Adoption of the Transformer Architecture for Supernova Detection in the DESI Survey

by

Anthony J La Barca

Submitted in Partial Fulfillment of the

Requirements for the Degree

Bachelor of Science in Physics and Astronomy

Supervised by Professor Segev BenZvi

Department of Physics and Astronomy
School of Arts and Sciences

University of Rochester

Rochester, New York

2023

For Nonni, who never let me loose sight of what is important in life

Acknowledgments

My time at the University of Rochester was impacted by so many people, but as I finish writing this 50 page summary of the past year of research, I am reminded of those who made it truly special.

First and foremost I want to thank the Department of Physics and Astronomy for providing a nurturing environment over the past four years, in and out of the COVID-19 pandemic. I commonly refer to B&L and the POA as my second home, and without the efforts of each and every professor, administrator, and faculty member, my experience would simply have not been the same.

I would also like to thank Professor Segev BenZvi for his patience and intuition during the past year. He has been an absolute pleasure to work with, both in person and online, and he continues to fascinate me with the passion he brings to every meeting. He has shown me what true dedication to a field is, and one day I hope to impact someones life as he has impacted mine.

Professor Kelly Douglass was the first professor at the University of Rochester I had the pleasure of meeting, providing me with words of wisdom after the first college lecture I ever attended (during a tour of the school). She continued to offer insight and advice throughout my four years studying physics and astronomy, and she has been the image of a fair, caring professor ever since I first walked into her classroom.

Thank you to my father and mother, Salvatore and Gigi LaBarca, for their ongoing and unending support of my work and my sisters, Marisa and Carrie, who always find a way to make me laugh and strive for greatness. Thank you to my friends on campus and at home who deal with my egregiously packed school-music-ultimate schedule, and my peers, too numerous to mention by name, who shared scientific advice and late nights together. Finally, thank you to Grace, for all of our astronomy discussions, but more importantly, for reminding me that there are always 5 minutes in a day to enjoy the company of those you love.

Abstract

The identification and classification of supernovae transients are common hindrances in observational astronomy as they require additional spectroscopic observations and traditional autonomous classification techniques that are computationally expensive. In an effort to create a more efficient and autonomous classification methodology, deep learning architectures have previously been trained and implemented for use on the Dark Energy Spectroscopic Instrument (DESI) pipeline. Despite being able to precisely classify some spectra, the networks are unable to definitively classify many more, leading to a recall rate of approximately 20%. The Spectral ViT is a vision transformer-based architecture trained for use on the DESI pipeline. When trained on spectra with redshift removed according to the fits provided by the DESI pipeline, the Spectral ViT (V1) is able to recall 26% more targets while also applying a stricter prediction cut, resulting in only a 13.3% decrease in performance. The Spectral ViT V2 was trained for the purpose of eliminating dependence on the DESI redshift fit. While unable to match the sub-type classification power of the Spectral ViT V1, the model shows capability in acting as a classifier for progenitor and supernovae type, with precision values of 86.5% and 91.8% respectively. We find the Spectral ViT V1 to be a possible replacement for the existing classification network pending further testing and the Spectral ViT V2 to be a supernovae filter and supplemental classifier.

Contents

Acknowledgments **iii**

Abstract **v**

List of Tables **viii**

List of Figures **ix**

1 Introduction **1**

1.1 Supernova Classification 2

1.2 Dark Energy Spectroscopic Instrument 4

1.3 Past Approaches to Spectroscopic Identification 5

1.4 Supernova Classification with Machine Learning 6

2 Deep Learning Techniques **9**

2.1 Multi-Layer Perceptrons 9

2.2 Convolutional Neural Networks 11

2.2.1 Using CNNs on Spectroscopic Data in Astronomy 13

2.3 From CNNs to Transformers 16

2.3.1 ViT on Spectroscopic Data 18

3 Creation and Training of Spectral ViT **20**

3.1 Creation of Spectral ViT 20

3.1.1 Validation of Spectral ViT Architecture 24

3.2 Creation of Synthetic DESI Spectra 25

3.2.1 Pre-Processing of DESI Spectra 27

3.2.2 Transformer Training 29

3.3 Training of Neural Networks 30

3.3.1 CNN Training 30

4 Spectral ViT Model Performance **33**

4.1 Spectral ViT V1: A suitable replacement 33

4.2 Spectral ViT V2: Specialized Classifier 35

4.2.1 Alternative Applications of the V2 Classifier 37

5 Conclusions **40**

A Diagrams of Neural Networks **47**

List of Tables

3.1 Hyperparameters of the CNN used to classify DESI spectra. CNN adapted from Sepeku (2022). 28
3.2 Hyperparameters of the smaller and larger Spectral ViT architecture used to classify DESI spectra. 29

