Dimensionality Reduction of Astronomical Spectroscopic Data using Autoencoders

Quanbin Ma
Department of Machine Learning
Carnegie Mellon University
quanbinm@andrew.cmu.edu

Abstract

Background. Recent major advances in the understanding of galaxy owe a great deal to highly successful galaxy surveys conducted with the Sloan Telescope. With massive amount of data, modern techniques such as machine learning comes in naturally for efficient handling. While many applications focused on characterizing and classifying astronomical objects within a predefined area of interest, unsupervised learning has great potential in processing newly obtained data without much human intervention.

Aim. The goal of this project is to harvest the recent breakthrough of deep learning in both methodologies and computing resources, to build an automated system to generate embeddings of a manageable size to represent the immense astronomical spectroscopic data, and further detect unintended outliers that bear potential interest.

Data. The Mapping Nearby Galaxies at APO (MaNGA) dataset provides a 3D data cube for each galaxy in its catalog, which consists of a series of spectral flux at different wavelengths for each spatial pixel. It is superior to many of its predecessors not only in range of galaxies included but also in the abundance of details within each galaxy.

Methods. The original galaxy data is far too large to handle for most existing anomaly detection algorithms to operate on, due to both curse of dimensionality and memory constraints. We thus first seek to map the galaxy observations to some low-dimensional representation using autoencoder, and then apply various manifold learning techniques and anomaly detection algorithms to find suspicious outliers. Convolutional layers are also introduced to further capture the internal correlation both along the spatial and spectral axes.

Results. We found that in general, after being compressed to low dimension, we can reconstruct back most of the shapes of the original spectra with spatial pattern retained, at the price of losing many important peak signals.The embeddings proved to contain high level features and could be used to successfully predict some attributes of the galaxy. Several instances of interesting outliers are detected due to their uniqueness in various aspects.

DAP Committee members:

Barnabás Póczos ⟨ bapoczos@cs.cmu.edu ⟩ (MLD);
Aarti Singh ⟨ aartisingh@cmu.edu ⟩ (MLD);
Brett Andrews ⟨ andrewsb@pitt.edu ⟩ (PITT).

1 Introduction

Modern astronomical surveys are capable of producing immense amount of data. The Sloan Digital Sky Survey (SDSS) has collected rich features for millions of astronomical objects covering almost one-third of the sky. This creates a platform with huge potential for us to better explore and understand our universe. In spite of the automated pipelines currently being used, it is still hard for human experts to inspect the data manually in order to discover particular phenomena to their interest.

Machine learning algorithms naturally fit in such context to provide automated solutions. For example, [1] used machine learning methods to successfully detect contaminated redshift estimations, while [2] classified quasar from stars with high accuracy. Though such supervised approaches could generally achieve satisfactory results, especially in many binary object classification applications, they either require a priori knowledge of the dataset for labelling, or take huge computation time to manually simulate synthetic instances for training, thus limiting their ability to scale.

A random-forest based method is proposed in [3] to automatically detect outliers of different classes without supervision, and picked out many interesting objects that have never been closely examined before. However, they treated spectra from the same galaxy as separate individuals, which makes little use of the spatial correlation within the galaxy.

In this project, we aim to use unsupervised machine learning to examine a newly released astronomical spectroscopic dataset for thousands of galaxies. We approach the problem by first building low-dimensional representations from the huge original inputs using convolutional autoencoders, which are able to capture both the spatial and spectral correlation within a galaxy, to enable further application of visualization and anomaly detection.

2 Data

The Mapping Nearby Galaxies at APO (MaNGA) [4] is one of the latest programs in the Sloan Digital Sky Survey (SDSS). Unlike a traditional survey that usually has the capacity of a few hundred of target galaxies and samples only a small sub-region at the center for each galaxy, the MaNGA survey aims at ultimately 10,000 nearby galaxies, and also investigate a galaxy as a whole using 17 simultaneous 'Integral Field Units' (IFUs), providing rich and complex internal structure.

