mean squared error (MSE) or binary cross entropy. If the input values are in the range [0, 1] then we typically use crossentropy, otherwise we use the mean squared error (In our case: MSE)

2.2.1 Keras Layers

Layers are the basic building blocks of neural networks in Keras. A layer consists of a tensor-in tensor-out computation function (the layer's call method) and some state, held in TensorFlow variables (the layer's weights). [1]

Layer Activations

Activations can either be used through an Activation layer, or through the activation argument. There are different activations available. The *relu function*, for example, applies the rectified linear unit activation function. With default values, this returns the standard ReLU activation: max(x, 0), the element-wise maximum of 0, and the input tensor. Modifying default parameters allows to use of non-zero thresholds, changes the max value of the activation, and uses a non-zero multiple of the input for values below the threshold.

The building blocks of the Keras library are called **Keras Layers**; this can be stacked together in order to create neural network models. There are different types of Keras layers available for different purposes while designing your neural network architecture. The *dense layer* is used for creating a deeply connected layer in the neural network where each of the neurons of the dense layers receives input from all neurons of the previous layer. At its core, it performs dot product of all the input values along with the weights for obtaining the output. The *flatten layer* is used for flattening the input.

In ReLU the negative part is totally dropped, while in Leaky ReLU assigns a non-zero

slope to it. The Leaky ReLU has the ability to retain some degree of the negative values that flow into it. This extended output range slightly higher flexibility to the model. Incorporating a nonzero slope for a negative part in Leaky ReLU improves the results.

Another important parameter is the *epochs*, which will be the one responsible for determining the number of iterations for training the model. To fit the model, a function called $fit_generator()$ will be applied, as a large set of data is being used. In this function, a Python generator is used for loading the data into memory using batches of data during the training phase.

The model will be trained to minimize a certain loss function which ensures that the output is close to the input. If the dimension of the hidden layer(h) is less than the dimension of the input layer, then we call it under complete autoencoder. On the contrary, if the dimension of the hidden layer(h) is greater than the dimension of the input layer, then we call it over the complete autoencoder.

Now that we have learned about autoencoders, we can explore the importance of Self-Organizing Maps (SOMs). It is important to recall that will use the compressed representation obtained from the variational autoencoder, known as the latent space, as the input data for our SOM. By combining these techniques, we can visualize and analyze the galaxy catalog in a more intuitive way. The SOM will help us identify clusters, spatial relationships, and patterns that may not be easily discernible in the raw data. This integrated approach offers a powerful method for gaining valuable insights into the characteristics and distributions of the galaxies we are studying.

2.3 Self Organizing Maps (SOM)

Self-organizing maps are a type of artificial neural network that performs unsupervised dimensionality reduction by building a map of the data. These can be visualized as non-parametric competitors to classical dimensionality reduction techniques. SOMs are able to convert complex, nonlinear statistical relationships between high-dimensional data items into simple geometric relationships on a low-dimensional display. The main advantage of SOM

is its ability to reduce the dimensionality of input data while preserving the relationships between data points, thus making those relationships easier to visualize [8]

Let us think about each neuron as an agent. These agents will compete with each other and will try to span the data clusters, so they self-organize themselves to cover the data. SOMS work from competitive learning using neural networks. The rates of each neuron are learned by interactions, which will be updated during the process of the creation of the map. Each neuron, which lives in a 2-dimensional grid, shall have similar rates as its nearest neighbors, a property that is preserved while mapping to the data.

2.3.1 Network topology: neighbors within a SOM

The SOMs work by connecting the input data to every node (also called a cell), where every one of these represents a weight vector, producing together a multidimensional lattice (or a map) over a space (in this work we will perform a two-dimensional lattice). This coordinate is said to be *mapped* to a weight, i.e. the neuron will be mapped to the data set, whose dimensions shall be the same as the weight vectors.

To visualize this process, let us imagine a 2-dimensional grid with nxm neurons and focus on a particular neuron α with coordinates $n_i x m_j$. The neighborhood of this neuron will be all the neurons that are adjoined to the one that is being analyzed. In other words, the neighborhood of the α neuron is those neurons, whose Euclidean distance to α is small enough.

Now, we want to preserve the neighborhood function, i.e. the topology of the system. At the beginning of the process, the neurons have random weights. While the algorithm works, the neurons will be affected by their neighbors and their weights will therefore become similar. Because we are working with a grid space and a data space, if we neighbor the grid space, we will expect the elements to be neighbors also in the data space also because there is a mapping between these both.

To exemplify this process, let again the neuron α be close to another neuron α' in terms of their coordinates in the grid. At the end of the process, we will expect that the weights

of both the α and α' neurons will also be close. So, neurons that are neighbors in the 2-dimensional grid space (that have similar coordinates/location) are also mapped to similar weight vectors in the data space. This is called the *preserving topology*.

SOM's training uses mainly **competitive learning**, a form of unsupervised learning, where constituent elements compete to produce a satisfying result, and only one gets to win the competition. When a training example is input into the grid, the **Best Matching Unit (BMU)**, i.e. the cell whose weights are closest to the trainig example (the winner), is determined, meaning that this input *wins* the competition and therefore gets to assign its weights to the input and return that as output. This process is called a **topographical map**

The *BMU* is determined using a discriminant function: the Euclidian distance between the node weights and the input vector (see equation 23)

$$d_j(x) = \sum_{i=1}^{p} (x_i - w_j i)^2. \tag{23}$$

This training is performed using topological ordering: a training sample is given as an input to the network, which picks the node that has the smallest distance to the input. Then, the nodes shift their weights towards the input value by a certain *learning rate*. The neighborhood of nodes closest to that node also shifts their location, where the strength of the movement depends on that node's distance to the winning node, subject to a learning rate. As the SOM is trained, the learning rate is annealed. In other words, the nodes located within a close neighbor of each other shall have similar weights and therefore can be mapped to nodes in a similar neighbor of each other. Each *update* can be thought of as a shift in the topology of a neighborhood local to the output point. With sufficiently many iterations through this process, the nodes will form the dimension-reduced feature space over the original input which best preserves local topology.

