MNRAS 000, 1–16 (2019) Preprint 29 September 2019 Compiled using MNRAS LATEX style file v3.0

Vector Quantized Variational AutoEncoder (VQ-VAE) on Emulating Galaxy Images and Unsupervised Machine Learning Classification for Galaxy Morphology

Ting-Yun Cheng[1,2⋆] Marc Huertas-Company,[3,4] Brant Robertson,[5] Nesar Ramachandra,[6] Francois Lanusse,[7] Claire Guilloteau[1,8]

[1] *Kavli Student Fellow*
[2] *School of Physics and Astronomy, University of Nottingham, University Park, Nottingham, United Kingdom*
[3] *Instituto de Astrofísica de Canarias (IAC); Departamento de Astrofísica, Universidad de La Laguna (ULL), E-38200, La Laguna, Spain*
[4] *LERMA, Observatoire de Paris, CNRS, PSL, Université Paris Diderot, France*
[5] *University of California Santa Cruz, California, United States*
[6] *Argonne National Laboratory, Illinois, United States*
[7] *Berkeley Center for Cosmological Physics, Department of Physics, University of California, California, United States*
[8] *IRAP, Université de Toulouse, France*

Accepted XXX. Received YYY; in original form ZZZ

ABSTRACT
In this report, we show the works done in the Kavli Summer Program in Astrophysics 2019 at University of California Santa Cruz. We explore a newly machine learning technique entitled 'Vector Quantised Variational Autoencoder (VQ-VAE)' on two projects: (1) emulating galaxy images for the development of future surveys e.g. the Euclid Space Telescope, and (2) exploring the morphological classification of galaxies using unsupervised machine learning in this summer. First, the vector quantisation process in the VQ-VAE successfully helps to speed up the training process in both VQ-VAE and PixelCNNs, and to generate images with a better diversity than current generative adversarial networks (GANs). For emulation task, we develop a pipeline for creating a synthetic galaxy image which has been deconvolved with a specific point spread function (PSF) that can be simply adapted to other surveys. However, a further investigation and improvement in conditional training for the PixelCNNs is necessary. For the clustering task, we reveal the opposite effect between the capability of reconstruction and the distinction ability in clustering. Additionally, we show the potential of exploring galaxy morphological classification using unsupervised machine learning by showing the trend of the Hubble Type discovered by our methods. A further follow-up for both projects are planned in the future work.

Key words: keywords – keywords – keywords

1 INTRODUCTION

In this program, we explore a newly machine learning technique called 'Vector Quantized Variational AutoEncoder (VQ-VAE)' which developed by Google DeepMind (van den Oord et al. 2017; Razavi et al. 2019) on two different astronomical projects: (1) The emulation and generation of galaxy images, and (2) the morphological classification of galaxies using unsupervised machine learning (clustering).

First, the generation of realistic simulated data is an essential but challenging task for either the current observations or the execution of the future surveys. To generate artificial galaxy images, we can start with a simple parametric physical model such as a de Vaucouleurs profile (de Vaucouleurs 1948, 1953) for elliptical galaxies and an additional exponential component for spiral galaxies (e.g. Erben et al. 2001; Bertin 2009). Afterwards, a more generalised model such as Sérsic profile (Graham & Driver 2005) is applied to produce different light distributions for emulating galaxy images (e.g. Meert et al. 2013). To avoid the limitation caused by the parametric forms and extend the exploration to a deeper field in the previous methods, non-

⋆ E-mail:ting-yun.cheng@nottingham.ac.uk

parametric methods were then developed such as shapelets formalism (Massey et al. 2004; Dobke et al. 2010). In addition to emulating galaxy images through modelling the distribution of real galaxies, cosmological simulations such as the IllustrisTNG simulations (Nelson et al. 2019) provide a great resolution and high-fidelity data through simulating the interaction and evolution of galaxies. However, the cosmological simulations generally require an expensive computation ability, and have a limited description towards deeper fields.

The development of machine learning techniques such as generative models (e.g. Generative Adversarial Network, Autoencoder, Variational Autoencoder) enable to model the complex distribution of data from the real observed data with a lower cost in computation than cosmological simulations (Ravanbakhsh et al. 2016; Rodríguez et al. 2018; Caldeira et al. 2019; Mustafa et al. 2019) which also might provide a more precise emulation result because the trained model is based on the real data but being more complex than the model obtained from the traditional parametric and non-parametric methods (e.g. shapelets formalism).

The generative model, VQ-VAE, we explore in this study has a much faster emulation process than other generative models due to the vector quantisation procedure in the architecture (see details in Section 2.1). In addition, Razavi et al. (2019) shows that this vector quantisation procedure also enable the VQ-VAE to generate new artificial images with better diversity when combining with other generative models such as PixelCNNs (van den Oord et al. 2016) than the BigGAN (Brock et al. 2018) which is currently one of the most powerful generative models for generating stochastic fake images.

Due to the diversity of the generated images VQ-VAE can provide, we explore the capability of the VQ-VAE on emulating galaxy images and generating new artificial galaxy images. This emulation is useful in developing astronomical analysis tools for observations and the usage of simulation for future surveys such as the Euclid Space Telescope (**Euclid**). For example, the deep field of the simulation for **Euclid** is based on the data from the Cosmic Assembly Near-infrared Deep Extragalactic Legacy Survey (CANDELS) (Grogin et al. 2011) which has a limited amount of resource for the simulation of **Euclid**. To generate a large number of high-fidelity galaxy images with a certain morphology can well benefit the execution of **Euclid** project.

Second, along with the data explosion by more and more survey projects in astronomy, e.g. The Sloan Digital Sky Survey (SDSS)[1], the Large Synoptic Survey Telescope (LSST)[2], the Dark Energy Survey (DES)[3], etc, which will image more than hundreds of millions of galaxies, the traditional manual classification analysis by experts is obviously impossible to deal with this enormous amount of data.

The series of the Galaxy Zoo projects (Lintott et al. 2008, 2011; Willett et al. 2013) are one of the most successful tool to solve the problem of large scale morphological analysis. It allows amateurs to do the classification by answering a series of questions based on galaxy images. However, classification analysis is complex and difficult such that background knowledge and experience are essential when doing it. In addition, while visual morphological classification with Galaxy Zoo is faster than for single individuals, it is also time-consuming. For example, the Galaxy Zoo Project spent around 3 years on obtaining the classifications of ∼300,000 galaxies, due to the need for so many individual classifications per object. DES and LSST, for instance, would take on the order of $>$ 100 years to classify with the Galaxy Zoo project. Therefore, an efficient automated classification method by computational science is essential for the future of this field.

The first application of machine learning on morphological classification can be traced to **?**. They applied a neural network with an input layer of 13 parameters, e.g. stellar properties, brightness profile, etc., which gave an output of five different types of galaxies. Since then, a slew of studies in astronomy have appeared utilising the technology of machine learning (e.g. Huertas-Company et al. 2009, 2011; Shamir 2009; Polsterer et al. 2012; Sreejith et al. 2018; Cheng et al. 2019b), neural networks (e.g. Maehoenen & Hakala 1995; Naim 1995; Lahav et al. 1996; Goderya & Lolling 2002; Ball et al. 2004; de la Calleja & Fuentes 2004; Banerji et al. 2010; Cheng et al. 2019b), and Convolutional Neural Networks (CNN) (e.g. Dieleman et al. 2015; Huertas-Company et al. 2015, 2018; Domínguez Sánchez et al. 2018; Cheng et al. 2019b) for the morphological classification of galaxies.

Along with the success of the supervised machine learning such as CNNs on a variety of aspects in the studies of galaxy morphology, the power of unsupervised machine learning has not been investigated in detail. Unlike supervised machine learning methods which require a large amount of labelled data, and data labelling can be expensive and misleading, unsupervised machine learning help to save efforts on data labelling and reduce the human bias while training a machine. Additionally, unsupervised machine learning (clustering) methods provide a first glimpse of classification for a large amount of data which is helpful and efficient to select the preliminary 'interesting data' in the future surveys.

Therefore, scientists have started to explore the application of unsupervised machine learning to, e.g. photometric redshifts (Geach 2012; Krone-Martins & Moitinho 2014; Carrasco Kind & Brunner 2014; Siudek et al. 2018), as well as classification using photometry or spectroscopy (Fustes et al. 2013).

The application of unsupervised machine learning (clustering) becomes more challenging when using high dimensional data such as images. Hocking et al. (2018) is one of the first studies of unsupervised machine learning applications using imaging data which applied the Growing Neural Gas algorithm (Fritzke 1994). Afterwards, Cheng et al. (2019a) applies a different approaching than Hocking et al. (2018) that using convolutional autoencoder (CAE) (Masci et al. 2011) to do feature extraction before connecting with unsupervised machine learning algorithms.

In this report, we follow the methods described in Cheng et al. (2019a) to explore the capability of VQ-VAE on recognising galaxies' visual morphology in an unsupervised manner (clustering), and investigate the effects towards the clustering results from the different types of autoencoders e.g. VQ-VAE and Convolutional Autoencoder (ConvAE).

[1] https://www.sdss.org
[2] https://www.lsst.org
[3] https://www.darkenergysurvey.org/

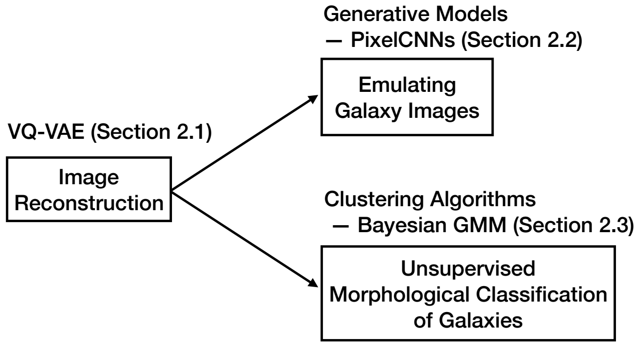


Figure 1. The illustration for the workflow of this work.

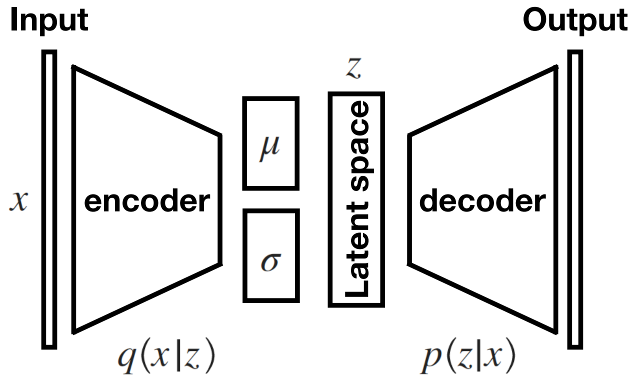


Figure 2. The illustration of the structure of Variational Autoencoders. The μ and σ represents the mean and the standard deviation, respectively.