Up to the most recent SDSS Data Release 14, the MaNGA dataset contains data cubes for 2,812 galaxies (indexed by [plate]-[ifudesign]) in total, before some duplication and commissioning plates are removed. Each galaxy has a map from spatial pixels (spaxel) of a 2D image to the spectrum at that location, resulting in an $N \times N \times 4563$ cube. The size of spatial dimension depends on the IFU design used for the observation, and varies across the dataset. The spectral dimension contains the flux at 4563 different wavelengths, ranging from 3621.6Å to 10353.8Å with equal space. Along with

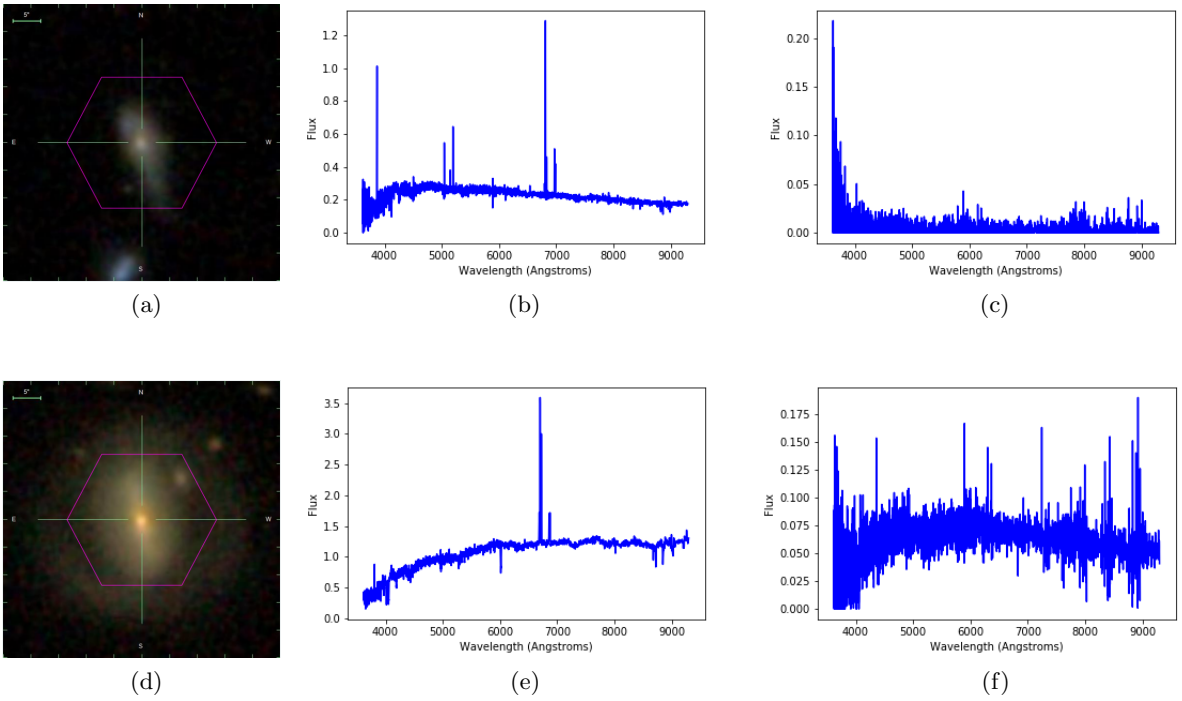
the raw observation data, we also have metadata references that include the buildout of the fiber bundles and summary of the targets observed.

During pre-processing, we first clipped all the flux values to be positive, since there exists several rare negative flux values due to instrumental error. In order to unify the shape of input to our model, we discarded a tiny portion of cubes that are smaller than 34×34 in the spatial dimension, as well as some observations that were flagged as 'DO_NOT_USE' due to poor measurement quality. The remaining cubes are resized to a fixed size in the spatial dimensions using linear interpolation. The spectral dimension are cut to have the length of an order of 2 to make pooling and unpooling easier, where the last few flux are removed with little information loss as they frequently contains huge noise. We further replaced all the flux between 5574Å to 5585Å with interpolation of their neighbors, where an instrumental noise occurs across all the observations due to residual sky emission. This finally leads to 2,627 data cubes of size $N \times N \times 4096$.

Figure 1 shows the galaxy images and spectra at two typical spaxels for two random MaNGA galaxies. The spectra at boundary are usually close to 0 and noisy, while the spectra near the center generally demonstrate more information of the galaxy. Different galaxies might have very different flux values due to its brightness, slopes due to red/blue-ness, peak locations due to redshift, and even jiggling patterns. The common peak of the two galaxies are commonly referred to as the H-α emission line, which always appears at 6562.8Å after deredshifting. However, we did not shift the spectra back to rest-frame as most astronomer would do, since this will almost double the length of the spectra and cause severe memory issues. This means that the flux at a wavelength are what we have observed directly on earth, instead of what we would see if we were in the target galaxy. We hope the model can learn the redshift itself by identifying the relative shift of the peaks, but in general we do believe that the result will be better with deredshifting.