3 Results and analysis

3.0.1 Statistical properties

The catalog considered for this project is based on the spectral energy distribution (SED) of a single stellar population (SSP). It consists of many measurements that depend on the flux of each of the simulated galaxies; each of these parameters can be considered as a *dimension*. To fully make use of all the properties, it is necessary to consider as much information as possible about its spectra, which is made by calculating statistics from each spectrum that are related to the physical properties.

LetÂ´s first recall the information on our database. It consists of 10,000 galaxies, each of them having certain parameters that describe them. The ones concerning the data processing are the spectra and wavelength, each of them being described by 4389 points. In order to process all this information, the dimension of the data has to be reduced, and here is where we implement the machine learning methods.

Coming back to our catalog, the properties of the SSPs are independently and uniformly sampled. We will initially work with the first 1,000 galaxies and their respective parameters. The general properties and sampling for each galaxy are the following:

- redshift: fixed at 0.496
- age: uniform sampling in range [0, 8.45], Gyr
- metallicity: uniform sampling in range [-0.5, 0.2], $log(Z/Z_{solar})$
- velocity dispersion: uniform sampling in range $[100, 300]$, km/s
- gas ionization: None
- gas metallicity: None
- stellar mass: uniform sampling in range $[10^8, 10^{12}]$, M_{solar}

Now that we have reviewed the basics of our database, let's proceed with the implementation of the machine learning algorithms employed for processing the galaxy SEDs.

3.1 Autoencoder implementation

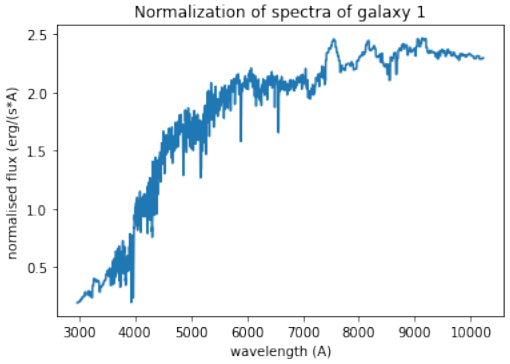
The first section of the code reads a FITS file, which contains our library of spectra of 10,000 rest-frame galaxies. From this catalog, we will initially work with the first 1,000 galaxies. The spectra and wavelength of each galaxy are defined by 4389 parameters. Thus, we will start preparing our spectrum and wavelength by first resampling the spectra onto a wavelength grid with a fixed length of 4096 parameters by generating a new wavelength grid with np.arange() and printing the length of the grid. This particular number is chosen because we will work with multiples of 2^n ($4096 = 2^{13}$), as numerically it is easier to process information described by base-2 parameters. This is done by generating a new wavelength grid using the *np.arange()* function. In order to obtain the respective wavelengths of the new resampled spectrum, we use the *Flux Conserving Resampler* function from the*specutils library* to resample each spectrum onto the new wavelength grid while conserving the overall integrated flux by defining a new wavelength grid and resampling each spectrum onto this new grid. The resampled spectra and resampled wavelength grids now with 4096 parameters are stored in order to be used later on.

Afterward, we normalize of the spectra by building a function that performs normalization on the input spectra based on the procedure described in the paper [6]. It takes as input the wavelength and spectra arrays and first selects the wavelength range of the visible region (4050-4150 $\circ A$). Then, for each spectrum, it calculates the mean flux of that wavelength range using *np.nanmean*, and divides the entire spectrum by this mean flux. Finally, it reshapes the spectra array and returns it. The purpose of this normalization is to account for differences in flux calibration between different spectra.

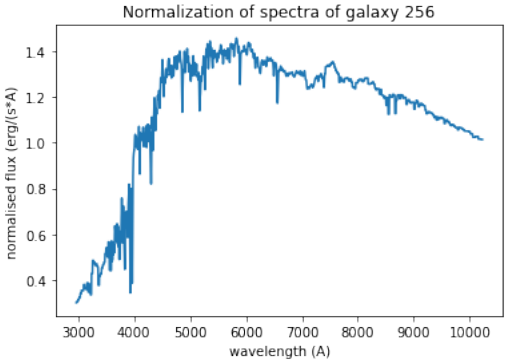
Now we are done with the preparation of our spectra. We start the autoencoder by defining the training sets. We first split the resampled spectra into three sets: training, validation, and testing, which contain 80 %, 10 % and 10 % of the total number of galaxies, respectively. The total number of spectra is the number of galaxies that will be analyzed. As we established, we will start working with 1,000 galaxies. Then, we define the number of wavelength bins and the number of channels in he spectral lines. We will work only with one channel, as the spectra is one-dimensional. After that, we initialize the three

arrays to store the training, validation, and testing sets of the spectra. These arrays are initially empty, but specify their dimensions (number of spectra, number of wavelength bins, and number of channels). Having defined the sets, we can now assign the spectra to each set. The purpose of this splitting is to train the deep learning model on the training set, tune its hyperparameters on the validation set, and evaluate its performance on the testing set. Considering the training set (which contains 800 galaxies, each of their spectra and wavelength described by 4096 parameters), we will plot the normalized flux of a single spectrum. Let's keep in mind that these have been divided by their mean flux in the range 4050-4150, therefore the normalzed part of the flux shall be around that region. We choose to plot six galaxies from this set, the first and last galaxy and four random ones in between the set in order to confirm that te normalization function is working appropiately.

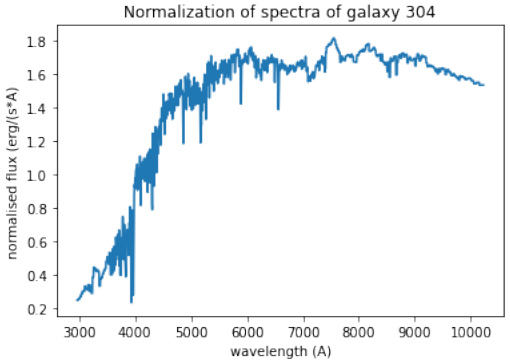
We can now confirm that our normalization function is working appropriately, as we notice how the wavelength around the visible region of every randomly chosen galaxy matches the value 1 in the normalized flux.



