Reproducing SDSS Spectral Compression Using a Standard Autoencoder

Zehao Peng

May 20, 2025

1 Introduction

The Sloan Digital Sky Survey (SDSS) provides high-resolution optical spectra of galaxies that contain rich physical information, but the high dimensionality (typically 1000+ wavelength bins per spectrum) makes interpretation and visualization challenging. To address this, Portillo et al. (Portillo et al., 2020) proposed using a Variational Autoencoder (VAE) to perform non-linear dimensionality reduction of galaxy spectra. In this project, I reproduce the core findings of that work using a standard (non-variational) autoencoder.

2 Method

2.1 Data Processing

The dataset contains ∼64,000 galaxy spectra from SDSS, originally compiled for the VAE study by Portillo et al. Each spectrum is sampled over 1000 logarithmically spaced wavelength bins ranging from approximately 3388 Å to 8310 Å. The raw spectra are reconstructed from PCA coefficients using the `sdss_corrected_spectra` utility from `astroML` (Vanderplas et al., 2012), then normalized by their respective PCA norms:

$$\mathbf{x}_i = \frac{\text{rawspec}_i}{\text{norm}_i}, \tag{1}$$

where $\mathbf{x}_i$ is the normalized spectrum of the i-th galaxy.

Normalization serves three main purposes: (1) it removes variations in total flux between galaxies, allowing the model to focus on spectral shape rather than brightness; (2) it stabilizes training by ensuring that all inputs have similar magnitudes; and (3) it aligns with the preprocessing used in the original VAE paper, enabling direct comparisons.

Each spectrum is associated with a pixel-wise error estimate σ_i, which is used to compute weights used later in the loss function:

$$\mathbf{w}_i = \frac{1}{\sigma_i^2 + \epsilon}, \quad \epsilon = \frac{1}{2 \times 10^6} \tag{2}$$

ϵ is used to cap extremely small uncertainties and avoid exploding weights. Pixels with non-physical or flagged values (e.g., from bad sky subtraction) are masked using a boolean mask $\mathbf{m}_i$ and assigned zero weight:

$$\mathbf{w}_i[\mathbf{m}_i = 1] = 0. \tag{3}$$

Please note that for a fair reproduction, all the data processing steps mirror exactly those from the work of Portillo et al. The dataset is randomly split into 47,000 training and 16,000 validation

spectra. All data is converted to 32-bit floating point format for compatibility with Apple's MPS backend (my device).

2.2 Network Architecture and Training

The autoencoder is a fully-connected feedforward neural network designed to compress and reconstruct the 1000-dimensional spectral input. It consists of two symmetric components:

- **Encoder:** Maps the input spectrum $\mathbf{x} \in \mathbb{R}^{1000}$ to a latent representation $\mathbf{z} \in \mathbb{R}^6$ via:

$$\begin{aligned} \mathbf{h}_1 &= \text{ReLU}(\mathbf{W}_1\mathbf{x} + \mathbf{b}_1), \quad \mathbf{W}_1 \in \mathbb{R}^{703\times1000} \\ \mathbf{h}_2 &= \text{ReLU}(\mathbf{W}_2\mathbf{h}_1 + \mathbf{b}_2), \quad \mathbf{W}_2 \in \mathbb{R}^{94\times703} \\ \mathbf{z} &= \mathbf{W}_3\mathbf{h}_2 + \mathbf{b}_3, \quad \mathbf{W}_3 \in \mathbb{R}^{6\times94} \end{aligned}$$

- **Decoder:** Reconstructs $\hat{\mathbf{x}}$ from the latent representation:

$$\begin{aligned} \mathbf{h}_3 &= \text{ReLU}(\mathbf{W}_4\mathbf{z} + \mathbf{b}_4), \quad \mathbf{W}_4 \in \mathbb{R}^{94\times6} \\ \mathbf{h}_4 &= \text{ReLU}(\mathbf{W}_5\mathbf{h}_3 + \mathbf{b}_5), \quad \mathbf{W}_5 \in \mathbb{R}^{703\times94} \\ \hat{\mathbf{x}} &= \mathbf{W}_6\mathbf{h}_4 + \mathbf{b}_6, \quad \mathbf{W}_6 \in \mathbb{R}^{1000\times703} \end{aligned}$$