The arrangement for this report is as follows. The VQ-VAE technique, the generative model for emulating, and the unsupervised machine learning technique adopted in this report are introduced in Section 2. Details about the data used in this study is described in Section 3. The implementation and the works done during the Kavli Summer Program in UC Santa Cruz are shown in Section 4. We list some discussions brought out from the preliminary results in this report and the future plans in Section 5. At last, the conclusion of the current progress is summarised in Section 6.

2 METHODOLOGY

In this work, we explore the application of the Vector Quantised-Variational AutoEncoder (VQ-VAE) on emulating galaxy images and the morphological classifications of galaxies using unsupervised machine learning algorithms. The workflow is described in Fig.1. In this section, we introduce the machine learning algorithms we use in this study that VQ-VAE for image reconstruction (Section 2.1), PixelCNNs for generating new galaxy images (Section 2.2), and Bayesian Gaussian mixture model (BayesianGMM) for unsupervised machine learning application (Section 2.3).

2.1 Vector Quantized Variational AutoEncoder (VQ-VAE)

The Vector Quantised-Variational AutoEncoder (VQ-VAE) is related to a variational autoencoder (VAE) and is applied to emulate the real galaxy images in our study. The task of image emulation is to, given a training data, learn the distribution of the given data, and reproduce the images with the distribution. The structure of a VAE (Fig. 2) contains an encoder with a posterior distribution $q\left(z|x\right)$ and a prior distribution $p\left(z\right)$ which is mapped using Gaussian distribution with an obtained mean and standard deviation, where x is the input data and z represents discrete latent variable, and a decoder with a distribution $p\left(x|z\right)$ for the input data.

The VQ-VAE is built based on the structure of VAE. Instead of using Gaussian distributions to map the latent space, the VQ-VAE applies an additional vector quantisation (VQ) procedure which make the posterior and prior distribution become categorical.

The posterior categorical distribution $q\left(z|x\right)$ is defined as below:

$$q\left(z=k|x\right)=\begin{cases}1 & for \quad k=argmin_j\|z_e\left(x\right)-e_j\|_2\\ 0 & otherwise\end{cases}, \quad (1)$$

where $z_e\left(x\right)$ is the output of the encoder, e_j represents a codebook which used for vector-quantising the $z_e\left(x\right)$, and k is the obtained index for the vector used in the selected codebook. We then measure the vector-quantised representation $z_q\left(x\right)$, which is the input of the decoder, through the Equation 1 and 2.

$$z_q\left(x\right)=e_k, \quad where \quad k=argmin_j\|z_e\left(x\right)-e_j\|_2. \quad (2)$$

Fig. 3 shows the VQ process (van den Oord et al. 2017). The output of the encoder, $z_e\left(x\right)$ can be represented by a combination of the index of different vectors, k, in the codebook. With this 'index map', we can rebuild a distribution, $z_q\left(x\right)$, with the same structure as $z_e\left(x\right)$ but each 'pixel' in $z_q\left(x\right)$ with the length of dimension, D, is quantised to one of the vector in the codebook for the input of decoder. The 'Embedding Space' shown in Fig. 3 represents the codebook, e_j.

The loss function of the VQ-VAE contains three parts: reconstructed loss, codebook loss, and commitment loss. The reconstructed loss is measured by comparing the results from the decoder with the input data. The codebook loss is used to make the selected codebook, e_j, approach to the output of the encoder, $z_e\left(x\right)$ while the commitment loss is applied to encourage the $z_e\left(x\right)$ to be close to the chosen codebook from the previous epoch.

Therefore, the loss function, L, for the VQ-VAE is described as below:

$$L=\log p\left(x|z_q\left(x\right)\right)+\|sg\left[z_e\left(x\right)\right]-e\|_2^2+\beta\|z_e\left(x\right)-sg\left[e\right]\|_2^2, \quad (3)$$

where the sq means the stopgradient operator and β is used for adjusting the weight for the commitment loss. van den Oord et al. (2017) found the results to be robust to the value of β. The results has no apparent change when β ranges from 0.1 to 2.0. We set $\beta=0.25$ in this study which follows the setting in van den Oord et al. (2017). The code of VQ-VAE

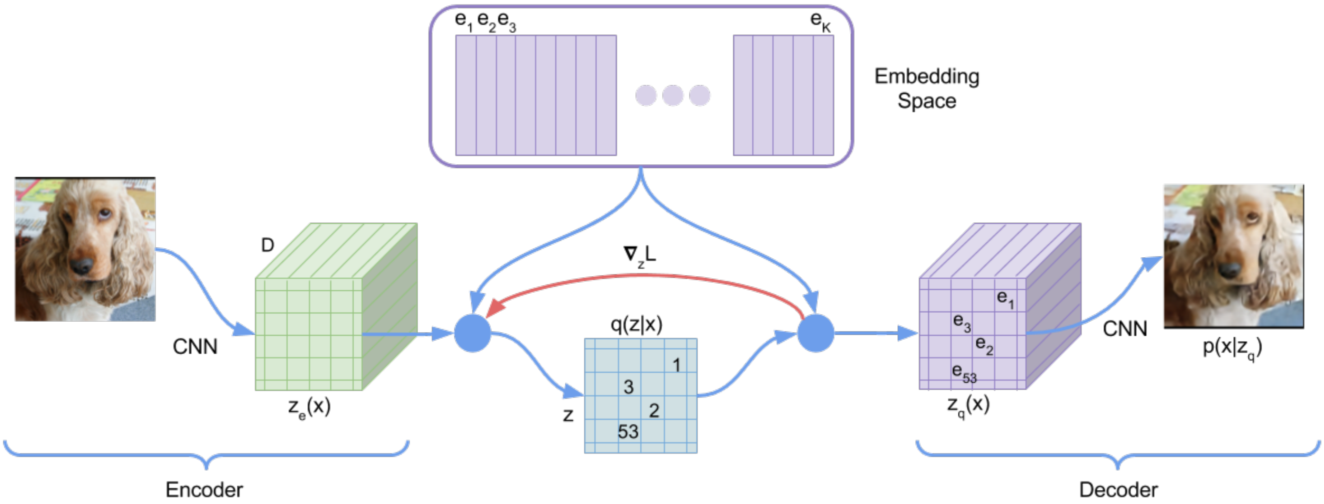


Figure 3. The illustration of the VQ-VAE. The source of this figure is from (van den Oord et al. 2017).

used in this study is built on **TensorFlow**[4] and **sonnet** library[5].

2.2 PixelCNNs

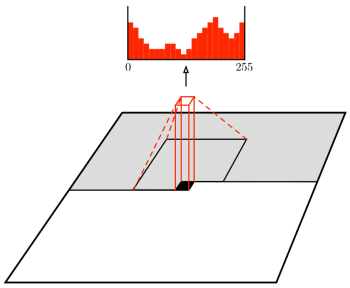
The prior distribution, $p(z)$, is a categorical distribution which is learnt through the process of reconstruction in VQ-VAE. This can be used to generate new images by providing different z in the feature map. In our study, we use PixelCNNs (van den Oord et al. 2016) to learn the prior distribution obtained from the VQ-VAE to generate random artificial galaxy images (Section 4.2).

The PixelCNNs is an autoregressive model which can model the joint distribution between pixels (Equation 4).

$$p(\mathbf{x}) = \prod_{i=1}^{n^2} p(x_i|x_1, \ldots, x_{i-1}), \quad (4)$$

where $\mathbf{x}$ represents input images and x_i indicates a single pixel in the $\mathbf{x}$. The learning process of PixelCNNs follows the conditional distributions $p(x_i|x_1, \ldots, x_{i-1})$ and every conditional distribution is modelled by a convolutional neural network. The ordering of the learning process is in raster scan order: row by row, pixel by pixel with every row which means that the mapping of each pixel only depends on the pixels above and the left of it. To fulfill this purpose, a masked filter is applied to images as shown in Fig. 4 (left). The prediction of PixelCNNs is also sequential that a pixel per time. Each pixel is predicted according to the previous pixels that has been predicted.

The generated images from PixelCNNs to this point is stochastic which means that the generation of each image is not related to each other in terms of physical properties. However, given practical labels for input images as a latent vector $\mathbf{h}$, we then can build a conditional PixelCNNs model



1	1	1	1	1
1	1	1	1	1
1	1	0	0	0
0	0	0	0	0
0	0	0	0	0

Figure 4. The illustration for the concept of PixelCNNs. The source of this figure is from (van den Oord et al. 2016). *Left:* The visualisation of how the PixelCNN maps the distribution of pixels. The model can only condition on the previously generated pixels, $x_1, \ldots, x_{i-1}$ to predict pixel x_i. *Right:* An example to show a filter applied to images to eliminate the contribution of the pixels below and at the right side of the pixels that has been predicted.

with the conditional distribution $p(x|\mathbf{h})$ that allows to generate artificial galaxy images with a certain label e.g. Hubble T-Types, stellar mass, luminosity, etc by adding the additional term, $\mathbf{h}$, into Equation 4:

$$p(x|\mathbf{h}) = \prod_{i=1}^{n^2} p(x_i|x_1, \ldots, x_{i-1}, \mathbf{h}). \quad (5)$$

where $\mathbf{h}$ as mentioned above is a specific class for labelling images which is coded as a one-hot encoding in the algorithm.

The PixelCNNs algorithm used in this study is built on **Tensorflow** and the main structure is from an online source [6].

4 https://www.tensorflow.org
5 https://sonnet.readthedocs.io/en/latest/

6 https://github.com/anantzoid/Conditional-PixelCNN-decoder

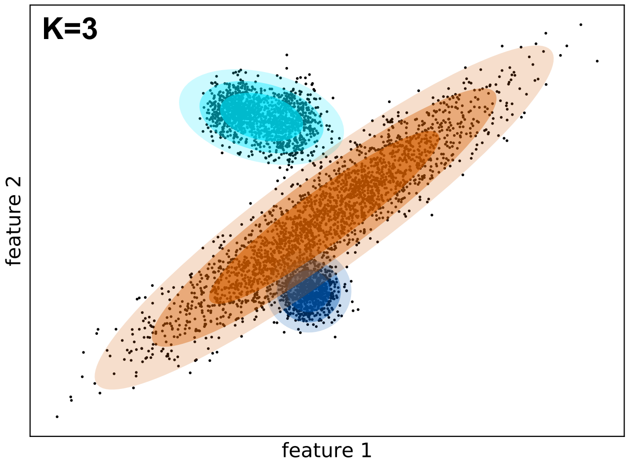


Figure 5. An illustration of the Gaussian Mixture model. The K is the number of Gaussian distributions. The black dots show the data distribution on the feature map, and the coloured ellipses represent three Gaussian distribution we applied here to fit the data distribution.