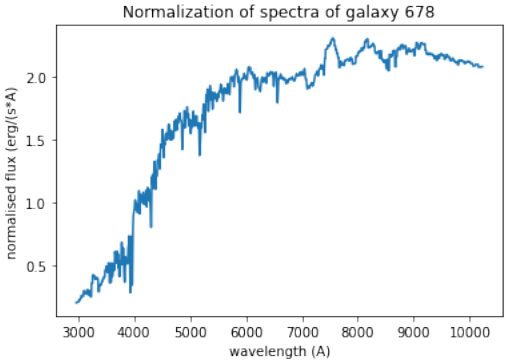
(a) Galaxy 1



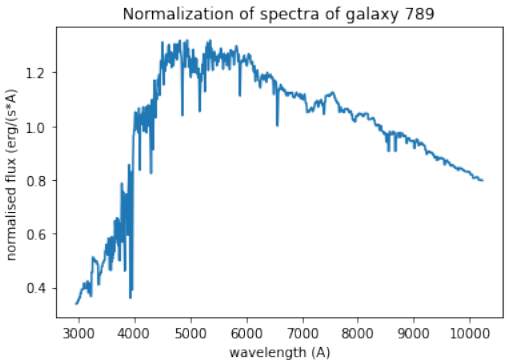
(b) Galaxy 256



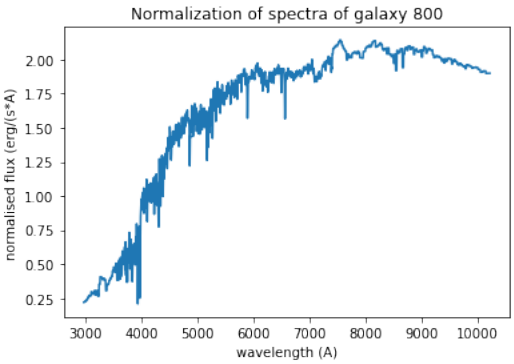
(c) Galaxy 304



(d) Galaxy 678



(e) Galaxy 789



(f) Galaxy 800

Figure 5: Normalized flux of a single spectrum of galaxies 1, 256, 304, 678, 789, and 800 around the visible range.

Now, it is possible to start building the autoencoder. We first define the building blocks of our convolutional autoencoder for 1D spectra data as follows:

- *encoder block*: This function takes in the input *encoder inputs*, applies a 1D convolutional layer with *number* of filters and *kernel size*, with an activation function *activation* and *strides*. It then applies *max pooling* to downsample the data and reduce its length while preserving the features. The output is then returned.

- *decoder block*: This function takes in the input *encoder inputs*, applies a 1D transposed convolutional layer with *number* of filters and *kernel size*, with an activation function *activation* and *strides*. It then applies upsampling to increase the length of the data by repeating each element in the input sequence *kernel size* number of times. The output is then returned.

- *final block 1D*: This function applies a 1D convolutional layer with *number* of filters and *kernel size*, with an activation function *activation* and *strides* and *padding*. The output is then returned. This function is used as the final layer of the decoder block to generate the output.

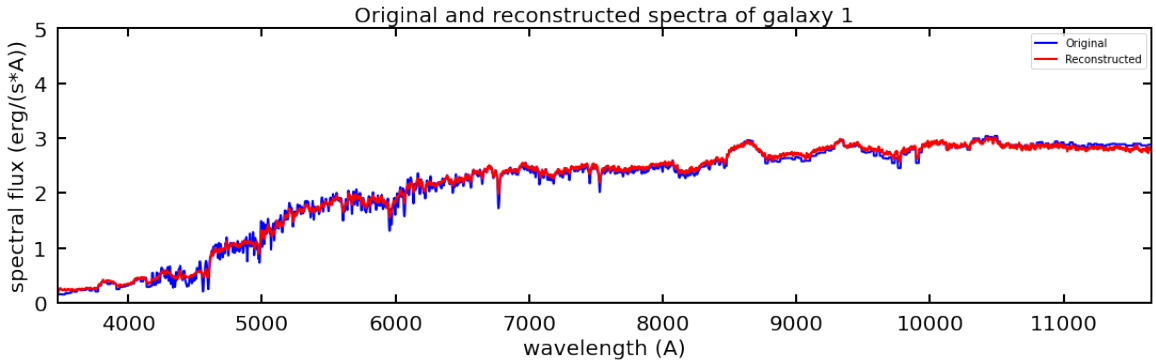
Having stated how our encoder and decoder blocks will work, we then define a function *get model 1D* that creates a convolutional autoencoder model for 1-dimensional input data using the encoder, decoder, and final blocks defined earlier. The function takes several arguments, including the number of input spectra, the number of filters used in the convolutional layers, the number of latent variables, and the activation function used in the convolutional and dense layers. This model architecture consists of a series of convolutional layers with downsampling (encoder block) and upsampling (decoder block) operations to learn and reconstruct the spectral features. The architecture includes a bottleneck layer with latent-variables neurons (the latent space) that reduces the dimensionality of the input data. The activation function we use in the convolutional and dense layers is LeakyRelu. Here, we aim to obtain as an output of the model the reconstructed spectra. In the end, the function will return the entire autoencoder model and the encoder model. It is worth explaining the latent space (the number of neurons or latent variables) deeper, as it will be the output we will be working with. This represents the bottleneck layer of the autoencoder, where the input data is compressed into a lower-dimensional latent space. This specifies the number of variables that will describe the wavelength and spectrum of each of our 1,000

galaxies. In this case, we followed the structure as in the paper [6], which defines a latent space of 256 variables. Again, we are choosing to work with parameters according to 2^n: $256 = 2^8$. To recall, we are reducing the dimensionality of the wavelength and spectrum of each of the 1,000 galaxies from 4096 to 256 parameters each.