The autoencoder is trained by optimizing the trainable parameters θ, which include all weights and biases in the encoder and decoder networks. The objective is to minimize a weighted mean squared error (MSE) loss over the training data:

$$\theta^* = \arg\min_\theta \mathcal{L}(\mathbf{x}, \hat{\mathbf{x}}_\theta, \mathbf{w}) = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{1000} w_{ij}(x_{ij} - \hat{x}_{ij})^2 \tag{4}$$

where $\hat{\mathbf{x}}_\theta$ is the network's reconstruction of $\mathbf{x}$ given parameters θ, N is the batch size, and j indexes the wavelength bins. The network hyperparameters are chosen per Portillo et al.'s findings, which show that 6 latent parameters and a 1000-703-94-6-94-703-1000 architecture achieve the best results in terms of reconstruction loss. This setup is the point beyond which further increases in model capacity yielded diminishing improvements. The intuition behind this autoencoder is that by setting minimizing reconstruction loss as the optimization objective and forcing the model to funnel information through a 'bottleneck', the latent parameters will capture the features most important for a high-fidelity reconstruction. In essence, this is similar to PCA. However, we are not restricted to reconstructing the spectra only via linear combinations of the latent parameters, which can yield a more interesting feature map. (Latent space)

Training was conducted using PyTorch Lightning (Falcon & The PyTorch Lightning team, 2019), which simplifies training loop management, and tracked using Weights
Biases (Biewald, 2020). Training is performed using the Adam optimizer (Kingma & Ba, 2017) with an initial learning rate of 10^{-3}. Essentially, we are performing a variation of stochastic gradient descent on the loss landscape, with the goal of reaching a global minimum. The learning rate corresponds to the step size of this descent. A `ReduceLROnPlateau` scheduler reduces the learning rate by a factor of 10 after 5 epochs of no improvement in validation loss. Early stopping halts training if no improvement is seen for 10 consecutive epochs. This training regimen is also taken from Portillo et al.

Figure 1 shows the architecture schematic of the model.