2.3 Bayesian Gaussian Mixture Model (BayesianGMM)

A Gaussian mixture model is a probabilistic model for either density estimation or clustering using a mixture of a finite number of Gaussian distributions to describe the distributions of data points on a feature map. For K clusters, the Gaussian mixture model is given as the form:

$$p(x) = \sum_{k=1}^{K} w_k G(x|u_k, \varepsilon_k), \tag{6}$$

where $G(x|u_k, \varepsilon_k)$ represents k-th Gaussian, u_k denotes the mean of the k-th Gaussian distribution, ε_k is the covariance matrix of the k-th Gaussian, and w_k is the prior probability (weight) of the k-th Gaussian. Where,

$$\sum_{k=1}^{K} w_k = 1. \tag{7}$$

An two dimensional illustration of the BGM is shown in Fig. 5 (Equation 6). The input data are distributed on the feature map (black dots). We use a number of Gaussian distribution, which is K=3 in this illustration (coloured ellipses), to fit the data distribution on each feature map. Each dot has a probability (weight) to each ellipse, the sum of probabilities for each dot is equal to 1 (Equation 7).

In unsupervised learning, expectation-maximization (EM) (Hartley 1958; Dempster et al. 1977; McLachlan & Krishnan 1997) is used to find the maximal log-likelihood estimates for the parameters of the Gaussian mixture model by an iterative process. The log-likelihood of the Gaussian mixture model is calculated using the formula:

$$\ln\left[p(x|u, \varepsilon, w)\right] = \sum_{n=1}^{N} \left\{ \ln\left[\sum_{k=1}^{K} w_k G(x|u_k, \varepsilon_k)\right]\right\}, \tag{8}$$

where N is the number of samples.

Bayesian Gaussian mixture model (BGM) is a variational Gaussian mixture model (Kullback & Leibler 1951; Attias 2000; Bishop 2006). In this study, we apply the BGM from the **scikit-learn** library [7] to do clustering.

3 DATA SETS

In this study, we use two different data sets: Cosmic Assembly Near-infrared Deep Extragalactic Legacy Survey (CANDELS) (Grogin et al. 2011) and Sloan Digital Sky Survey (SDSS) Data Release 7 (York et al. 2000; Abazajian et al. 2009).

The CANDELS combines several surveys using multiple Telescopes in both space such as the *Hubble Space Telescope* (HST) and ground-based Telescopes which includes the Great Observatories Origins Deep Survey (GOODS), GOODS-N and GOODS-S (Giavalisco et al. 2004), Extended Groth Strip (Davis et al. 2007), COSMOS (Scoville et al. 2007), and Ultra-deep Survey (Lawrence et al. 2007; Cirasuolo et al. 2007). In this study, we use the GOODS data from the CANDELS in h-band. The available data contains the CANDELS/Deep Survey, the CANDELS/Wide Survey, and the Early Release Science (ERS) data (Windhorst et al. 2011) which we have ∼6000 galaxy stamps in h-band in total.

As mentioned in Section 1, for the purpose of emulation, one of our goal is to build an emulator for the development of the Euclid Space Telescope **Euclid**. Therefore, the CANDELS data is an ideal choice for training our emulator. However, we have insufficient data available for training our machine in this summer program. Therefore, for the development of our methodology, we mainly applied the SDSS data in this preliminary work. In this study, we use the SDSS Data Release 7 (DR7) data which contains five-band photometry for 357 million distinct objects. However, we only use r-band imaging data and focus on galaxy class in this work. We apply a redshift cut z≲0.15 according to the catalogue of the morphological classification of galaxies for the SDSS DR7 from Meert et al. (2015) (M15 hereafter). Additionally, we also apply the Hubble T-Type classification from this M15 catalogue to do conditional training in the task of emulation (Section 4.2.2).

4 IMPLEMENTATION & RESULTS

4.1 Image reconstruction through the VQ-VAE

We first use the VQ-VAE as the feature extractor for both emulating and clustering problems. The architecture of our VQ-VAE, which follows the design in (Razavi et al. 2019), is shown in Fig. 6 and the details of each layer is listed in Table 1. We applied the VQ-VAE on both CANDELS and SDSS data (Section 3). The hyper-parameter settings for both data sets are listed in Table 2 and the reconstruction results are shown in Fig. 7 and Fig. 8. The reconstructed images using the VQ-VAE well reproduce the features of the original images and generally show a smoother background than the original images.

An observed image is convolved with a specific point

[7] https://scikit-learn.org/stable/index.html

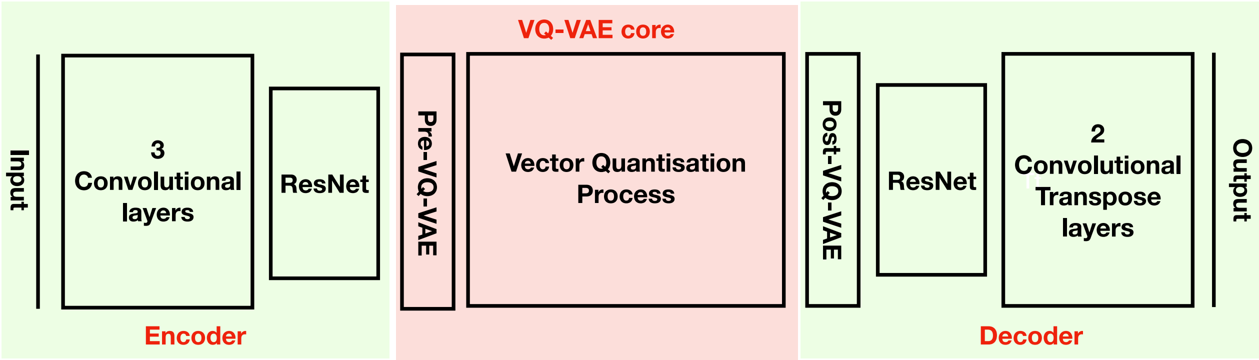


Figure 6. An illustration for the architecture of VQ-VAE used in this study. The details in each layer are shown in Table 1

Type	#features	filter size	stride size	non-linearity
		ResNets		
Conv2D_res1	32	3×3	1×1	ReLu
Conv2D_res2	128	1×1	1×1	ReLu
		Encoder		
Conv2D_1	64	4×4	2×2	ReLu
Conv2D_2	128	4×4	2×2	ReLu
Conv2D_3	128	3×3	1×1	ReLu
ResNets				
		Pre-VQ-VAE		
Conv2D_4	64	1×1	1×1	
		Decoder		
Conv2D_5	128	3×3	1×1	ReLu
ResNets				
Conv2DTranspose_1	64	4×4	2×2	ReLu
Conv2DTranspose_2	1	4×4	2×2	

Table 1. The hyper-parameters for the architecture of the VQ-VAE used in this study.

Hyper-parameters	CANDELS	SDSS
Input size	84×84	64×64
Feature map size after encoder	21×21	16×16
β (see Equation 3)	0.25	0.25
Batch size	32	32
Residual layers	2	2
Codebook size	512	512
Codebook dimension	64	64
Training step	100000	100000

Table 2. The hyper-parameters of VQ-VAE encoder and decoder used for CANDELS and SDSS data.

spread function (PSF) which is mainly due to the diffraction of light through the telescope and also can be caused by the seeing during the observation. Considering an image characterised by its intensity distribution, I, corresponding to the observation of a 'real' image, O through an optical system,

$$\begin{aligned} I(x,y) &= \iint_{-\infty}^{\infty} P\left(x-x^{'}, y-y^{'}\right) O\left(x^{'}, y^{'}\right) dx^{'} dy^{'} \\ &= (P * O)(x,y), \end{aligned} \tag{9}$$

where P is the PSF.

To enable our emulator to adapt to other surveys, a convolution process with a PSF corresponding to the input data using the Convolution Theorem (Equation 9) is added into the structure of the VQ-VAE after the last convolutional transpose layer (*Conv2DTranspose_2* in Table 1) to generate a final output with PSF (Fig. 6).

$$I(x,y) = (P * O)(x,y) = F^{-1}\left[F(P)\,F(O)\right], \tag{10}$$

where F represents the operation of Fourier Transform and F^{-1} is the Inverse Fourier Transform.

This extra step of adding a PSF into the output images forces the machine deconvolve the output of the last convolutional transpose layer (*Conv2DTranspose_2* in Table 1) during the training process.

In this study, we simply apply the PSF downloaded from

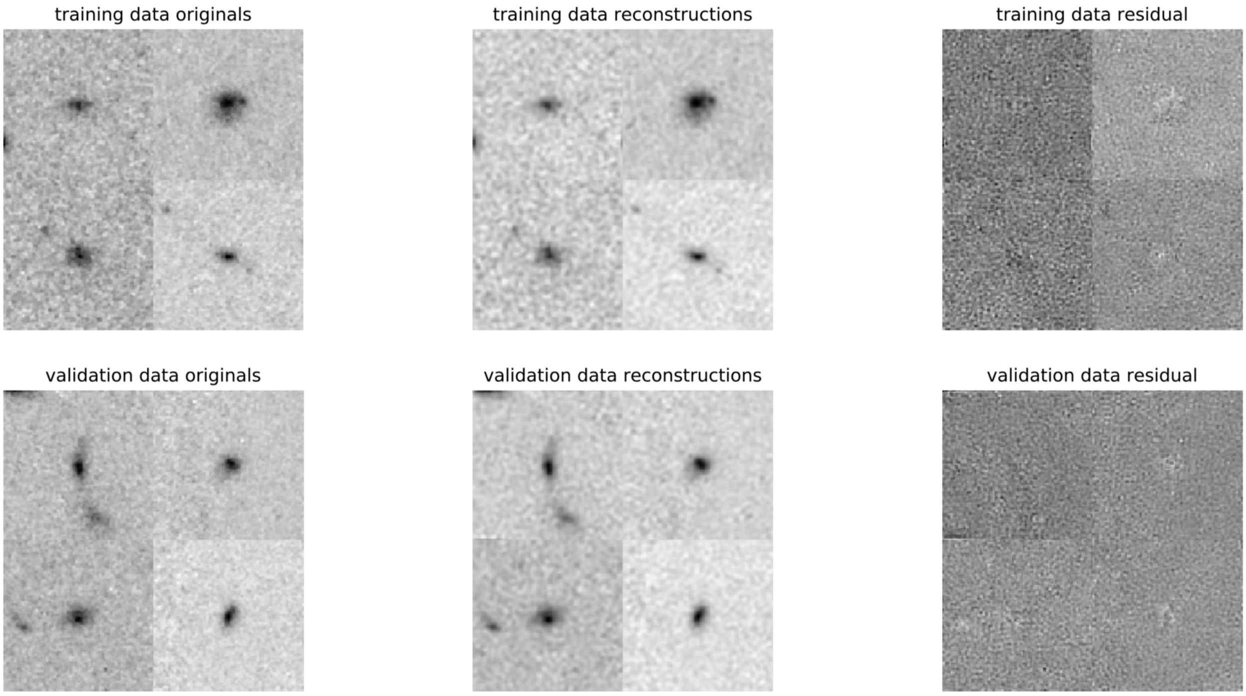


Figure 7. The reconstruction of CANDELS data using VQ-VAE. The first column is the original images while the second column shows the reconstructed images. The final column shows the residual.