3 Method

Autoencoders (AE) are neural networks that aim to reconstruct the input with minimum distortion. The encoder transforms an input into a lower-dimensional representation, and the decoder tries to map it back. By comparing the reconstructed output from decoder against the original input, the whole model can be trained in an unsupervised way, and learn to extract important features and compose an latent embedding, providing an aggregated view of the input data that is of manageable size for subsequent tasks like clustering and anomaly detection. The general idea of the AE family is that the unique pattern of a single instance could be extracted as the embedding, while the general distribution and other global information can be stored in the model parameters. The mapping in both the encoder and the decoder are typically done by stacks of fully connected layer. Figure 2a shows an one-layered AE, where x is the input vector, z is the output of encoder, i.e. the low-dimensional embedding, and x' being the output of decoder, i.e. the reconstruction. The encoder and decoder in this simple case is just two weight matrices. We refer interested readers to [5] for more details.



(a) (b) (c)

(d) (e) (f)

Figure 1: Galaxy 8261-9102 and 8391-9101's **(a,d)** images; **(b,e)** spectra at the center; **(c,f)** spectra near boundary.

Deep convolution-based neural network has remarkably outperformed traditional artificial neural networks, and many of its other competitors in most image-related tasks in the past few years. It is natural to replace the fully connected layers in the auto-encoder with convolutional ones to get convolutional auto-encoders (CAE). When applied to image-like inputs, CAE can better capture the spatial correlation between neighbor pixels, while AE can only treat each pixel as an independent feature. CAE also greatly reduce redundancy in model parameters when the input size is large as filters are only applied locally instead of globally, thus can be scaled well to large inputs such as MaNGA cubes.

3.1 Architecture

We mainly focused using CAE for dimensionality reduction. In the encoder, we incorporated convolution layers to detect local pattern, followed by a pooling layer to downsample and combine information in a neighborhood. Multiple blocks of such combination were stacked together to extract high-level features, gradually reducing the dimensionality of the input cubes to finally get a desired low-dimensional representation vector.

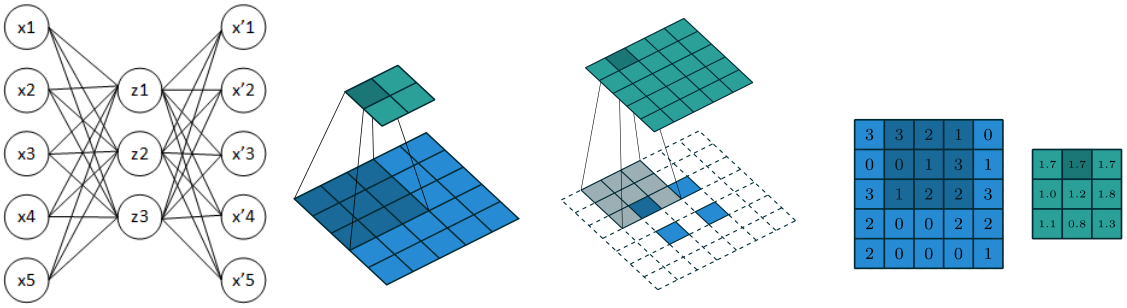
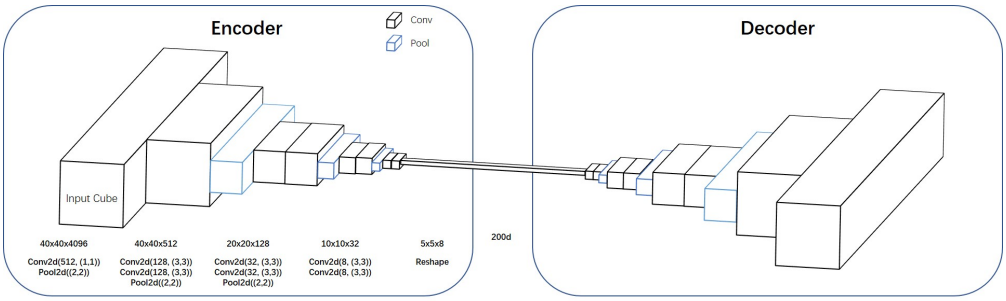


Figure 2: **(a)** Demonstration of an one-layered AE. **(b,c)** Illustration of convolution and transposed convolution with kernel size of 3×3, with the bottom blue image being input and top green image being output. The convolution operation maps the 5×5 input to 2×2 without padding, while the transposed convolution does the opposite. As shown, transposed convolution can be seen equivalently as convolution with a fractional stride, where zeros are inserted between input pixels. **(d)** Illustration of pooling, in this case the output is the numerical average of a neiborhood in the input.