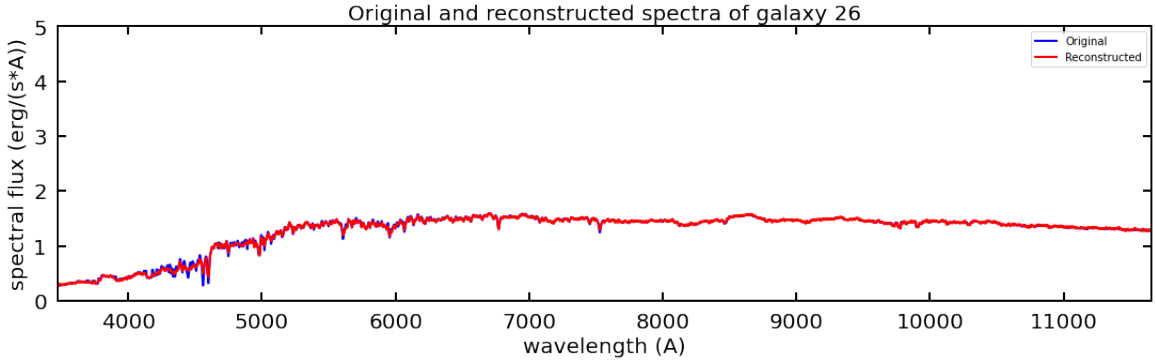
Now we start the training process of the autoencoder. We define and compile a 1-dimensional convolutional autoencoder model for processing spectral data. We first clear the Keras backend session to free up any memory that might have been allocated by previously defined models. Then, we call the get-model-1D function to define the architecture of the autoencoder model. This function returns two models, one for the encoder and one for the entire autoencoder, and the script assigns the autoencoder model to cnn-spec1D and the encoder model to encoder-spec1D. Finally, we print out a summary of the architecture of the autoencoder model using the *summary()* method and compile the model using the Adam optimizer. We choose a learning rate of 0.001 after having tested the VAE with multiple values, concluding that this value is the one that fits our data best. For the loss function, we choose the mean squared error (MSE), based on the paper [6]. We train the autoencoder model for the first 5 epochs with a batch size of 100 and use mean squared error as the loss function. This number of epochs gives us noisy reconstructed spectra, thus, we increment the number of epochs until we got an acceptable result with 50 epochs. We also applied an early stopping by monitoring the validation loss. As the training loss and validation loss are decreasing over the epochs, we can say the model is having a good performance. This indicates that the model is able to learn and generalize the patterns in the data. The final validation loss is very low, which means that the model is able to accurately reconstruct the input spectra.

To visualize our traning (the results of the VAE), we first define a function that generates a comparison plot of the original and reconstructed spectra, where the x-axis is the wavelength and the y-axis is the flux. We choose to plot six galaxies from the training set established previously (conformed of 100 galaxies), the first and last galaxies of the set, and four random ones in between. This, in order to confirm that the reconstructed spectra match the original one. In other words, to confirm that our autoencoder is reducing the dimensionality of the

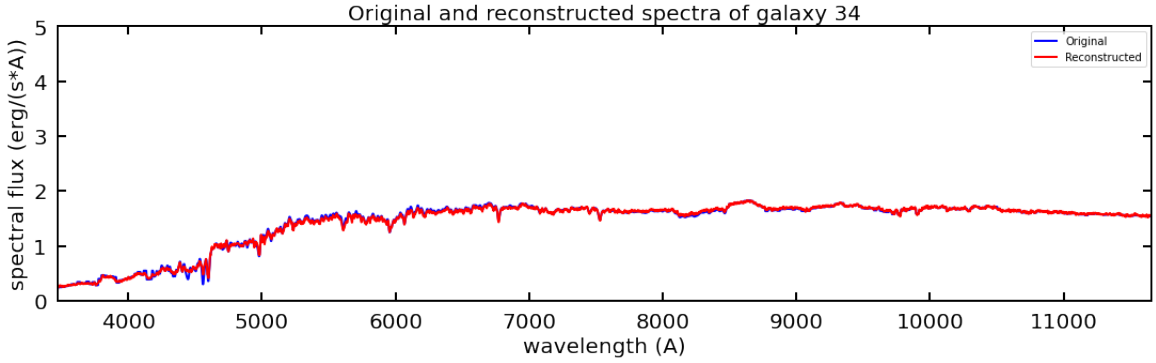
wavelength and spectra appropriately, having a minimum error.