List of Figures

1.1 Supernova Classification 2
1.2 Publicly Classified SNe captured by DESI 4
1.3 SNe Influencing DESI Redshift Fit 5
1.4 Spectral Template of SN Ia SNe 7

2.1 Diagram of a Multi-Layered Perceptron 10
2.2 Example of a CNN Unit 12
2.3 Traditional CNN Architecture for Classification 13
2.4 CNN diagnostics 14
2.5 CNN's Confidence in Classification 15
2.6 CNN diagnostics with a confidence cut 15
2.7 Transformer Architecture 17
2.8 ViT Architecture 18

3.1 Positional Embeddings for ViT 21
3.2 Transformer Attention 23
3.3 Synthetic spectra 24
3.4 Verification of ViT: Synthetic Dataset 26

3.5 Spectra Pre-Processing – Redshift Corrected 27
3.6 Spectra Pre-Processing – Not Redshift Corrected 27
3.7 CNN Training 28
3.8 Training of Spectral ViT: V1 29
3.9 Training of Spectral ViT: V1 Big 30
3.10 Training of Spectral ViT: V1.2 30
3.11 Training of Spectral ViT: V1.3 31
3.12 Training of Spectral ViT: V2 32

4.1 Spectral ViT V1 Diagnostics 33
4.2 Spectral ViT V1 Confidence 34
4.3 Spectral ViT V1 Diagnostics: 99% Cut 34
4.4 Spectral ViT V1 Diagnostics: 99.9999% Cut 35
4.5 Spectral ViT V2 Diagnostics 35
4.6 Spectral ViT V2 Confidence 36
4.7 Spectral ViT V2 Diagnostics: 99% Cut 36
4.8 Spectral ViT V2 Binary Diagnostics 37
4.9 Spectral ViT V2 Progenitor Classifier Diagnostics 38
4.10 Spectral ViT V2 Type Classifier Diagnostics 39

A.1 CNN architecture. Figure from Sepeku (2022). 47
A.2 Spectral ViT Architecture 48
A.3 Spectral ViT Block 49

1. Introduction

Bright astrophysical transients such as novae and supernovae have been observed with naked-eye observations for millennia, but only in the past 130 years, with the advent of astrophysical imaging and spectroscopy and the advance of nuclear physics, have we been able to produce a taxonomy of transients. For example, in 1885 and 1895, two intensely bright transients were observed for the first time through telescopes, but astronomers were unable to definitively identify their origin (de Vaucouleurs and Corwin 1985; Schaefer 1995). In the 1930s, these objects were eventually named 'supernovae' (SNe) by Walter Baade and Fritz Zwicky, who were the first to suggest that these events were the cataclysmic death of stars, and predicted the remaining object to be a cooling neutron star (Baade and Zwicky 1934). Since then, astronomers have defined SNe to be the result of either the core-collapse of stars with a mass $\gtrsim 8M_{\odot}$, or a thermonuclear explosion in binary star systems involving a white dwarf sharing a stellar envelope with a larger star (Filippenko 1997).

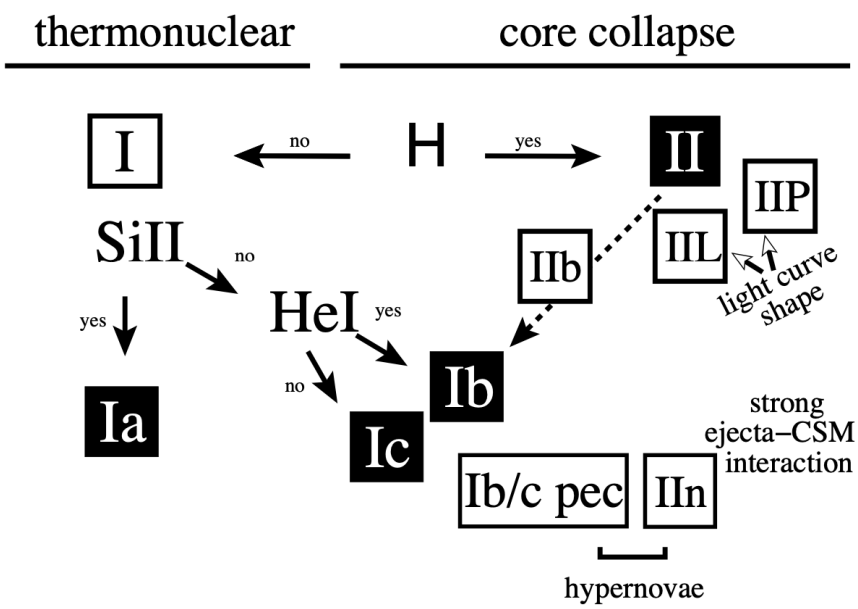


Figure 1.1: Supernova classification. The classification scheme is based on the presence or absence of hydrogen and silicon features in the spectra of SNe. Figure from Turatto (2003).