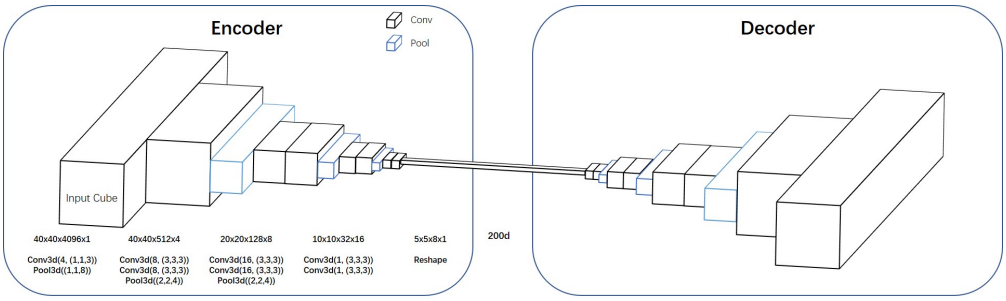
The vector was then passed to the decoder, where we performed the exact opposite of the operations in the encoder. Transposed convolution, sometimes referred to as deconvolution, was performed. With a large stride, it would act like an unpooling layer to get back the lost neighboring information during pooling. However, such reconstruction, although enlarged in size, would be rather sparse. As pointed out in [6], such unpooling would result in low-resolution checkerboard artifacts, so we adopted their solution to use a nearest-neighbor interpolated resize layer followed by a convolution layer to further blend the sparse pixels and densify the outputs, which has shown success in super-resolution problems [7]. Normal deconvolution were used in decoder to reverse the convolution in encoder. Figures 2b to 2d [8] illustrates and compares the difference of these operations.

The depth of the MaNGA cubes is much larger than that of a typical image which has only 3 channels (RGB). This leaves us multiple choices in terms of how to best utilize its internal correlation in both the spatial dimensions and along the spectral axis. We experimented several different flavors of CAE, where each variants has its own advantage in some aspect of the reconstruction. Detailed model specifications are shown in Figure 3.

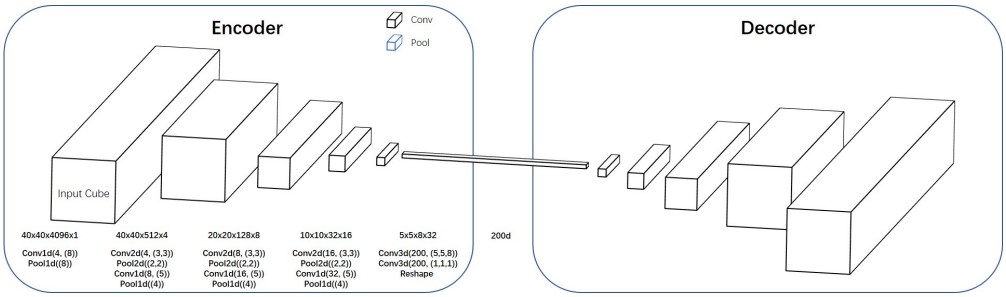
2D CAE The first and simplest model we tried is the traditional CAE that have been widely used on regular image data. Each slice of the cube along the spectral dimension, representing the mapping from spaxels to their flux at a certain wavelength, is treated as an independent 'channel'. The convolution kernels will have the same depth as the input to the layer, and the depth of output for a layer will be the number of different filters used. Convolution in this case will only be performed along the two spatial dimensions only, and the operation we performed on the spectral



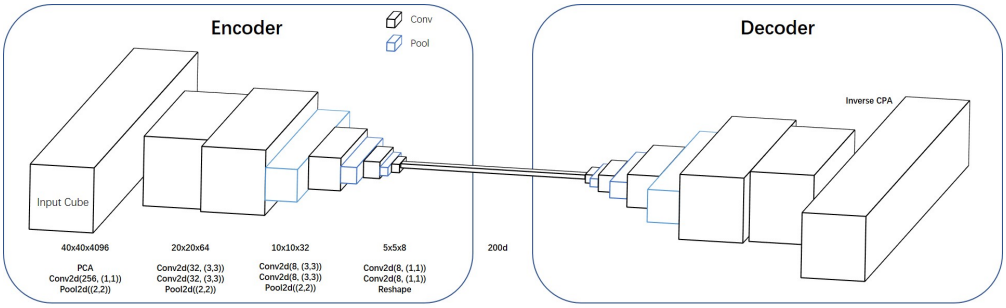
(a)



(b)



(c)



(d)

Figure 3: Model specifications for **(a)** 2D CAE; **(b)** 3D CAE; **(b)** 1D-2D CAE; **(b)** 2D CAE on compressed spectra.

dimension will be somewhat similar to fully-connected layers. With such a large input size, this inevitably requires a huge number of parameters, but the output depth after each convolution layer can be tuned by number of filters used, thus we can easily compress the cube to a desired low-dimensional representation.