(a) Galaxy 1

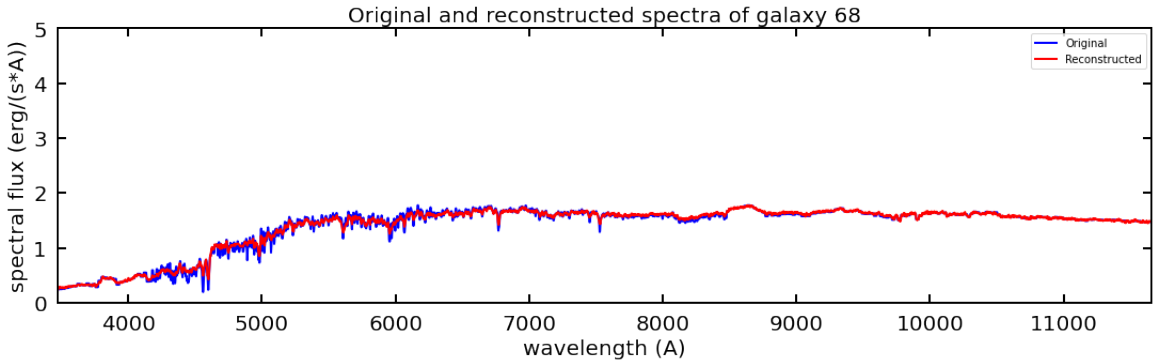


(b) Galaxy 26

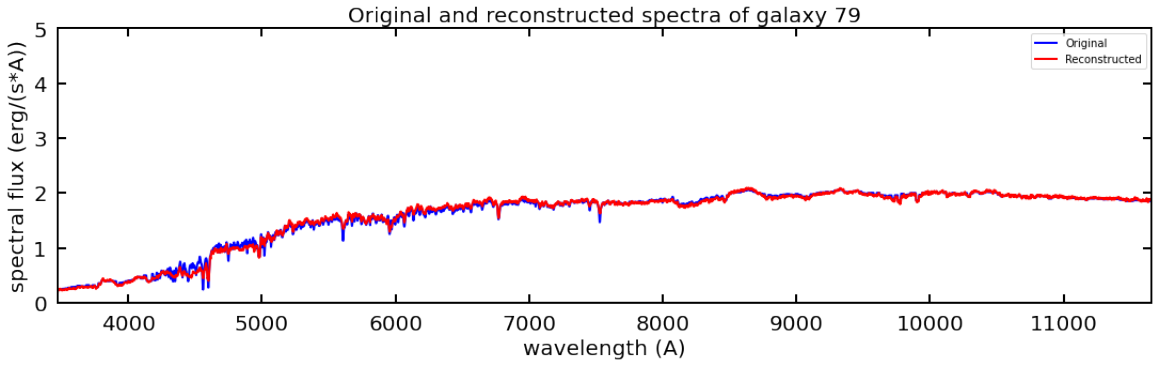


(c) Galaxy 34

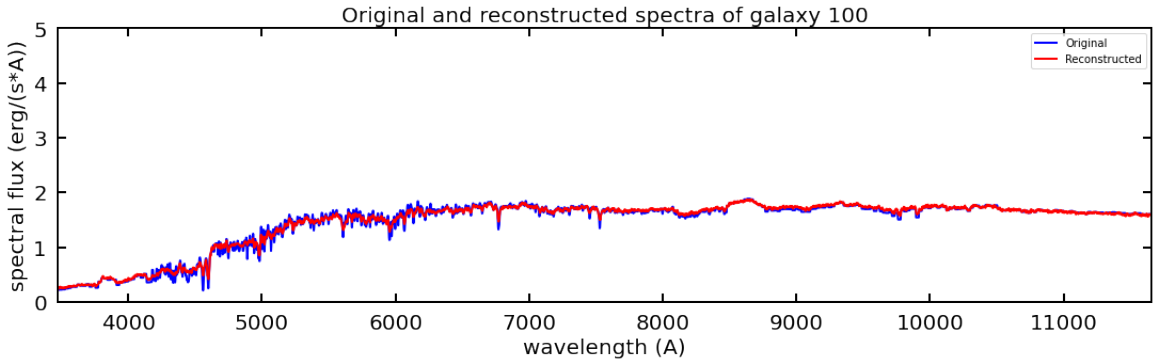
Figure 6: Wavelength against spectral flux data, where the original and reconstructed spectrum of galaxies 1, 26, and 34 are plotted in blue and red, respectively.



(a) Galaxy 68



(b) Galaxy 79



(c) Galaxy 100

Figure 7: Wavelength against spectral flux data, where the original and reconstructed spectrum of galaxies 68, 79, and 100 are plotted in blue and red, respectively.

We can now confirm that our autoencoder is working appropriately. We now have a representation of the spectra of 800 galaxies compressed in 256 parameters, which now will be used as the input for our further analysis.

3.2 Clustering implementation

Clustering is an important step in unsupervised learning, where we group data points that share similar properties or characteristics. In the case of clustering the spectra based on their corresponding latent space representation, it can help us identify underlying patterns in the data that are not immediately apparent in the original high-dimensional space in order to make a previous analysis before making the SOM. In the context of spectroscopy, clustering will help identify groups of spectra that have similar chemical or physical properties.

We analyze the clustering by making a trial-error practice. Because we are working with 800 galaxies, we want to cluster the spectra in a meaningful way, such that the elements in each group tell us physical information. We can decide when to stop increasing the number of clusters one we recognize that the elements in each cluster are similar and the elements do not overlap each other. There are several python packages and libraries that help us do the clustering, in this work, we will use the library *K-means*. Having this in mind, let's start the trial.

For the clustering, we will implement a python library by importing the KMeans class from the sklearn.cluster mode, which is commonly used for performing clustering algorithms. We start by stablishing the number of clusters for our database. Afterwards, a KMeans object is created with the specified number of clusters (n-clusters) and a random state of 0. We also use the fit() method by calling it on the object, which performs the k-means clustering algorithm on the latent-vectors data. The algorithm assigns cluster labels to each data point, and the resulting KMeans object (kmeans) stores these labels.

After setting this parameter, the k-means algorithm is applied to the latent vectors from the VAE using the *scikit-learn* implementation with the specified number of clusters. The cluster labels are assigned to each of the latent vectors. Finally, the number of spectra in each cluster is printed to the console. This allows the user to assess whether the clustering produced sensible results or if some clusters may be too small to be useful. We also create a loop that iterates over each cluster, from 0 to n-clusters - 1. For each cluster, it counts the number of spectra belonging to that cluster by using the $np.sum()$ function on the condition

$cluster-labels == i$, where i represents the current cluster number. This condition creates a boolean array, where elements are True if the corresponding spectrum belongs to the current cluster, and False otherwise. The $np.sum()$ function counts the number of True values, which gives the total number of spectra in the cluster. The result is then printed, displaying the cluster number and the number of spectra in that cluster.