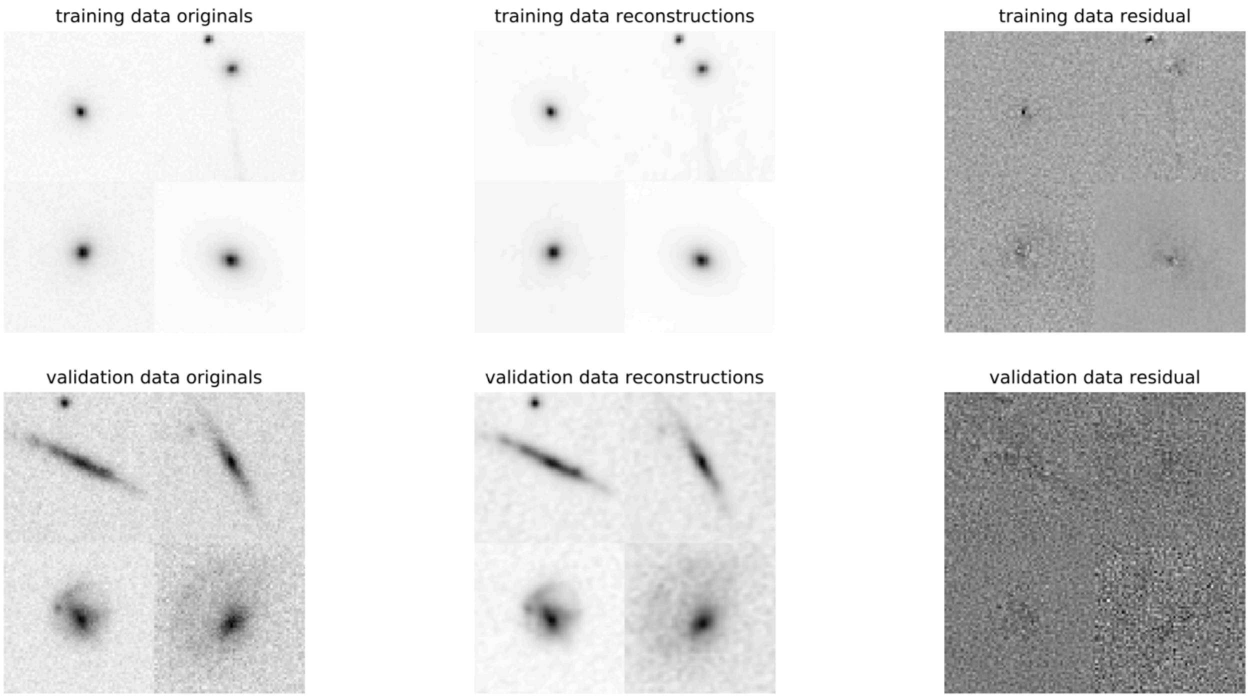


Figure 8. The reconstruction of SDSS data using VQ-VAE. The first column is the original images while the second column shows the reconstructed images. The final column shows the residual.

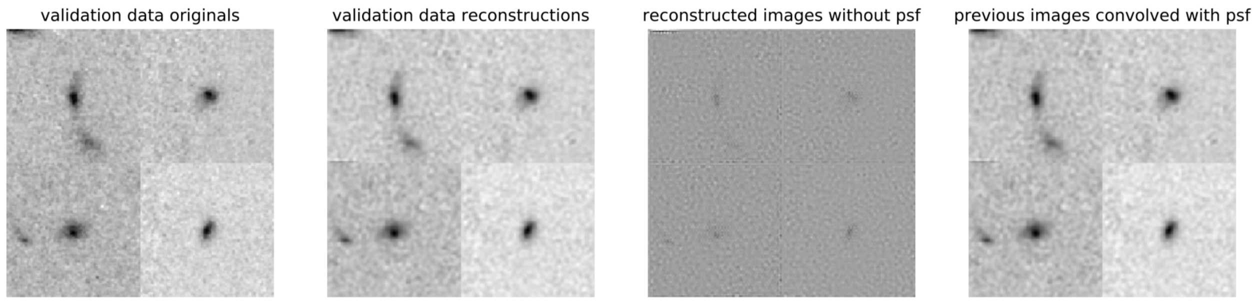


Figure 9. The example of the PSF deconvolution using the VQ-VAE. The first column is the original images and the second column shows the reconstructed images while the third column presents the deconvolved images and the fourth one is the validation that using the same PSF to convolve the previous images.

3D-HST[8] for CANDELS data (GOODS-S, WFC3, F160W) to all CANDELS data. The result is shown in Fig. 9. This preliminary result confirms the feasibility of this deconvolution process in the VQ-VAE; however, to obtain a precise deconvolved images a corresponding PSF for each input image is required in the future work.

4.2 Emulating Galaxy Images using PixelCNNs

There are two main output from the VQ-VAE: (1) each image is presented as a two dimensional feature map and each pixel in the feature map is an index to indicate the representation code in the codebook; (2) a codebook which includes 512 codes with a dimension of 64 in our study. With these two information, we can simply reconstruct a image from a feature map using a pre-trained VQ-VAE decoder.

To generate a reliably synthetic galaxy image from the real observed data, machine need to learn the prior distribution of the input data so that given a stochastic latent variable, machine can generate a random realistic galaxy image based on the prior distribution. In this program, we apply a powerful generative model called 'PixelCNNs' (Section 2.2 to learn the prior distribution from the vector quantised feature maps extracted from the input data (the output after 'VQ-VAE core' in Fig. 6). Given a latent variable as an outset, it then construct a new feature map pixel by pixel based on the distribution of previous predicted pixels. Inputting this new feature map to a pre-trained VQ-VAE decoder, we then obtain a new synthetic galaxy image.

The application of PixelCNNs on the feature maps obtained before fed into the VQ-VAE decoder instead of the original input images speed up the training process of PixelCNNs in general (Razavi et al. 2019) because of the size difference between the input images and the feature maps (see Table 2) and the limited options of representation codes in the codebook (512 codes are available in this study) for each pixel when constructing a feature map.

4.2.1 Unconditional Training using CANDELS data

We firstly trained our PixelCNNs unconditionally (Equation 4) using the CANDELS data and a PSF downloaded from the 3D-HST. The setting of the hyper-parameters for the PixelCNNs is listed in Table 3. The hyper-parameters: 'Decay_steps' and 'Decay_rate' control the decay of the learning rate during training. The decay learning rate is updated through the formula:

Hyper-parameters	CANDELS	SDSS
Conditional training	No	T-Type
Input size	21×21	16×16
Batch size	32	32
Learning rate	0.0003	0.0003
Decay_steps	50000	50000
Decay_rate	0.5	0.5
Grad_clip	5.0	5.0
#Features	441	256
#Layers	18	18
Training step	100000	100000

Table 3. The hyper-parameters of the PixedlCNNs used for CANDELS and SDSS data.

$$lr_{decay} = lr \times R_{decay}^{\left(\frac{S_{global}}{S_{decay}}\right)}, \tag{11}$$

where lr_{decay}, R_{decay}, and S_{decay} represents the decay learning rate, 'Decay_rate', and 'Decay_steps', respectively.

The preliminary generation of synthetic galaxy images trained by CANDELS data is shown in Fig. 10. To obtain the deconvolved galaxy images (the right one in Fig. 10), we retrieve the output images of the last convolutional transpose layer (*Conv2DTranspose_2* in Fig. 6) before convolved with a PSF.

4.2.2 Conditional Training using SDSS data

To enable us to have more controls in generating galaxy images with a specific property, we then train our PixelCNNs to condition on Hubble T-Type (Kartaltepe et al. 2015). However, we have insufficient number of CANDELS data (∼6500 images) to do conditional training for a range of T-Type classification (T). Therefore, we decide to test our method on SDSS data to obtain a preliminary result for methodology development. The setting of PixelCNNs for this test is listed in Table 3.

We use the T-Type classification from M15 which contains ∼670,000 objects from the SDSS Data Release 7 in the

[8] https://3dhst.research.yale.edu/Data.php

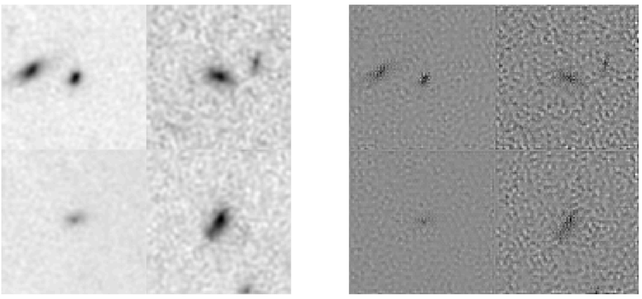


Figure 10. The example of the synthetic galaxy images generated by PixelCNNs using CANDELS data. *Left:* the generated images. *Right:* the deconvolved generated images.

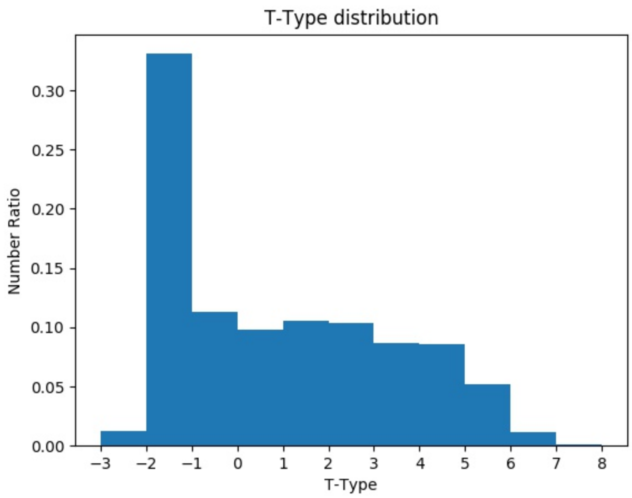


Figure 11. The distribution of the T-Type classification of galaxies in the M15 catalogue.

catalogue to condition our PixelCNNs. The distribution of the T-Type classification in the M15 catalogue is shown in Fig. 11 which we also apply a selection of redshift $z < 0.15$. The distribution clearly shows that we have relatively more galaxies with $-2 \leq T < -1$ and apparently less numbers of galaxies with $-3 \leq T < -2$, $6 \leq T < 7$, and $7 \leq T < 8$ in this catalogue. We categorise the floating value of T-Type of each object to the closest integer and define 10 classes of T-Type from $T = -3$ to $T = 6$. We then train conditionally on the data with categorical class of T-Type.