3D CAE A major drawback in the previous design is that the spectrum at a spaxel is obviously not independent features. To further utilized the continuity and order information that lie within, we want to do convolution along the spectral dimension as well. This leads to CAE with 3D-convolution. The MaNGA cube can be imagined as a four-dimensional tesseract, where the last dimension is of size 1, representing channel. The convolution kernels will now be tiny cubes that convolve along all three dimensions. At a given point, the filters will not only captures information at nearby spaxels, but also at nearby wavelengths.

Although this seems like an ideal model to solve the problem, in practice the 3D convolution will generate outputs that have almost the same size as their inputs. Multiple stacks of 3D convolution layers near the top of the encoder will take up a huge amount of memory. This add constraints on the model size itself, as well as some hyperparameters such as the batch size during training, which we found lead to sub-optimal solution at convergence.

1D-2D CAE A 3D convolution kernel can extract both spatial and spectral correlations, but one may argue that these two correlations might be orthogonal to each other. For example, in a $3 \times 3 \times 3$ cube, the center block should have high correlation with the 6 blocks that are directly adjacent to it, but might not depend as heavily on its diagonal neighbors. This inspired us to take turns to perform 2D convolution along the spatial dimension only, treating the wavelengths as totally separate examples in a batch, and 1D convolution along the spectral dimension. This is equivalent to doing 3D convolution, with constraints such that all subsequent channels of a kernel must be a reweighted duplicate of the first channel. This further reduce the model complexity.

2D CAE on compressed spectra Instead of doing alternating 1D and 2D convolution, we divide the whole process into two stage. In the first stage, spectra within a given cube are treated as individual examples, and traditional dimensionality reduction techniques are performed to get a rather compressed representation for each spectrum. PCA and shallow AE are two common choices. Then in the second stage we convolve along the spatial dimension using 2D CAE. Note that after the compression, each element in the compressed spectral dimension represents a high level feature of the original spectrum, such as mean, slope and location of its peak, thus there won't be any ordered information and features are indeed independent, making it unnecessary to convolve along the spectral dimension.

4 Experiments

4.1 Training

In our experiment, we picked $N = 32$ so that the input MaNGA cubes would have the shape of $32 \times 32 \times 4096$. Activation function is chosen to be ELU [9], which appears to work better than ReLU. We used average pooling to downsample the input, which turned out to work better than max pooling in our case. After the encoder, we would obtain a 4×4 cube to be our desired low-dimensional representation. We calculated the reconstruction error by comparing the output of decoder and the original input, and used the mean error to be reconstruction loss. Mean absolute error turned out to produce reconstruction with more accurate mean and slope, but totally ignore the peak locations that are of great importance, while mean square error seemed to cause the whole curve to be dragged up by the peaks as it is less robust, so we chose to go with MAE. The first 2,000 examples were selected as training set, and the rest as validation set to monitor the training performance. The models were re-trained on the whole dataset for the usage of downstream outlier detection, since we will not be applying it to new data. We optimized the model for 50 epochs using the ADAM [10] optimizer, with learning rate shrinking to half every 10 epochs. Batch size was set to 50 for 2D CAE, and 10 when the memory complaints for the rest cases.

We applied batch normalization [11] after each convolution/transposed-convolution, which normalizes the layers to follow the standard Gaussian distribution. There has been a heated debate around whether the trick works besides the internal-covariate-shift theory proposed by the original authors, but empirically we found that adding such operation indeed helps the network converge to a much better optimum during training.

We polished the loss function to better fit the MaNGA dataset semantics. The IFUs are in fact of a hexagon shape, so that the values in the four corners of the spatial square are often very weak, with excessive noise near boundary as shown in Figure 1f. Also, since the sensors are intentionally aiming at the center of the target galaxy, most useful signals describing the galaxy will lie in the center spaxels. Thus we multiplied the errors in the loss function with a pre-defined mask, weighting up the cost of making wrong reconstruction in the center, and down weighting that near boundary.