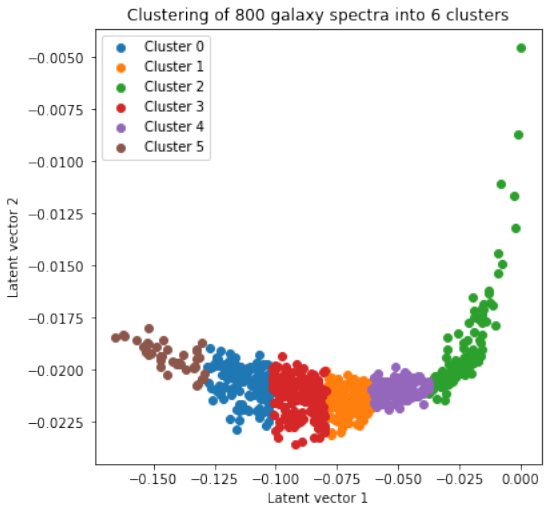
In order to decide the number of clusters we will use, we run our code by trial and error, analyzing the representation for each of the numbers chosen. After many trials, we can conclude that 6 clusters is a meaningful number for our database, because even though the number of galaxies in each cluster is not overall even when increasing the number of clusters, the code is still recognizing differences between them. The number of galaxies for each cluster is presented as follows:

- Cluster 0: 120 spectra
- Cluster 1: 199 spectra
- Cluster 2: 108 spectra
- Cluster 3: 192 spectra
- Cluster 4: 140 spectra
- Cluster 5: 41 spectra

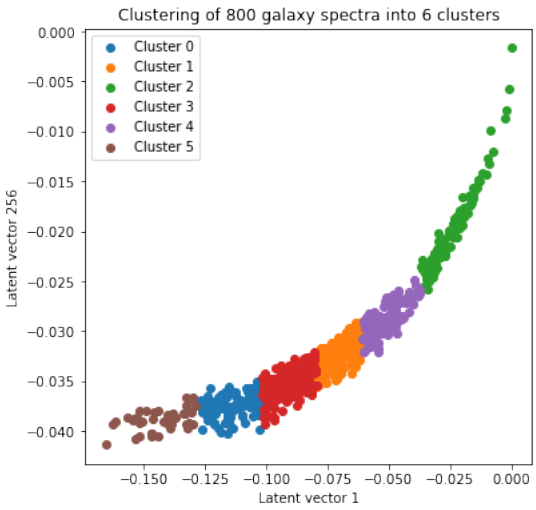
Now, we can code a clustering on the latet vectors and visualize these clusters in a scatter plot in order to provide insights into the grouping patterns of the galaxy spectra based on their latent representation. To do so, we first initialize a KMeans object with the specified number of clusters we established previously. We set a random state of 42 to ensure the reproducibility of the clustering results. Then, we call the $fit.predict()$ method on the KMeans object, taking the latent space representation of our VAE as the input. This will perform the k-means clustering on the latent space and assign a cluster label to each vector. Then, we create a figure for the scatter plot. Having done this, we create a loop that plots the clustered data, which iterates over each cluster and plots the latent vectors belonging to the current cluster i in the scattered plot. This plot is created using any two

values from the 1st to the 256th dimensions of the latent vectors. It is important to mention that these dimensions do not have a physical interpretation, they are rather the dimensions of the compressed representation of our original spectra for each of our initial galaxies. We can decide to plot the clustering in any random pair of dimensions of the latent space.

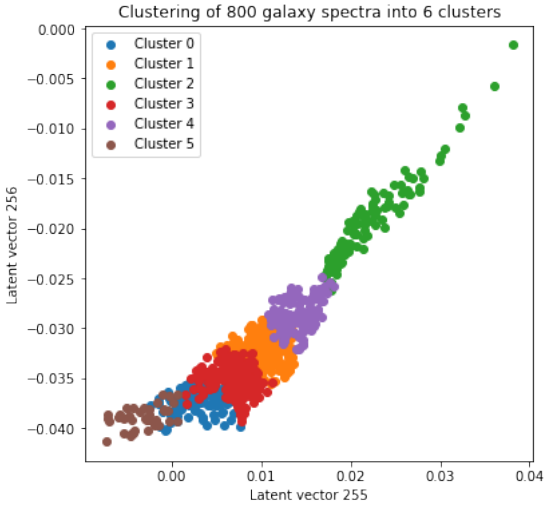
To visualize our clustering, we randomly chose four pairs of dimensions of our latent space, as seen in figure 8.



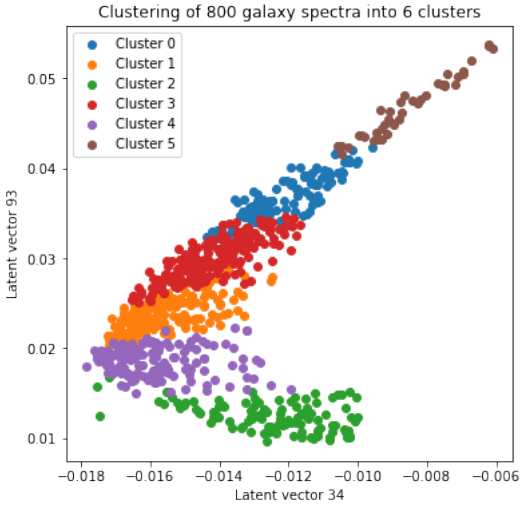
(a) Clustering with dimensions 1 vs. 2



(b) Clustering with dimensions 1 vs. 256



(c) Clustering with dimensions 255 vs. 256



(d) Clustering with dimensions 34 vs. 93

Figure 8: Scatter plot of the latent space after applying k-means clustering with 6 clusters, where each color corresponds to a different cluster.

It would be meaningful to obtain clusters that group the properties of each galaxy as follows acording to, for example, the following parameters:

- Metallicity and Gas Metallicity: Metallicity refers to the abundance of elements heavier than hydrogen and helium in a galaxy, while gas metallicity specifically refers to the metallicity of the interstellar gas. These parameters are crucial for understanding the chemical evolution and star formation history of galaxies.

- Redshift and Age: Redshift is a measure of the cosmological distance to a galaxy, which provides information about its distance and the epoch at which the light was emitted. Age, on the other hand, represents the age of the galaxy, indicating when it formed or experienced significant events. The correlation between redshift and age can reveal insights into the cosmic evolution of galaxies.

- SFR (Star Formation Rate) and $SFR - 100Myr$: SFR represents the rate at which a galaxy forms new stars, indicating its current level of star formation activity. $SFR - 100Myr$ represents the star formation rate averaged over the past 100 million years. These parameters can provide insights into the ongoing star formation processes and the star formation history of galaxies.

- E(B-V) and Dust Mass: $E(B-V)$ is a measure of the dust extinction, which quantifies the amount of dust obscuring the light from stars in a galaxy. Dust mass refers to the total amount of dust present in a galaxy. The correlation between these parameters can provide information about the role of dust in absorbing and scattering light and its impact on the observed properties of galaxies.