1.1 Supernova Classification

SNe have traditionally been classified based on their spectra starting with Minkowski (1941), who first classified SNe into two groups based on the presence or absence of hydrogen lines. This evolved into the current classification scheme, which is based on the presence or absence of hydrogen and silicon in the spectra of SNe (Figure 1.1).

SNe with hydrogen in their spectra are classified as Type II SNe, while those without hydrogen are classified as Type I. Type I SNe are further subdivided into Type Ia, Ib, and Ic (Turatto 2003). The presence of SiII is a distinguishing component in the Type Ia classification, with Ib and Ic differentiated by HeI.

Although divided into three subtypes, Type I SNe are caused by two different

mechanisms. Type Ia SNe are caused by the thermonuclear explosion of a white dwarf star, while Type Ib and Ic SNe are caused by the core-collapse of a massive star (Filippenko 1997). Thus Type Ib and Ic SNe are actually more related to Type II SNe, and are collectively referred to as core-collapse SNe. We now know the physical origin of the spectral differences between the subtypes: stars stripped of their hydrogen envelopes appear as Type Ib SNe, while those stripped of both their hydrogen and helium envelopes are classified as Type Ic SNe.

SNe are important for studying the end stages of massive stars, and because some subtypes have standardizable magnitudes (specifically SNe Ia and II-P) they are used in the cosmological distance ladder. Therefore, merely detecting a supernova is not enough, as astronomers must identify its type to study the progenitor and, if possible, compute its distance modulus. Generally, a supernova is observed by several means, with discovery in photometric observations and classification using follow-up spectroscopy (e.g. Perlmutter et al. (1995) and references therein). This process is time-consuming, requiring not only the initial discovery of the supernova but also follow-up observations to determine the type before it fades away. Due to the large volume of new discoveries and the relatively small number of single and multi-object spectrographs suitable for follow-up, only about 10% of discovered SNe receive spectroscopic classification (*Transient Name Server* 2015).

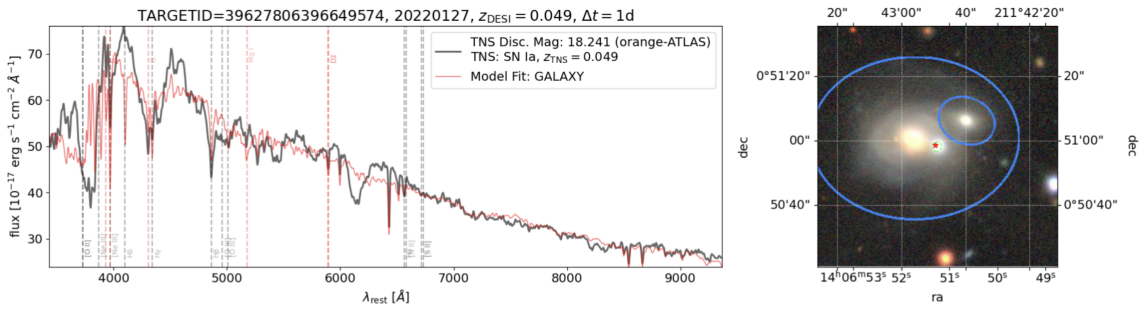


Figure 1.2: DESI host galaxy spectrum contaminated by a Type Ia supernova (January 2022). The right panel shows the image of the host galaxy, with the red star indicating the position of the DESI fiber. Independent spectroscopic follow-up reported to the TNS public database confirmed this is an SN Ia.

1.2 Dark Energy Spectroscopic Instrument

The Dark Energy Spectroscopic Instrument (DESI) is a spectroscopic system located at Kitt Peak National Observatory in Arizona. Installed at the Mayall 4-m telescope, DESI is fitted with 5,000 fiber positioners and ten 3-arm spectrographs capable of resolving the [O II] $\lambda\lambda 3726, 3729$ doublet used for redshift fits of galaxies (Guy et al. 2023).