In addition to the reconstruction loss, we attached another stream of network after the encoder, which takes in the output embeddings of the encoder, and uses two layers of fully-connected layer to make a single prediction of the average redshift. The regression MSE loss will be reweighted by a λ and added back to form the final loss function. This also gives us a more intuitive metric to evaluate how well the model (encoder part) is learning.

For PCA-2D, besides computing loss by comparing the flux, we also tried another approach by computing MSE on reconstructed principal components. The principal components are normalized to balance their contribution to the total reconstruction loss. We also find it useful to manually boost up the weight for the first two principal components, which account for the mean and slope

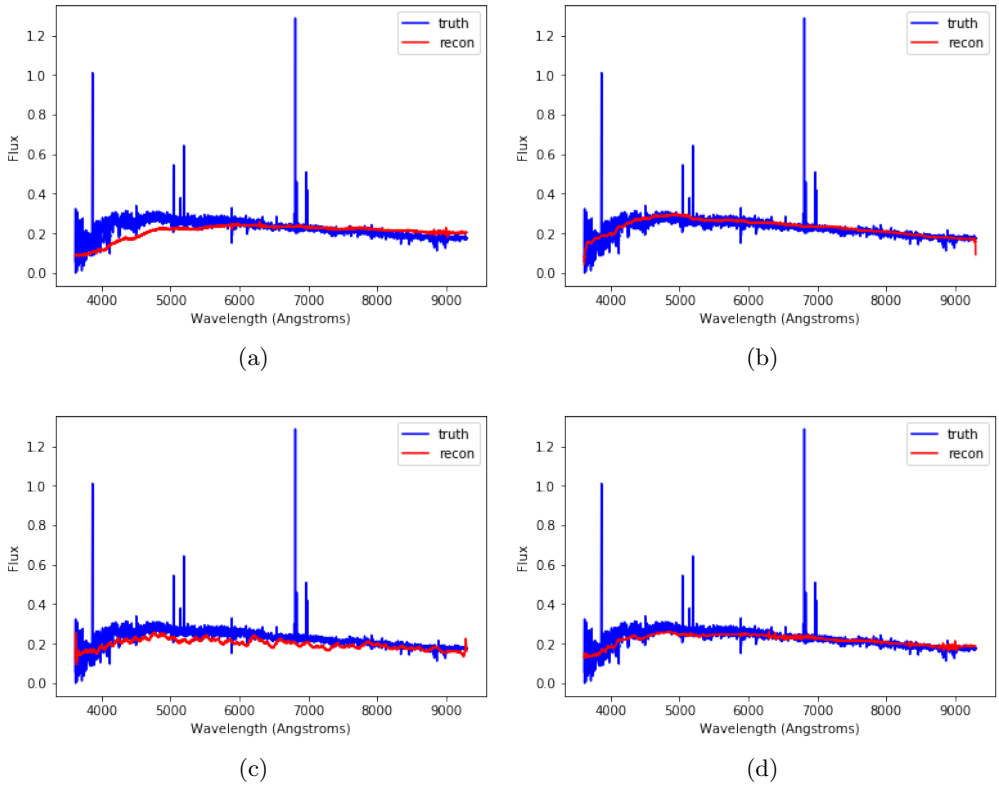


Figure 4: Reconstructions of the spectrum at the center spaxel for galaxy 8261-9102 by **(a)** 2D CAE; **(b)** 3D CAE; **(b)** 1D-2D CAE; **(b)** 2D CAE on compressed spectra.

of the original spectrum, over the rest which each controls the peak or jiggling pattern at certain wavelength. However, this turned out to be not working as well, due to the volatility in the principal components.