- AGN (Active Galactic Nucleus) Properties: Parameters like $AGN - tau$, fagn (AGN fraction), and $agn - tau$ (AGN obscuration timescale) relate to the presence and characteristics of an AGN in a galaxy. These parameters can provide insights into the influence of the supermassive black hole at the galactic center on the galaxy's properties and its impact on star formation and gas dynamics.

3.3 SOM implementation

Having done the pre-processing of the clustering, we are ready to build the SOM. To do so, we will implement the MiniSom library, which provides us with an implementation of the SOM algorithm. We start by defining the shape of our SOM grid. This parameter is established according to the number of elements that will be arranged in the map. Because we have 800 galaxies, it would be meaningful to choose a number of neurons, such that if the distribution were completely even, each neuron would contain approximately the same number of galaxies. We, therefore, choose a 10×10 grid.

We can now initialize the SOM specifying the imput data, which will be the latent space representation of the wavelength and spectra of our galaxies, i.e. the number of features that will be fed onto the map. We then set a random seed for reproducibility, which states that the random number generation will be the same each time the code is run, ensuring the reproducibility of results.

After having established these parameters, we can create our SOM by fixing certain features. The first one is *sigma*, which refers to the radius of influence of each neuron on the map; in other words, its neighborhood, which, as explained in section 2.3, controls how far the influence of a neuron extends during training. The smaller the sigma value, the more detailed representation we will obtain, while the larger the value, the more generalized and smooth representation will be. It is important to mention that this value affects the duration of the training process, because the more exact the solution, the longer it takes to find it. Therefore, it is crucial to balance between accuracy and time taken. Another important parameter is the *learning-rate*, which controls the rate at which the weights in the SOM grid are updated during training and determines the magnitude of weight adjustments.

We are now ready to initialize the weights of the SOM with random values drawn for a uniform distribution. Then, the training method is called to train the SOM on the input data. Here, we need to specify a sufficient number of epochs, which will train the SOM on the latent vectors for a certain number of iterations (epochs). Here we apply the train-random method, which updates the weights of the SOM using random samples from the dataset.

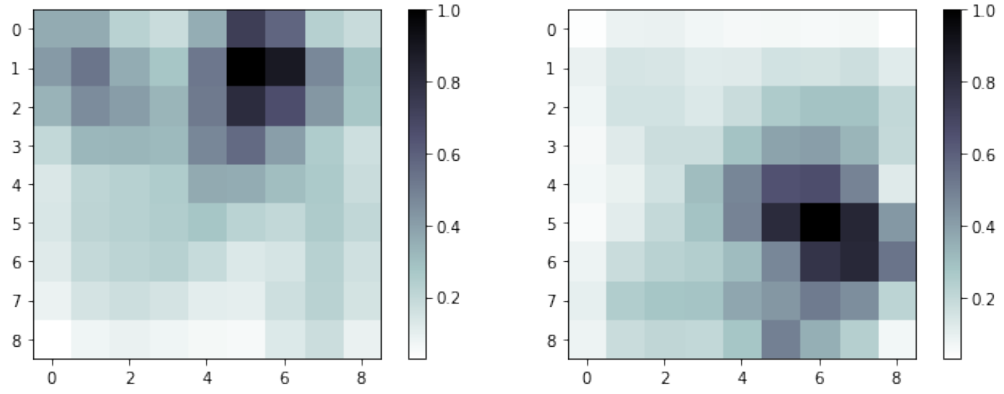
Now, we obtain the position of each input vector in the SOM grid by finding the winning neurons. This position maps each input vector to its corresponding position in the SOM grid. What this method does is return a dictionary where the keys are the coordinates of each neuron in the grid, and the values are lists of the input vectors mapped to that neuron.

Next, we define a function that takes in the SOM object the input data *latent vectors*, and the cluster labels *clusters*. The function first obtains the coordinates of the winning neuron for each input vector using the BMU method (recall section 2.3.1) of the SOM object. Then, we plot the input vectors as points in the SOM grid, with each point colored according to its cluster label.

Finally, we plot the SOM weights and the clustered input data. The resulting plot shows the SOM weights on the left side and the clustered input data on the right side, with each cluster labeled with a different color, as shown in figures 9.

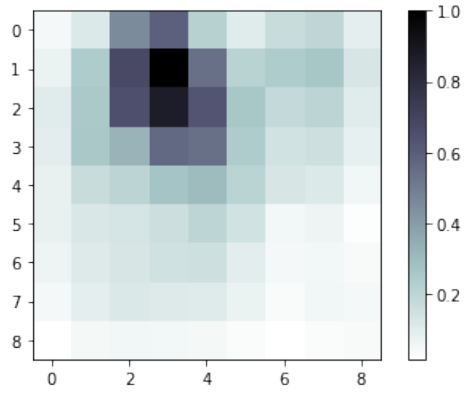
The colors of each neuron represent the different clusters or groups that have been formed based on the similarity of the latent vectors. Each color represents a different cluster or group, and spectra that have similar latent vectors are grouped together and represented by the same color. In this case, our input data is 1-dimensional spectra, which represent the intensity of electromagnetic radiation at different wavelengths. The intensity of radiation at different wavelengths can be used to extract the physical properties of the sources that emit the radiation, such as their temperature, chemical composition, and density.

The colors are used to visualize the clusters in the SOM, where each cluster is assigned a color. The clusters are formed based on the similarity of the latent vectors of the input data points. In this way, the SOM provides a visual representation of the distribution of the spectra in the latent space, making it easier to identify clusters and patterns that may not be apparent in the original data. The shades of gray on the right side of the SOM represent the frequency of winning of the neuron. Darker areas correspond to neurons that win more frequently, while lighter areas correspond to neurons that win less frequently.