To start with a fair training condition, we randomly pick 12,000 galaxies from each class. For the classes having insufficient data (i.e. $T = -3$ and $T = 6$), we randomly rotate the images in these two categories to increase the available number of data, then randomly pick 12,000 galaxies from each new pool.

The preliminary result is shown in Fig. 12. Visually, it is difficult to observe a clear trend of galaxy morphology in the generated galaxy. However, we roughly can observe that the generated galaxies with smaller value of T-Type (more elliptical) look less spiral than the generated galaxies with larger value of T-Type (more spiral).

4.2.3 Validation Tools

One way to validate the quality of our generated images is to examine their physical properties to compare with those properties of real observed data. There are a lot of software is available to this task such as **PyMorph**[9] (Vikram et al. 2010), **Morfometryka** (Ferrari et al. 2015), and **statmorph**[10] (Rodriguez-Gomez et al. 2019). To examine the generated images, we apply a **Python** package entitled **statmorph**[11] (Rodriguez-Gomez et al. 2019) which relatively simply access to measure non-parametric morphological factors of galaxies e.g. C-A-S statistics (Conselice 2003).

However, although we roughly observe a 'trend' of galaxy morphology shift in our preliminary results of conditional PixelCNNs, the non-parametric 'galaxy' properties (e.g. concentration, asymmetry, smoothness) of the generated images have no significant change along with the galaxy morphology (T-Type) which might be caused due to some issues in the training process of the PixelCNNs. To solve this issue, a further investigation for the conditional training is of great importance. Additionally, other generative models such as Masked Autoregressive Flows (MAFs) (Papamakarios et al. 2017) is considered in the future work. The detailed discussion is shown in Section 5.

4.3 Unsupervised Machine Learning Exploration on Galaxy Morphology

In this part, we only work on the SDSS data because of the larger available number of SDSS data than CANDELS data we had. We follow the same procedure as Section 4.2 to build up our training data for the unsupervised machine learning application so that each class (T-Types from $T = -3$ to $T = 6$) contains 12,000 galaxies. Differently, we apply a scaling process using *arcsinh* function to images that reducing the contract of brightness between components if there are multiple objects in one image.

In this section, in addition to VQ-VAE, we also briefly compare the performance of the VQ-VAE (Section 4.3.1) with a Convolutional Autoencoder (ConvAE) in Section 4.3.2.

4.3.1 VQ-VAE Clustering

For the application of unsupervised machine learning, we follow Cheng et al. (2019a) to use VQ-VAE as a feature extractor to capture the representative features then connect them with an unsupervised machine learning algorithm, 'Bayesian Gaussian mixture model (BayesianGMM)' (Section 2.3).

One of the main hyper-parameter in BayesianGMM is the number of classification cluster used to separate the galaxies. However, there is not yet a simple and a proper way to select the optimal number of cluster. Cheng et al. (2019a) applies a factor called 'Area under the Receiver operating characteristic curve (AUC)' to optimise the number of cluster used in their study; however, this factor is measured by comparing with other confirmed classification re-

[9] https://github.com/vvinuv/pymorph
[10] https://statmorph.readthedocs.io/en/latest/
[11] https://statmorph.readthedocs.io/en/latest/

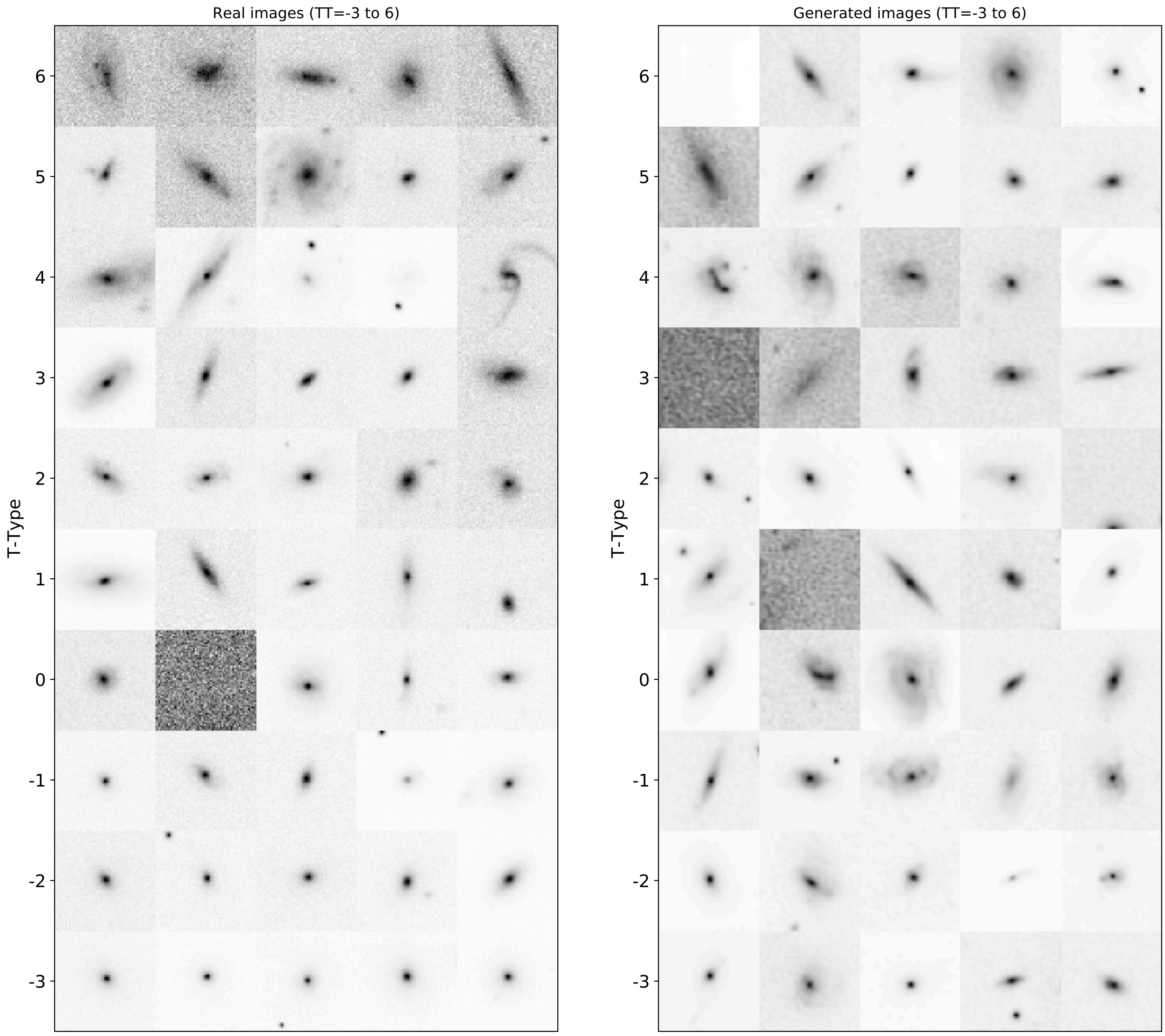


Figure 12. The synthetic galaxy images generated by the combination of the VQ-VAE and PixelCNNs. Each row from top to bottom shows different type of galaxy from $T = 6$ to $T = -3$. Left: real galaxy from SDSS. Right: generated galaxy images.

sults. Therefore, it is not an ideal selection method for this study.

Guo et al. (2017) suggests that the number of cluster shall be the same as the number of extracted features in the autoencoders. Therefore, we decide to initially use the number of features to the number of cluster in this part of the study. For this purpose, we slightly change the architecture of the VQ-VAE described in Section 2.1 by adding an extra convolutional layer in both encoder and decoder with a kernel size of 4 and a stride size of 2 to make the number of feature reduce to 64 (8×8) rather than 256 (16×16). This modification is mainly for the convenience of analysing less number of clusters in the preliminary clustering results.

The number of extracted features is decided through a test using several available number: 4, 16, 64, 256 in the VQ-VAE. We visually check the reconstructed images and measure the reconstruction loss (mean squared errors) to compare the capability of reconstruction using different value. Eventually we decide to use 64 as the number of features as well as the number of cluster in this investigation of the unsupervised machine learning.

The T-Type distribution of each cluster is shown in Fig. 13 (codebook size=512 in the VQ-VAE). Around 12 clusters show a clear inclination towards early-type galaxies; however, only about 4-5 clusters show an insignificant trend of late-type galaxies. Most clusters do not show a trend following the galaxy morphology; however, it may follows different galaxy properties which will be an interesting future work to continue investigating.

Additionally, we also test the impact using different size of the codebook: 16, 32, 64, 128, 256, 512. We observe that the size of codebook affects the performance of reconstruction that smaller size of codebook has slightly worse reconstruction ability in recovering intensity or background noises. However, as shown in Fig. 14, using smaller size of codebook such as 16 used in the figure, different classifi-

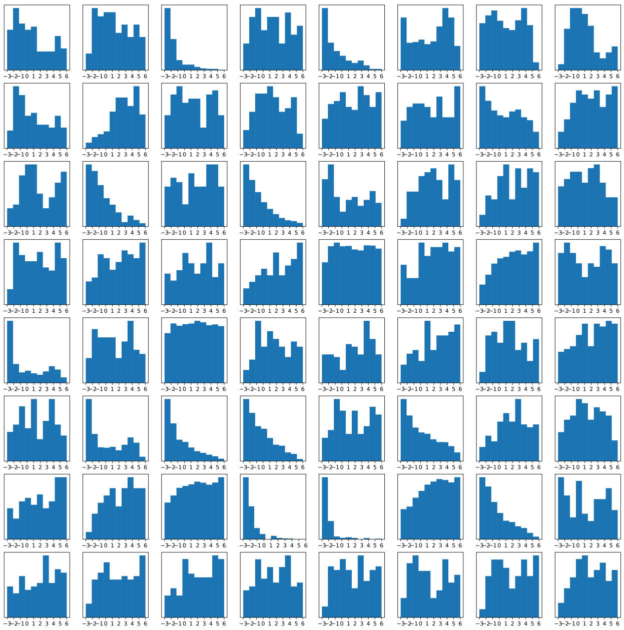


Figure 13. The T-Type distribution of clusters obtained from the combination of the VQ-VAE (size of codebook=512) and BayesianGMM. The x-axis from left to right represents galaxies are more elliptical, lenticular, then more spiral.