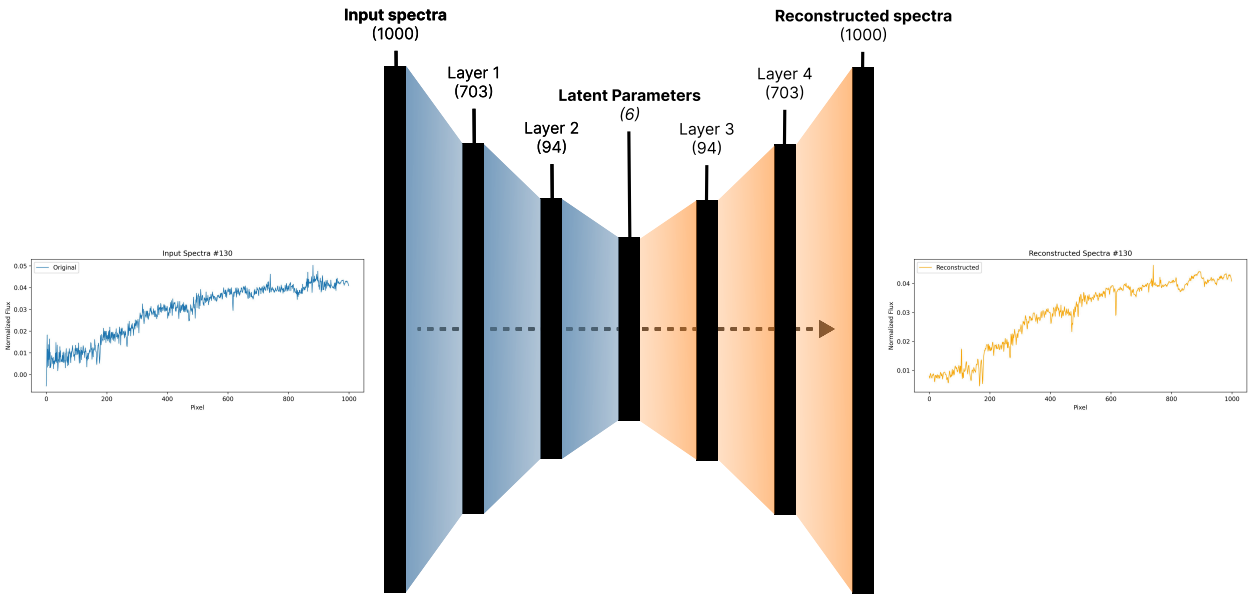

Figure 1: Schematic diagram of the autoencoder architecture used in this project.

3 Results

3.1 Latent Space Representation

After training until validation loss convergence, the 6-dimensional latent representations are visualized using corner plots and UMAP projections. These show clustering by galaxy spectral type, similar to the results of Portillo et al.

Figure 2 illustrates the distribution of the latent parameters.

The clustering of spectra by galaxy type indicates that the learned latent parameters can differentiate between physically distinct spectral classes for quiescent, emission line, narrow-line AGN, and broad-line AGN galaxies, suggesting that the autoencoder successfully captures key features of galaxy spectra in its low-dimensional representation.

We may also see this clustering behaviour more clearly by further compressing the six latent parameters into two via means of the Uniform Manifold Approximation Program (UMAP), which again applies a non-linear dimensionality reduction to our latent space.

Figure 3 shows the result.

Observe that the UMAP space has been roughly partitioned into four distinct regions corresponding to quiescent, emission line, narrow-line AGN, and broad-line AGN galaxies, coloured accordingly. This separation again indicates that the latent parameters are good discriminators of galaxy type. Due to the nature of UMAP, these axes are not very meaningful. They are therefore simply named UMAP1 and UMAP2.

3.2 Spectral Reconstruction

We will now examine its performance in spectral reconstruction.

Figure 4 shows one such example.

Notice that the general shape is captured very well, and the emission line around $\tilde{6}500$ Å (possibly the $H\alpha$ line) is reconstructed correctly.