4.2 Evaluation

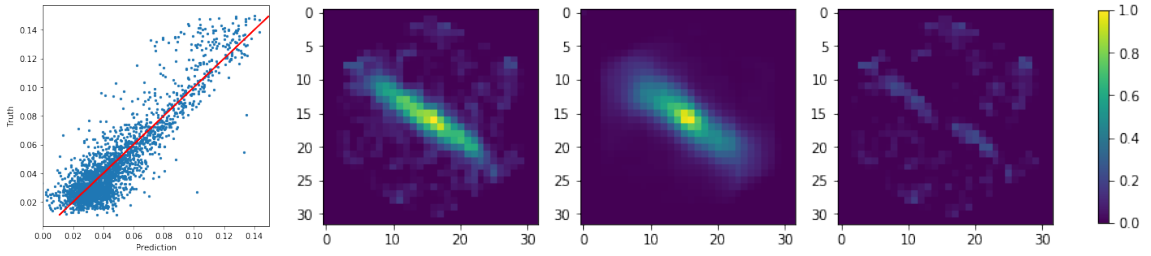
We evaluated the model mainly by the quality of their reconstruction. Figure 5 shows the reconstruction of the center spaxel for a typical galaxy using the four models discussed. The 2D CAE can generally put the reconstruction around the correct mean, while the 3D CAE can fit the original curve much more accurately. The 1D-2D CAE seems to be consistently repeating a certain local jiggling pattern to form the whole spectrum, which might be a result of using too few parameters. PCA-2D is equivalent as fixing a pre-trained fully connected layer on the top for 2D CAE, and it turned out to reduce the training difficulty a bit. However, some occasional noise pattern can often be observed for PCA-2D, mainly due to the volatility mentioned earlier. Unfortunately none of

Model	Recon MAE Loss	Pred MSE Loss
2D CAE	49831	0.00486
3D CAE	44387	0.00158
1D-2D CAE	48742	0.00478
PCA-2D CAE	43978	0.00190

Table 1: Performance of different models. The reconstruction loss is the mean absolute difference between the reconstructed cube and the original, summed across the whole cube with the spaxels within 2 spaxels to the boundary down-weighted by a factor of 5. The prediction loss is the mean squared error between the prediction and actual average redshift.

these models has captured the emission peak, though, since we are using MAE in the first place. A MSE reconstruction loss, however, would fail to get even the mean right.

As shown in Table 1, the 3D CAE outperformed others in the both metrics. We further looked into its reconstruction performance in the spatial dimensions. The predictions generally lies around the red reference line of perfect prediction, and performing much better than just guessing the mean, which shows that our encoder is indeed extracting high-level information. In Figure 5b, we take out a slice from the cube at a certain wavelength to see the spatial distribution of flux. It can be seen that the reconstructed slice can capture spatial pattern of this particular galaxy regardless of its irregular shape.

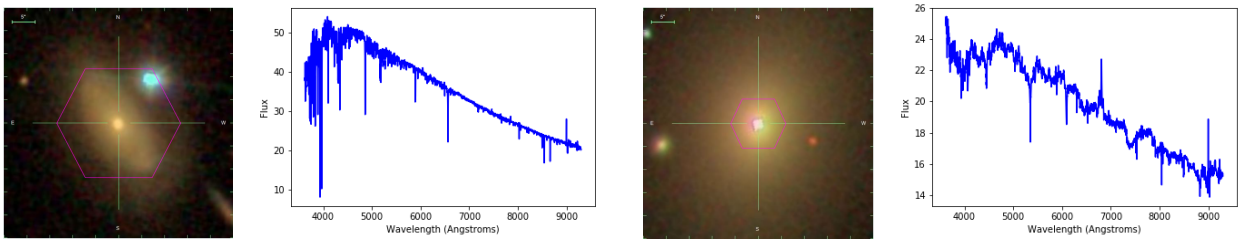


(a) (b)

Figure 5: Further results for 3D CAE: **(a)** predicted v.s. truth average redshift for all galaxy cubes; **(b)** Original, reconstructed and residual (difference between original and reconstructed) spatial flux map for galaxy 8338-12702 at 5739.8 Å, under same color mapping scheme.

4.3 Outlier detection

With the dimensionality reduced to an acceptable range, we experimentally tried many widely-used manifold learning techniques, including from Isomap to multidimensional scaling, to visualize the distribution and identify outliers. We also performed distance based outlier detection algorithm,



(a) (b)

Figure 6: Images and spectra at the weird/center spaxel of example galaxies detected as outliers: **(a)** a star in the foreground at top right corner; **(b)** a blazar.

such as the k-NN based method, where a galaxy is scored by the its average distance to its top-k nearest neighbors. We found that several galaxy are consistently far away from the majority clusters regardless of the method used, showing high probability of being an outlier.

There are also some galaxies that all models failed to reconstruct well. In this context, this in fact might be a good thing as it actually gives us a more straightforward way to classify them as outlier based on the reconstruction loss. Those reconstructed with large error might well have different distribution from the rest and are likely to be weird.