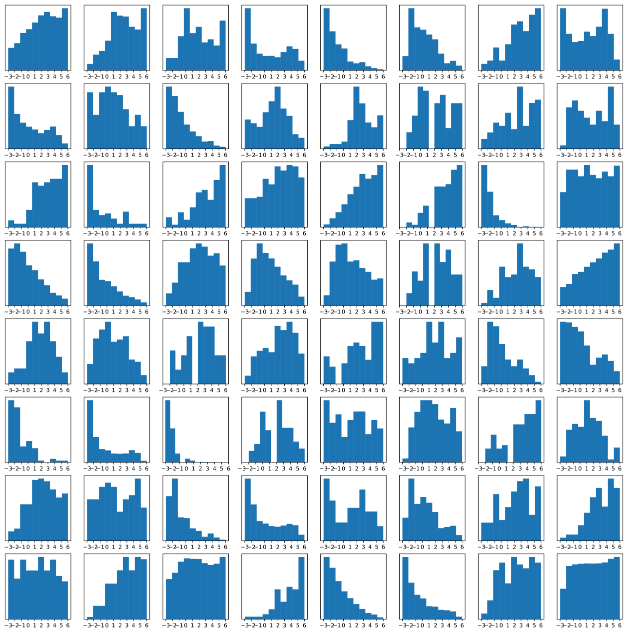


Figure 14. The T-Type distribution of clusters obtained from the combination of the VQ-VAE (size of codebook=16) and BayesianGMM. The x-axis from left to right represents galaxies are more elliptical, lenticular, then more spiral.

cation clusters show more clear tendency towards different morphology of galaxies that ∼18 clusters and ∼16 clusters show the features of early-type galaxies and late-type galaxies, respectively. In addition, interestingly, we also observe a more clear trend for Hubble Type transition in different clusters using a codebook size of 16 (Fig. 14) than the VQ-VAE using the codebook with a size of 512 (Fig. 13). We pick some relatively more representative clusters to show the observed Hubble Type trend of galaxy morphology in Fig. 15.

4.3.2 Comparing with Convolutional Autoencoder Clustering

The astronomical application of the unsupervised machine learning technique combined with the architecture of convolutional neural networks starts with (Cheng et al. 2019a). They apply a Convolutional Autoencoder (ConvAE) to extract features from images before fed into a clustering algorithms on simulated strong gravitational lensing images for Euclid Space Telescope (Metcalf et al. 2019) and they obtain a great success in distinguishing several different types of lensing such as different sizes of Einstein rings and arcs structures. Therefore, we follow Cheng et al. (2019a) to apply a ConvAE to make a comparison in this study.

We test two different architecture of ConvAEs: (1) without and (2) with dense layers. Both architectures are designed to extract 64 features before fed into the clustering algorithm, BayesianGMM (Section 2.3). The architecture (1) is to mimic the architecture of the VQ-VAE without the vector quantisation process. In the vector quantisation process, the dimensionality of features maps shrink from the size of $8{\times}8{\times}64$ into 64 features ($8{\times}8$); therefore, we add 4 extra convolutional layers with filter sizes of 64, 16, 4, 1 to instead of ResNet in Fig. 6 in the encoder to gradually reduce the number of features to 64.

On the other hand, we modify the architecture (2) by adding 1 extra convolutional layer with a filter size of 64 and 4 dense layers with sizes of 4096, 1024, 256, 64 in the encoder to gradually extract 64 features through dense layers.

First we compare the reconstruction ability between these four architectures in Fig. 16. The comparison clearly shows that the ConvAE with dense layers has the worst reconstruction performance amongst all. The VQ-VAE with codebook size of 16 then shows slightly worse reconstruction result in mean squared errors of the residuals than another VQ-VAE and the ConvAE without dense layers. The other two architectures, VQ-VAE with codebook size of 512 and the ConvAE without dense layers have similar reconstruction ability to each other which are shown in the measure of the mean squared errors as well as the visual check.

We then also plot the T-Type distribution of the clustering results from other two ConvAE models in Fig. 17 and Fig. 18. Both ConvAE models (with and without dense layers) show a better distinction in galaxy morphology than the VQ-VAE with a codebook of a size of 512. It is difficult to decide which model is the better one amongst the three models: VQ-VAE (codebook size=16) and two ConvAEs. However, the ConvAEs cannot preserve rotation invariant (Cheng et al. 2019a) which means that the ConvAEs classify the same galaxy into different categories when we rotate the image. The convolutional layers extract localised features; the ConvAEs compresses input images into several categorical localised features, it then reproduces images based on

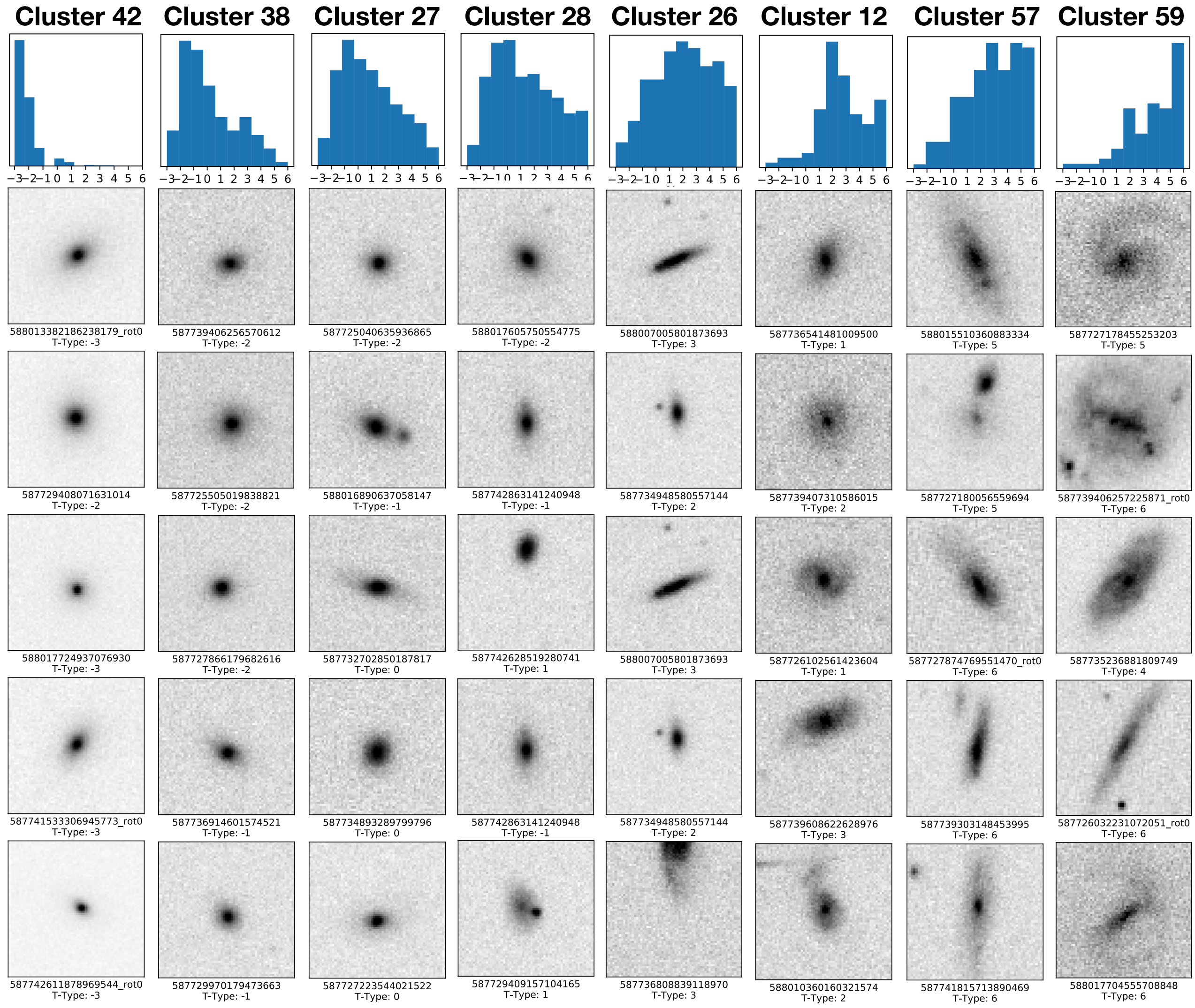


Figure 15. The presentation of some representative clusters which show a trend of Hubble Type using the codebook size of 16 in the VQ-VAE. The text above shows the cluster ID and the first row shows the T-Type distribution of the cluster.

these features. Therefore, the location of each pixel in the image is of great importance in the ConvAEs that the rotation of images causes different clustering results. On the contrast, the series of variational autoencoder e.g. VQ-VAE assign a distribution to map the extracted features. Through this mapping process, machine connects similar features with different location by describing through distributions that might help to preserve the rotation invariant.

We are at the starting point of applying unsupervised machine learning techniques on astronomical imaging data. There are only a few studies working on this application using imaging data (Hocking et al. 2018; Martin et al. 2019; Cheng et al. 2019a); therefore, a quantitative validation method such as Receiver operating characteristic curve (ROC curve) for supervised machine learning application is not developed for the application of unsupervised machine learning yet. To define a quantitative examination and build a standard and systematic validation method are of great importance in the future application of unsupervised machine learning.

5 DISCUSSION & FUTURE WORKS

This work can be separated into three main parts to do discussion and each of them has its own interesting approaching can be improved and explored in the future. Here we list several issues we have spotted in the current progress and present a potential approaching can be applied and investigated in the future for each part.

5.1 Image Reconstruction

First, along with the fast development of machine learning field, there are many options of machine learning techniques for image reconstruction e.g. variational autoencoder (VAE) (Kingma & Welling 2013) and generative adversarial network (GAN) (Goodfellow et al. 2014). It is very challenging to clearly discuss the pros and cons for all the available methods. Therefore, to develop a standard validation tool will be very helpful in this topic, and this is also one of the most important part for other two sections.

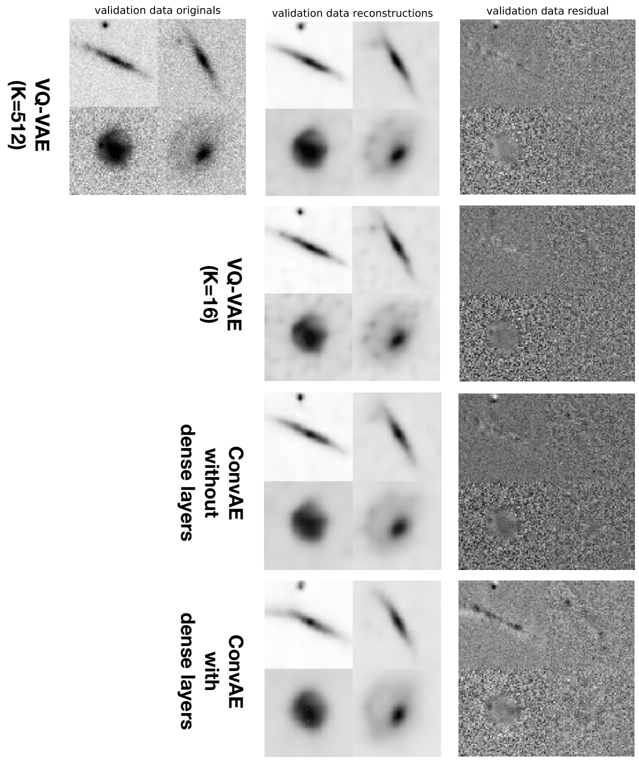


Figure 16. The comparison between four different autoencoders: VQ-VAE (the size of codebook, K=512), VQ-VAE (the size of codebook, K=16), ConvAE without dense layers, and ConvAE with dense layers. The first column shows the original images while the second and the third columns present the reconstructed images and the residuals.