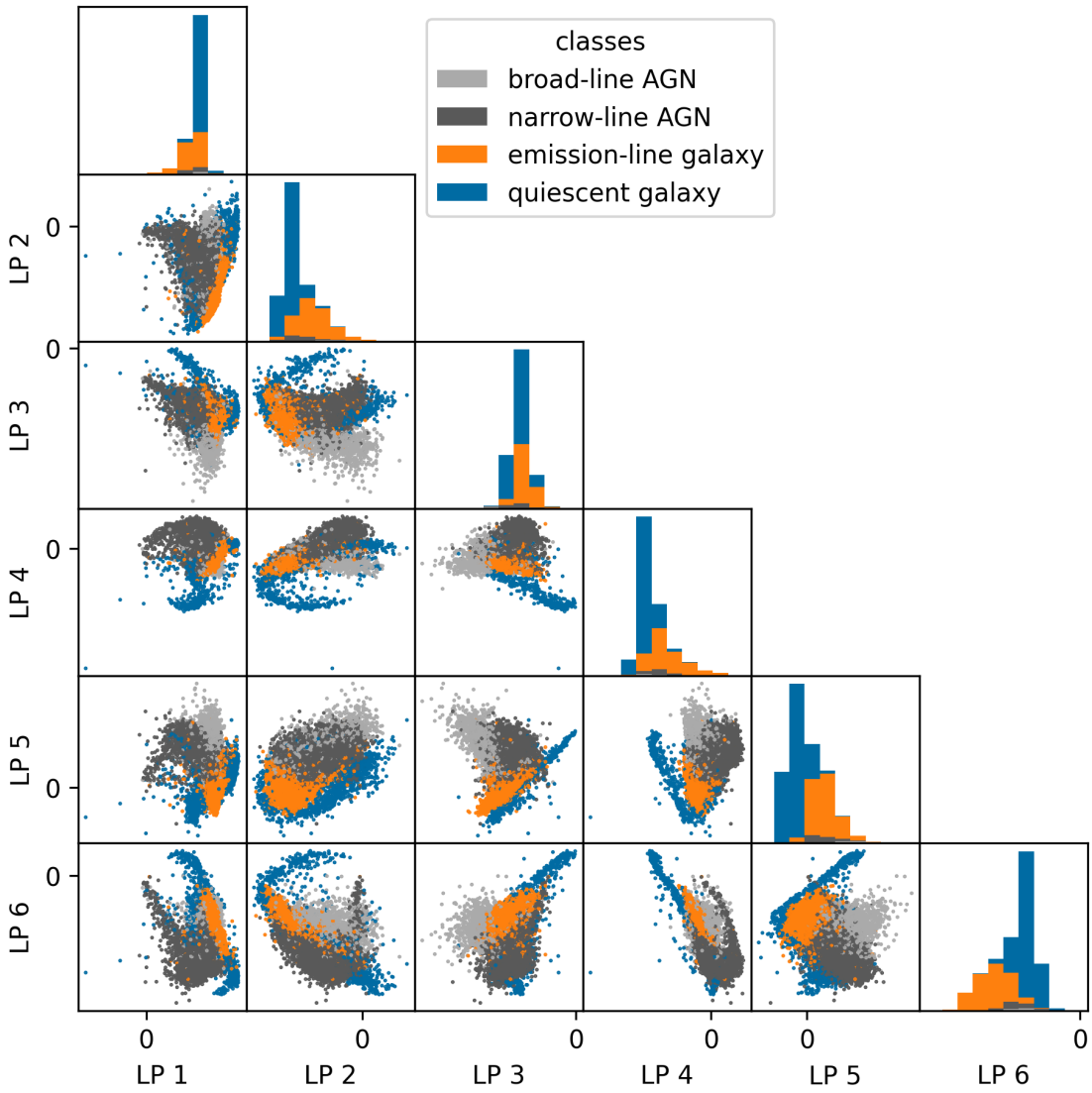

Figure 2: Corner plot showing the distribution of the 6 latent parameters across the dataset.

To further evaluate reconstruction quality, we generate a 2D histogram of residuals (difference between input and reconstruction) as a function of wavelength and inverse variance. This visualization helps assess whether reconstruction errors are systematically correlated with spectral features or uncertainty levels.

Figure 5 displays this.

Notice that in general, residuals are tightly distributed around 0, with flux deviations usually no more than 0.05 in either direction. This indicates that the reconstructions are generally very accurate. We also notice, however, that there are vertical bands at common emission line wavelengths (e.g., $H\alpha$, [OIII], [NII]), which indicate that the autoencoder struggles to fully reconstruct narrow, high-contrast features.