During our initial runs of outlier detection, many of the observations that have foreground stars lying in between the IFU and the target galaxy are identified as outliers. These cubes will have unexpected large flux values at a certain spatial patch near its boundary, thus triggering the model to treat them as abnormal samples. Figure 6a is an example, where a foreground star triggers huge flux with downward slope (indicating it's blue) at the top right corner. While this is indeed an true outlier, we in fact already have the knowledge of the star's existence. In order to let the model extract more 'unknown unknowns' that haven't been discovered before, we mask out these patches in the input data, as well as in the reconstruction loss used during training.

We continue to manually inspect the galaxy cube and identify the source of their weirdness. Figure 6 shows several typical examples. Figure 6b is one of the mostly picked galaxy, which turned out to be an blazar, one of the most energetic phenomena in the universe.

5 Conclusions

This projects mainly attempted to build low-dimensional representation for the MaNGA dataset, which contains spatial maps of astronomical spectroscopic data for galaxies, using convolutional autoencoders in an unsupervised fashion. Multiple variants of CAE with different focus and thus different architectures have been experimented and evaluated. Current results still have huge

space for improvement, thus possible directions are discussed. From the dimensionality-reduced summarizing embeddings, preliminary outlier detection is performed to locate weird galaxies among the whole dataset. A potential future work is to detect outlier spaxels within a single galaxy cube.

References

[1] B. Hoyle, M. M. Rau, K. Paech, C. Bonnett, S. Seitz, and J. Weller, "Anomaly detection for machine learning redshifts applied to SDSS galaxies," , vol. 452, pp. 4183–4194, Oct. 2015.

[2] C. M. Peters, G. T. Richards, A. D. Myers, M. A. Strauss, K. B. Schmidt, Ž. Ivezić, N. P. Ross, C. L. MacLeod, and R. Riegel, "Quasar Classification Using Color and Variability," , vol. 811, p. 95, Oct. 2015.

[3] D. Baron and D. Poznanski, "The weirdest SDSS galaxies: results from an outlier detection algorithm," , vol. 465, pp. 4530–4555, Mar. 2017.

[4] K. Bundy, M. A. Bershady, D. R. Law, R. Yan, N. Drory, N. MacDonald, D. A. Wake, B. Cherinka, J. R. Sánchez-Gallego, A.-M. Weijmans, D. Thomas, C. Tremonti, K. Masters, L. Coccato, A. M. Diamond-Stanic, A. Aragón-Salamanca, V. Avila-Reese, C. Badenes, J. Falcón-Barroso, F. Belfiore, D. Bizyaev, G. A. Blanc, J. Bland-Hawthorn, M. R. Blanton, J. R. Brownstein, N. Byler, M. Cappellari, C. Conroy, A. A. Dutton, E. Emsellem, J. Etherington, P. M. Frinchaboy, H. Fu, J. E. Gunn, P. Harding, E. J. Johnston, G. Kauffmann, K. Kinemuchi, M. A. Klaene, J. H. Knapen, A. Leauthaud, C. Li, L. Lin, R. Maiolino, V. Malanushenko, E. Malanushenko, S. Mao, C. Maraston, R. M. McDermid, M. R. Merrifield, R. C. Nichol, D. Oravetz, K. Pan, J. K. Parejko, S. F. Sanchez, D. Schlegel, A. Simmons, O. Steele, M. Steinmetz, K. Thanjavur, B. A. Thompson, J. L. Tinker, R. C. E. van den Bosch, K. B. Westfall, D. Wilkinson, S. Wright, T. Xiao, and K. Zhang, "Overview of the SDSS-IV MaNGA Survey: Mapping nearby Galaxies at Apache Point Observatory," , vol. 798, p. 7, Jan. 2015.

[5] G. Hinton and R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504 – 507, 2006.

[6] A. Odena, V. Dumoulin, and C. Olah, "Deconvolution and checkerboard artifacts," *Distill*, 2016. [Online]. Available: http://distill.pub/2016/deconv-checkerboard

[7] C. Dong, C. Change Loy, K. He, and X. Tang, "Image Super-Resolution Using Deep Convolutional Networks," *ArXiv e-prints*, Dec. 2015.

[8] V. Dumoulin and F. Visin, "A guide to convolution arithmetic for deep learning," *ArXiv e-prints*, Mar. 2016.

[9] D. Clevert, T. Unterthiner, and S. Hochreiter, "Fast and accurate deep network learning by exponential linear units (elus)," *CoRR*, vol. abs/1511.07289, 2015. [Online]. Available: http://arxiv.org/abs/1511.07289

[10] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014. [Online]. Available: http://arxiv.org/abs/1412.6980

[11] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *CoRR*, vol. abs/1502.03167, 2015. [Online]. Available: http://arxiv.org/abs/1502.03167