Second, focus on the technique we used in this study, VQ-VAE, some choices of the hyper-parameters such as (1) the size and dimensionality of the codebook, (2) the number of training epochs, and (2) the design of the architecture need to be investigated more.

The size of codebook in the VQ-VAE controls the available options of code on rebuilding the feature maps which means that it decides how detailed the machine can reproduce the images. In this sense, we expect that the smaller size of codebook, the worse reconstruction results we have, and this seems the case in our test.

We test different size of codebook in the VQ-VAE: 16, 32, 64, 128, 256, 512. Although the difference of results using different size of codebook is insignificant, we observe that the reconstruction results are relatively noiseless and have lower brightness when the less numbers of code are applied (Fig. 16). This phenomenon is due to the limited number of codes can be used to recover the features map. When machine has less number of codes available, it will force itself to apply this code on more significant features instead of background noises and slight brightness change at the outskirt.

This effect also influences the choices of codebook size for emulation and clustering. For emulation, it depends on researches, in the case that to generate synthetic denoise images, smaller size of codebook is more appropriate, but to generate images with details, the larger size of codebook is

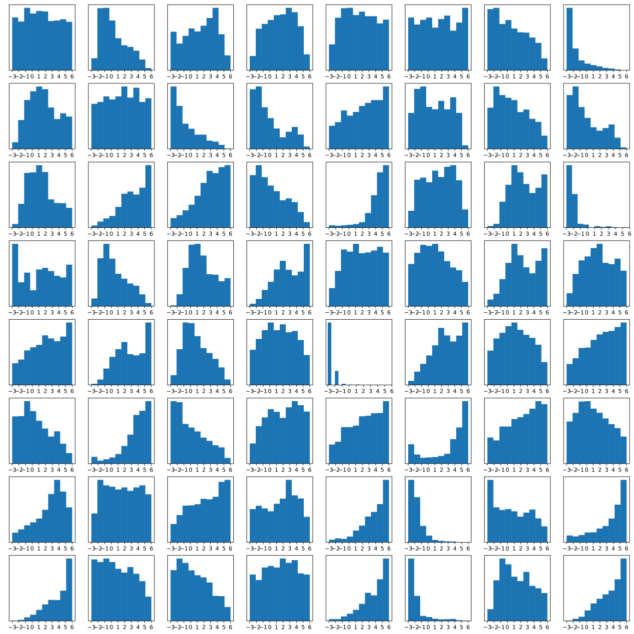


Figure 17. The T-Type distribution of clusters obtained from the combination of the ConvAE without dense layers and BayesianGMM. The x-axis from left to right represents galaxies are more elliptical, lenticular, then more spiral.

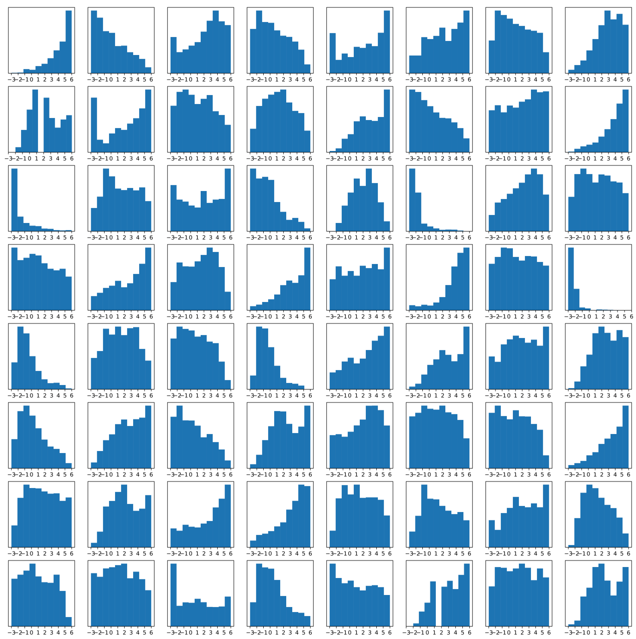


Figure 18. The T-Type distribution of clusters obtained from the combination of the ConvAE with dense layers and BayesianGMM. The x-axis from left to right represents galaxies are more elliptical, lenticular, then more spiral.

applied. On the other hand, for clustering on galaxy morphological classification, a smaller size of codebook is suggested because the noise level and insignificant change in images are not the main concern in classification of galaxy morphology.

The dimensionality of the codebook is supposed to influence the capability of reconstruction as well but might have much less impact than the size of codebook. However, we have no time to investigate this hyper-parameter in this report which will be an interesting part to explore in the future if we continue applying the VQ-VAE.

The number of training step is a ambiguous hyper-parameter to be determined because the reconstruction loss converges much earlier before a good reconstruction result is obtained in the autoencoder. This situation might be caused by more modifications are constructed and deconstructed to build up the details but remain a certain reconstruction loss after converging. Therefore, the number of training step in this study is literally determined by trial and error which needs to find a more quantitative or qualititative way to decide.

At last, for the purpose of emulation, a modification for the architecture of the VQ-VAE might be useful to reconstruct images with better resolution rather than the smoother images we obtain in this study. Razavi et al. (2019) applies a two-levels sequential training in the VQ-VAE that provides a learning process to hierarchically learn latent codes. This extra process can impressively improve the resolution of reconstruction images that might be helpful for the emualtion task.

5.2 Emulating Galaxy Images

Firstly, we have not successfully generate conditional synthetic galaxy images in this task yet. To approach this goal in the future development of our methodology, a further investigation for the choices of hyper-parameters, the number of training data, the number of training epochs is necessary. On the other hand, a lot of alternative generative models can be applied in this study such as Masked Autoregressive Flows (MAFs) (Papamakarios et al. 2017).

In this task, we are also interested in a potential improvement from adding colour information. We have a clear scheme of the relationship between the galaxy morphology and the galaxy properties such as colour in general that early-type galaxies are mostly massive, with older stellar populations, and redder while late-type galaxies, which include spiral galaxies and irregular galaxies, and consist of a younger population show blue features in the disk structures and red features in the bulge. This scheme in real galaxy might be very helpful to constrain machine to generate artificial galaxy images with a precise brightness distribution.

In this project, one of our main goal in the future work is to build an emulator for the development of the Euclid Space Telescope. To approach this, we suppose to train our machine with data of deeper field from the Hubble Space Telescope such as CANDELS or COSMOS data (Leauthaud et al. 2007). However, to train our pipeline, we need a large amount of data for the conditional training. The approaching of data augmentation or alternative options will be essential to continue this work in the future.

At last, another future work we are interested in is to condition on galaxy size or other properties instead of galaxy morphology (e.g. Hubble T-Type) which might be relatively simple to examine and obtain a robust conclusion for developing our pipeline. Additionally, to condition on floating values instead of categorical classes is an useful extension in our future work as well.

5.3 Unsupervised Machine Learning on Galaxy Morphological Classification

First, we compare the VQ-VAE with another autoencoder called 'Convolutional autoencoder (ConvAE)' in this study. However, the comparison in this study is done by a visual check to the images in the classification clusters and the T-Type distribution in each cluster. To show galaxy properties such as T-Type distribution in each cluster is a straight and clear way to examine the performance of clustering which will be interesting to see the comparison of other galaxy properties in the future works. However, this might not be sufficiently robust and convincing in this aspect. The application of unsupervised machine learning using imaging data is fairly new in astronomy (Hocking et al. 2018; Martin et al. 2019; Cheng et al. 2019a). A standard and systematic analysis procedure and validation tools to determine the performance of a clustering algorithm such as Receiver operating characteristic curve (ROC curve) for supervised machine learning application have not been developed in both computer science and astronomy yet. To develop a validation standard for comparing different unsupervised machine learning techniques will be very useful for the future research.

Second, as we have discussed above in Section 5.1. The codebook size in the VQ-VAE can significantly influence the clustering results. This is a combat between reconstruction ability and classification distinction ability. To use a less number of codes in the VQ-VAE, we obtain a relatively worse reconstruction result, but have a significantly improvement in the ability of distinguishing different galaxy morphology.

Using the codebook size of 16 in the VQ-VAE, we observe a trend of Hubble Type in the clustering results (Section 4.3.1 and Fig. 15). However, one of the study we intend to explore through clustering techniques is to investigate other possibilities of the galaxy morphological classification which might differ from the traditional Hubble Type though machine's view. This can also be applied to many different investigations such as galaxy evolution and other astronomical topics.

Third, in this study, we do not have a proper and quantitative selection for the number of clusters we used in this study. This is also another crucial question for applying unsupervised machine learning techniques. There are a few clusters algorithms approaching a optimal number of cluster themselves e.g. hierarchical clustering such as Agglomerative Hierarchical Clustering (Bouguettaya et al. 2015) and density-based clustering such as DBSCAN (Ester et al. 1996), can be considered in the future work to instead of Gaussian mixture models used in this study. However, to build a optimisation process for selecting the number of clusters when using any unsupervised machine learning techniques will be very useful for the future application in the field. Some potential approaching might can be done using Principal Component Analysis (PCA), t-SNE (van der

Maaten & Hinton 2008) which a further investigation is surely needed in the future.

Fourth, a relative minor future work is to add the colour information in the application which as mentioned in Section 5.2 is very reasonable improvement can be applied in the future work for the galaxy morphological classification.

At last, one of an interesting goal for developing this clustering methodology is to apply this method on the simulated data of the James Webb Space Telescope which might give us a prediction of the morphology distribution at a higher redshift.

6 CONCLUSION

In this report, we present the work done in the Kavli summer program at the University of California Santa Cruz. We explore the astronomical application of a newly technique entitled 'Vectore Quantised Variational Autoencoder (VQ-VAE)' on emulating galaxy images for the development of future surveys and the investigation of galaxy morphological classification using unsupervised machine learning.