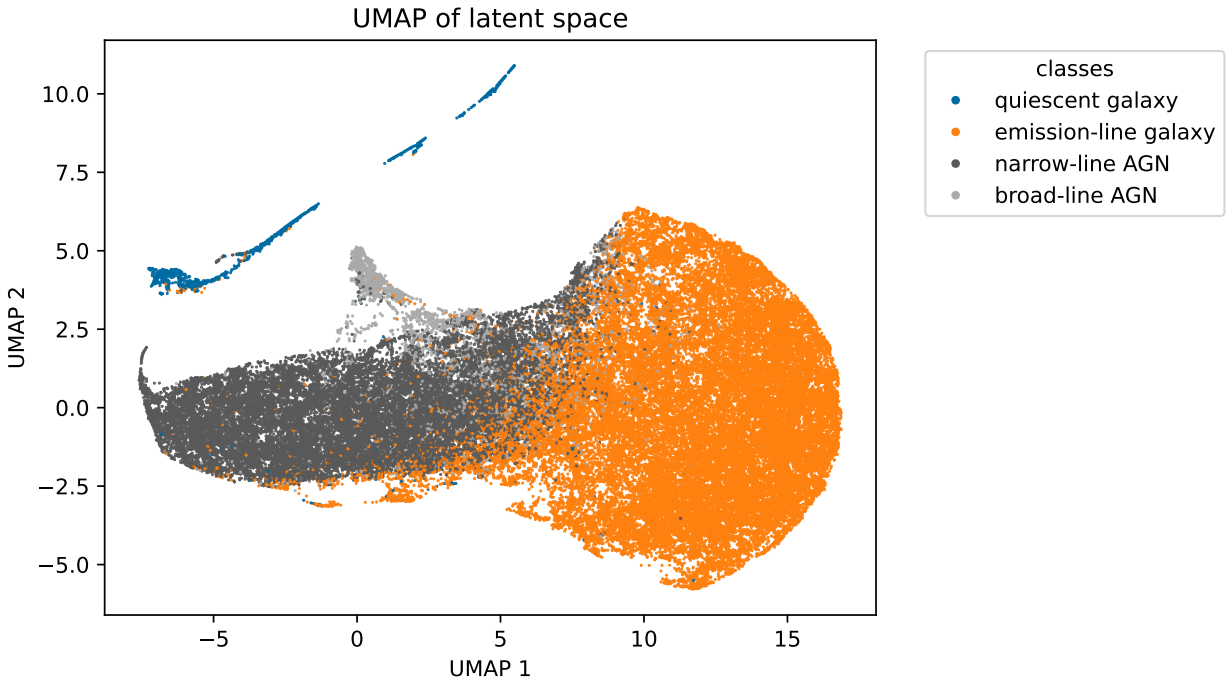

Figure 3: 2D UMAP embedding of the 6D latent space.

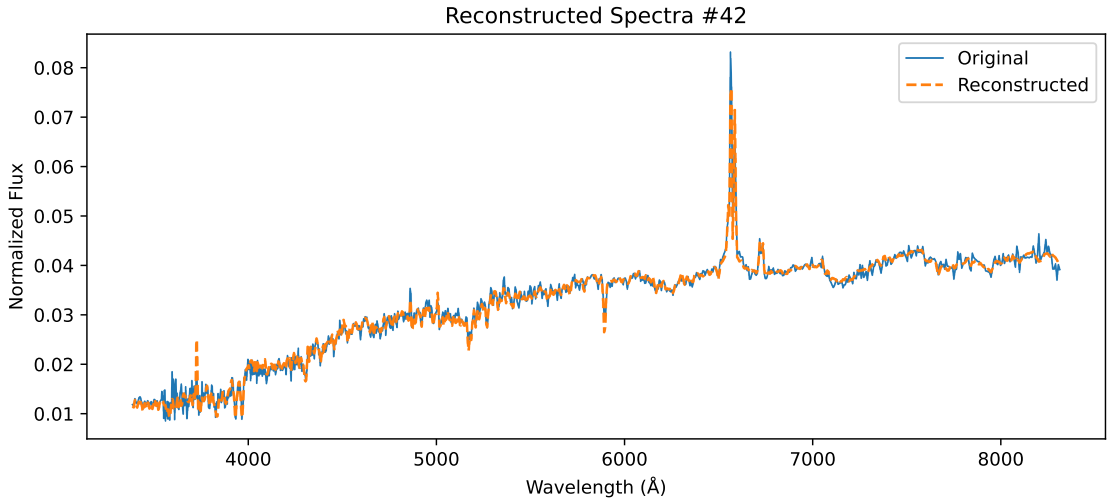

Figure 4: Randomly selected example of reconstructed spectra (orange) compared to original input spectra (blue).