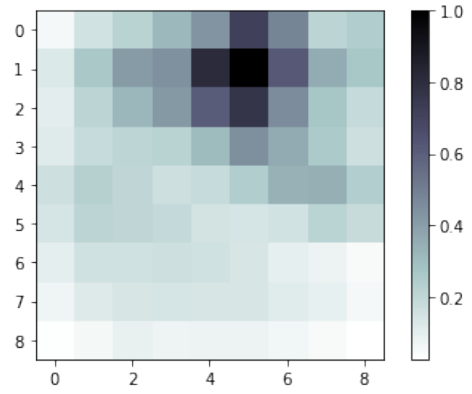


(a) Trial 1: SOM

(b) Trial 2: SOM



(c) Trial 3: SOM



(d) Trial 4: SOM

Figure 9: SOM that consists of a 2D grid of nodes, each representing a different region in the latent space. The color of each node corresponds to the average value of the latent vectors of the spectra that are mapped to that node.

The neurons of the SOM do not have a direct physical representation, they are rather computational units within the artificial neural network that serves as the underlying model. For our case of study, galaxies occupy certain neurons in a SOM signify that during the training process, each galaxy is mapped to the neuron in the SOM that best represents its characteristics or properties. The mapping is based on the similarity between the galaxy's features and the weight vectors associated with the neurons in the SOM.

However, the arrangement of neurons in the grid can provide useful insights into the organization and clustering of the galaxies based on their features. Neurons that are close

to each other in the SOM grid are considered to be similar in terms of their weight vectors and the characteristics they represent.

Thus, by examining the galaxies that are mapped to a particular neuron, we can infer that these galaxies share similar features or properties. This allows us to identify clusters or groups of galaxies with similar characteristics in the dataset. The SOM provides a way to visualize and analyze the high-dimensional data in a lower-dimensional grid, making it easier to interpret and understand the underlying patterns and relationships among the galaxies.

It is important to recall that the SOM algorithm is an unsupervised learning algorithm that initializes the weights of the neurons randomly. During the training, the weights of the neurons are adjusted based on the input data, which can lead to slightly different results each time the SOM is trained, as seen in the four trials of figure 9.

As the training progresses, the neurons organize themselves to represent the input data in a topological manner. However, the exact position of the neurons and the resulting colors can vary due to the random initialization of the weights and the stochastic nature of the training process. Therefore, running the SOM multiple times may result in slightly different color patterns in the neurons. However, the overall organization and topology of the SOM should remain consistent in order to observe the general clustering and relationships between the data points.

4 Conclusions

In conclusion, this project has focused on the analysis of galaxy spectra through the utilization of machine-learning techniques. By using the power of advanced algorithms and data-driven approaches, we have aimed to extract meaningful insights and patterns from the vast amount of spectral data available.

Firstly, we were able to process galaxy SED, specifically their spectra and wavelength by implementing a VAE in order to reduce the dimensions of these parameters. We obtained a successful result when plotting the original and reconstructed spectra. The implementation

of the VAE allowed us to capture the essential features and information within the SEDs, while significantly reducing the complexity of the data. This reduction in dimensionality facilitated easier visualization, analysis, and interpretation of the galaxy SEDs.

Moreover, when examining the clustering of the spectra, we utilized the reduced representation obtained from the VAE as our input. However, despite conducting a thorough analysis of the galaxies within each cluster, we encountered challenges in deriving meaningful physical interpretations. This was primarily due to the presence of similar parameters among different clusters and thus, was difficult to distinguish distinct characteristics. The limitations of our analysis can be attributed to factors such as the number of galaxies considered and the specific parameters employed during the training phase of our model.

In the case of the SOMs, we successfully constructed four of them by taking the latent space representation of the spectra from the VAE as the input. Since SOMs are an unsupervised machine learning technique, each map produced was unique. However, despite their distinctiveness, we observed similarities among the different SOMs, which aligns with our expectations. It is important to note that we encountered challenges when distinguishing between the distinct characteristics within each neuron of the SOMs, i.e. the physical properties of the galaxies that were cataloged in each cell.

Despite these challenges, the SOMs still provided valuable insights into the clustering and organization of the spectra, enabling us to identify general trends and similarities among different groups of galaxies, such as the grouping according to their redshift. They serve as a useful tool for exploratory analysis and provide a foundation for further investigations into the underlying properties and relationships of the spectra.

In future research, our goal is to expand our analysis to the entire catalog of galaxy SEDs, which includes the dataset of 10,000 galaxies. By applying the same process and techniques used in this study to a larger sample size, we aim to gain a more comprehensive understanding of the spectral properties and characteristics of galaxies. Ultimately, by conducting this process on the entire catalog of galaxy SEDs, we aim to advance our understanding of galaxy evolution, the physical processes driving the observed spectra, and the broader implications

for the formation and development of galaxies in the universe.

References

[1] Dor Bank, Noam Koenigstein, and Raja Giryes. Autoencoders. *arXiv preprint arXiv:2003.05991*, 2020.

[2] Dale A. Ostlie Bradley W. Carrol. *An introduction to modern astrophysics, 2nd Edition.* Pearson International Edition, 2007.

[3] Charlie Conroy, James E. Gunn, and Martin White. A Simple Model for the Observed Dispersion in the Cosmic Star Formation History. *Astrophysical Journal*, 699:486–506, June 2009.

[4] Van den Bosch F. White S. Mo, H. *Galaxy formation and evolution.* Cambridge University Press, 2010.

[5] Stephen KN Portillo, John K Parejko, Jorge R Vergara, and Andrew J Connolly. Dimensionality reduction of sdss spectra with variational autoencoders. *The Astronomical Journal*, 160(1):45, 2020.

[6] Hossen Teimoorinia, Finn Archinuk, Joanna Woo, Sara Shishehchi, and Asa FL Bluck. Mapping the diversity of galaxy spectra with deep unsupervised machine learning. *The Astronomical Journal*, 163(2):71, 2022.

[7] Giuseppe Vettigli. Minisom: minimalistic and numpy-based implementation of the self organizing map, 2018.

[8] Derek Wilson, Hooshang Nayyeri, Asantha Cooray, and Boris HÃ¤ußler. Photometric redshift estimation with galaxy morphology using self-organizing maps. *The Astrophysical Journal*, 888(2):83, jan 2020.