First, we develop a pipeline of the emulator using the VQ-VAE and PixelCNNs which given a specific PSF image can also deconvolve images during training process of the VQ-VAE. We can do both unconditional and conditional generation using our pipeline. However, the preliminary results of conditional training show that some issues still need to be investigated in our generated images which might be caused by the conditional training process in the PixelCNNs. If we can overcome the problem of the final steps to generate high-fidelity synthetic galaxy, this emulator can generate galaxy images without PSF convolution that can be simply adapted to other surveys. One of the useful application is to train our emulator by either CANDELS or COSMOS data to generate synthetic galaxy images for the development of the Euclid Space Telescope.

Second, we explore the unsupervised machine learning application using the VQ-VAE and Convolutional Autoencoder (ConvAE). We observe that reducing the size of codebook in the VQ-VAE decrease the capability of reconstruction but increase the distinction ability in clustering. Additionally, we discover a trend of Hubble Type in the clusters classified by the VQ-VAE (codebook size=16).

On the other hand, we test two different architecture of ConvAEs (with and without dense layers) but with the same number of features and output clusters as the VQ-VAE. Both ConvAEs show similar performance to each other, and show a better distinction than the VQ-VAE with larger size of codebook (codebook size=512). However, these preliminary results are lacking a quantitative description. Through this investigation, we realise a better and more robust validation tools needs to be define in the field of unsupervised machine learning application.

The application of unsupervised machine learning is an relatively unexplored territory in astronomy; therefore, we think of a lot of interesting approaching in the future which can be continue working on. However, the first step of all is to develop the methodology. Therefore, we decide to continue working on SDSS data to develop the methodology because there are many information and analysis done for SDSS data which is helpful for examining our method. Once we build up our methodology, a further exploration to the data from the Dark Energy Survey, which has a better resolution and deeper depth than SDSS, will be applied. Additionally, we are also planning to apply our method on the simulated data (e.g. from IllustrisTNG simulations) for the James Webb Space Telescope to predict the galaxy morphology distribution at the higher redshift in the future work.

ACKNOWLEDGEMENTS

This work was initiated as a project for the Kavli Summer Program in Astrophysics held at the University of California Santa Cruz (UCSC) in 2019. We thank the organisers and UCSC for their great effort and generous support. Also a thank you to all the students, mentors, and staff, for giving us such a great memory, working experience, and motivation in both life and work.

REFERENCES

Abazajian K. N., et al., 2009, ApJS, 182, 543
Attias H., 2000, in In Advances in Neural Information Processing Systems 12. MIT Press, pp 209–215
Ball N. M., Loveday J., Fukugita M., Nakamura O., Okamura S., Brinkmann J., Brunner R. J., 2004, MNRAS, 348, 1038
Banerji M., et al., 2010, MNRAS, 406, 342
Bertin E., 2009, Mem. Soc. Astron. Italiana, 80, 422
Bishop C. M., 2006, Pattern Recognition and Machine Learning (Information Science and Statistics). Springer-Verlag, Berlin, Heidelberg
Bouguettaya A., Yu Q., Liu X., Zhou X., Song A., 2015, Expert Syst. Appl., 42, 2785
Brock A., Donahue J., Simonyan K., 2018, CoRR, abs/1809.11096
Caldeira J., Wu W. L. K., Nord B., Avestruz C., Trivedi S., Story K. T., 2019, Astronomy and Computing, 28, 100307
Carrasco Kind M., Brunner R. J., 2014, MNRAS, 438, 3409
Cheng T.-Y., Li N., Conselice C. J., Aragón-Salamanca A., 2019a, "*Identifying Strong Lenses with Unsupervised Machine Learning using Convolutional Autoencoder*", in Prep.
Cheng T.-Y., et al., 2019b, arXiv e-prints, p. arXiv:1908.03610
Cirasuolo M., et al., 2007, MNRAS, 380, 585
Conselice C. J., 2003, ApJS, 147, 1
Davis M., et al., 2007, ApJ, 660, L1
Dempster A. P., Laird N. M., Rubin D. B., 1977, JOURNAL OF THE ROYAL STATISTICAL SOCIETY, SERIES B, 39, 1
Dieleman S., Willett K. W., Dambre J., 2015, MNRAS, 450, 1441
Dobke B. M., Johnston D. E., Massey R., High F. W., Ferry M., Rhodes J., Vand erveld R. A., 2010, PASP, 122, 947
Domínguez Sánchez H., Huertas-Company M., Bernardi M., Tuccillo D., Fischer J. L., 2018, MNRAS, 476, 3661
Erben T., Van Waerbeke L., Bertin E., Mellier Y., Schneider P., 2001, A&A, 366, 717
Ester M., Kriegel H.-P., Sander J., Xu X., 1996, in Proceedings of the Second International Conference on Knowledge Discovery and Data Mining. KDD'96. AAAI Press, pp 226–231, `http://dl.acm.org/citation.cfm?id=3001460.3001507`
Ferrari F., de Carvalho R. R., Trevisan M., 2015, ApJ, 814, 55
Fritzke B., 1994, in Proceedings of the 7th International Conference on Neural Information Processing Systems. NIPS'94. MIT Press, Cambridge, MA, USA, pp 625–632, `http://dl.acm.org/citation.cfm?id=2998687.2998765`
Fustes D., Manteiga M., Dafonte C., Arcay B., Ulla A., Smith K., Borrachero R., Sordo R., 2013, arXiv e-prints, p. arXiv:1309.2418

Geach J. E., 2012, MNRAS, 419, 2633
Giavalisco M., et al., 2004, ApJ, 600, L93
Goderya S. N., Lolling S. M., 2002, Astrophysics and Space Science, 279, 377
Goodfellow I. J., Pouget-Abadie J., Mirza M., Xu B., Warde-Farley D., Ozair S., Courville A., Bengio Y., 2014, arXiv e-prints, p. arXiv:1406.2661
Graham A. W., Driver S. P., 2005, Publ. Astron. Soc. Australia, 22, 118
Grogin N. A., et al., 2011, ApJS, 197, 35
Guo X., Liu X., Zhu E., Yin J., 2017, in ICONIP.
Hartley H., 1958, Biometrics, 14(2), 174
Hocking A., Geach J. E., Sun Y., Davey N., 2018, MNRAS, 473, 1108
Huertas-Company M., et al., 2009, A&A, 497, 743
Huertas-Company M., Aguerri J. A. L., Bernardi M., Mei S., Sánchez Almeida J., 2011, A&A, 525, A157
Huertas-Company M., et al., 2015, ApJS, 221, 8
Huertas-Company M., et al., 2018, ApJ, 858, 114
Kartaltepe J. S., et al., 2015, ApJS, 221, 11
Kingma D. P., Welling M., 2013, arXiv e-prints, p. arXiv:1312.6114
Krone-Martins A., Moitinho A., 2014, A&A, 561, A57
Kullback S., Leibler R. A., 1951, Ann. Math. Statist., 22, 79
Lahav O., Naim A., Sodré L. J., Storrie-Lombardi M. C., 1996, MNRAS, 283, 207
Lawrence A., et al., 2007, MNRAS, 379, 1599
Leauthaud A., et al., 2007, ApJS, 172, 219
Lintott C. J., et al., 2008, MNRAS, 389, 1179
Lintott C., et al., 2011, MNRAS, 410, 166
Maehoenen P. H., Hakala P. J., 1995, ApJ, 452, L77
Martin G., Kaviraj S., Hocking A., Read S. C., Geach J. E., 2019, arXiv e-prints, p. arXiv:1909.10537
Masci J., Meier U., Cireşan D., Schmidhuber J., 2011, in Proceedings of the 21th International Conference on Artificial Neural Networks - Volume Part I. ICANN'11. Springer-Verlag, Berlin, Heidelberg, pp 52–59, `http://dl.acm.org/citation.cfm?id=2029556.2029563`
Massey R., Refregier A., Conselice C. J., David J., Bacon J., 2004, MNRAS, 348, 214
McLachlan G., Krishnan T., 1997, The EM algorithm and extensions. Wiley, New York
Meert A., Vikram V., Bernardi M., 2013, MNRAS, 433, 1344
Meert A., Vikram V., Bernardi M., 2015, MNRAS, 446, 3943
Metcalf R. B., et al., 2019, A&A, 625, A119
Mustafa M., Bard D., Bhimji W., Lukić Z., Al-Rfou R., Kratochvil J. M., 2019, Computational Astrophysics and Cosmology, 6, 1
Naim A., 1995, in American Astronomical Society Meeting Abstracts. p. 1416
Nelson D., et al., 2019, Computational Astrophysics and Cosmology, 6, 2
Papamakarios G., Pavlakou T., Murray I., 2017, arXiv e-prints, p. arXiv:1705.07057
Polsterer K. L., Gieseke F., Kramer O., 2012, in Ballester P., Egret D., Lorente N. P. F., eds, Astronomical Society of the Pacific Conference Series Vol. 461, Astronomical Data Analysis Software and Systems XXI. p. 561
Ravanbakhsh S., Lanusse F., Mandelbaum R., Schneider J., Poczos B., 2016, arXiv e-prints, p. arXiv:1609.05796
Razavi A., van den Oord A., Vinyals O., 2019, CoRR, abs/1906.00446
Rodriguez-Gomez V., et al., 2019, MNRAS, 483, 4140
Rodríguez A. C., Kacprzak T., Lucchi A., Amara A., Sgier R., Fluri J., Hofmann T., Réfrégier A., 2018, Computational Astrophysics and Cosmology, 5, 4
Scoville N., et al., 2007, ApJS, 172, 1
Shamir L., 2009, Monthly Notices of the Royal Astronomical Society, 399, 1367
Siudek M., et al., 2018, arXiv e-prints, p. arXiv:1805.09905
Sreejith S., et al., 2018, MNRAS, 474, 5232
Vikram V., Wadadekar Y., Kembhavi A. K., Vijayagovindan G. V., 2010, MNRAS, 409, 1379
Willett K. W., et al., 2013, MNRAS, 435, 2835
Windhorst R. A., et al., 2011, ApJS, 193, 27
York D. G., et al., 2000, AJ, 120, 1579
de Vaucouleurs G., 1948, Annales d'Astrophysique, 11, 247
de Vaucouleurs G., 1953, MNRAS, 113, 134
de la Calleja J., Fuentes O., 2004, MNRAS, 349, 87
van den Oord A., Kalchbrenner N., Vinyals O., Espeholt L., Graves A., Kavukcuoglu K., 2016, CoRR, abs/1606.05328
van den Oord A., Vinyals O., kavukcuoglu k., 2017, in Guyon I., Luxburg U. V., Bengio S., Wallach H., Fergus R., Vishwanathan S., Garnett R., eds, , Advances in Neural Information Processing Systems 30. Curran Associates, Inc., pp 6306–6315, `http://papers.nips.cc/paper/7210-neural-discrete-representation-learning.pdf`
van der Maaten L., Hinton G., 2008