4 Conclusion

This project demonstrates that a standard autoencoder can reproduce many of the core results of Portillo et al. (2020), including accurate spectrum reconstructions and, consequentially, a meaningful latent space capable of separating galaxies based on their spectrum.

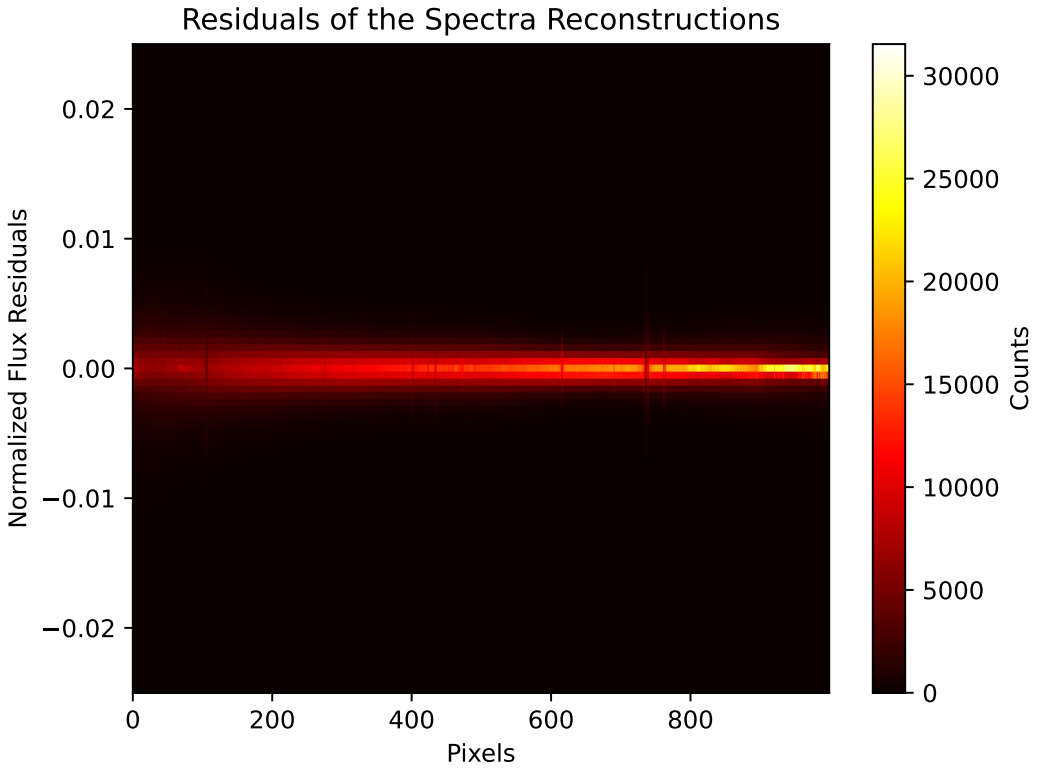

Figure 5: 2D histogram of reconstruction residuals binned by wavelength and residual flux.

References

Biewald, L. 2020, Experiment Tracking with Weights and Biases

Falcon, W., & The PyTorch Lightning team. 2019, PyTorch Lightning

Kingma, D. P., & Ba, J. 2017, Adam: A Method for Stochastic Optimization. https://arxiv.org/abs/1412.6980

Portillo, S. K. N., Parejko, J. K., Vergara, J. R., & Connolly, A. J. 2020, arXiv preprint arXiv:2002.10464

Vanderplas, J. T., Connolly, A. J., Ivezić, Ž., & Gray, A. 2012, in Conference on Intelligent Data Understanding (CIDU), 47–54