The DESI survey focuses on four classes of targets: 14 million bright galaxies[1] in a Bright Galaxy Survey (BGS) at redshifts $z < 0.5$; 8 million luminous red galaxies (LRGs) at redshifts $0.4 < z < 1.2$; 17 million emission line galaxies (ELGs) at $0.6 < z < 1.6$; and 3 million quasi-stellar objects (QSOs) between $0.9 < z < 4$. In this work, we will focus on the Bright Galaxy Survey, which is observing host galaxies with a median redshift of $z \approx 0.2$ (DESI Collaboration et al. 2016; Hahn et al. 2022).

DESI began its main survey in May 2021. Over the five-year span of observations, DESI will inevitably observe host galaxies with active SNe, leading to contaminated

[1]The BGS contains galaxies with r-band magnitudes above 19.5.

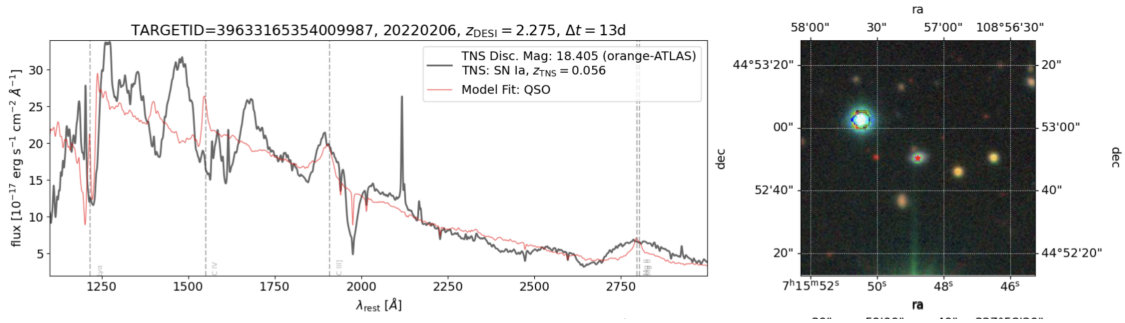


Figure 1.3: DESI host galaxy spectrum contaminated by a Type Ia supernova (February 2022). The right panel shows the image of the host galaxy, with the red star indicating the position of the DESI fiber. Independent spectroscopic follow-up reported to the TNS public database confirmed this is an SN Ia. Note the extremely poor redshift fit provided by the DESI pipeline.

spectra. The BGS may contain as many as $\sim 10^5$ serendipitously observed supernova (DESI Collaboration et al. 2016). An example of a supernova observed in a BGS host by DESI and with an independent spectroscopic classification is shown in Figure 1.2.

The presence of transients in DESI spectra is a contaminant for the survey's dark energy program, since the transient can cause a catastrophic failure in the redshift fit, as shown in Figure 1.3. Thus it is useful to identify these contaminants whenever possible. Moreover, the serendipitous observation of supernovae is inherently useful since the spectroscopy provides an opportunity to classify *most* types of supernovae without requiring additional follow-up. Both of these factors motivate this study.

1.3 Past Approaches to Spectroscopic Identification

The idea to search for active transients in spectroscopic data sets is not new; see, for example, Madgwick et al. (2003), Graur and Maoz (2013), Baron and Poznanski

(2017), Muthukrishna, Parkinson, and Tucker (2019), and Davison, Parkinson, and Tucker (2022). Until the past 5 years, most spectroscopic supernova identification algorithms would compare observations against supernova spectral templates; more recent searches have applied machine learning techniques. In nearly all cases, the searches proceed by characterizing and subtracting the galactic continuum of the host galaxies and then fitting the template to the residual spectra or running a machine-learning algorithm on the residual spectra.

The use of templates and residual spectra, while robust and easy to understand, leads to several computational inefficiencies. First, the host galaxy continuum must be fit accurately in the presence of a possibly dominant transient. An incorrect fit, regardless of the presence of a supernova, can lead to misclassification. Second, even if it is assumed that the residual spectrum contains only a supernova, a template search must include all of the subtypes of SNe adjusted for all possible redshifts appropriate for the targets. In addition, the classification of SNe is also time-dependent, so the evolution of each subtype must also be included in each template. An example SN Ia model showing the spectral evolution is provided in Figure 1.4.

1.4 Supernova Classification with Machine Learning

Machine learning (ML) has been applied broadly to different fields for its flexibility and ability to find patterns in data. The emergence of Deep Learning (DL) has allowed ML to be applied to tasks such as image classification (Krizhevsky, Sutskever, and Hinton 2012), speech recognition (Nassif et al. 2019), and natural language

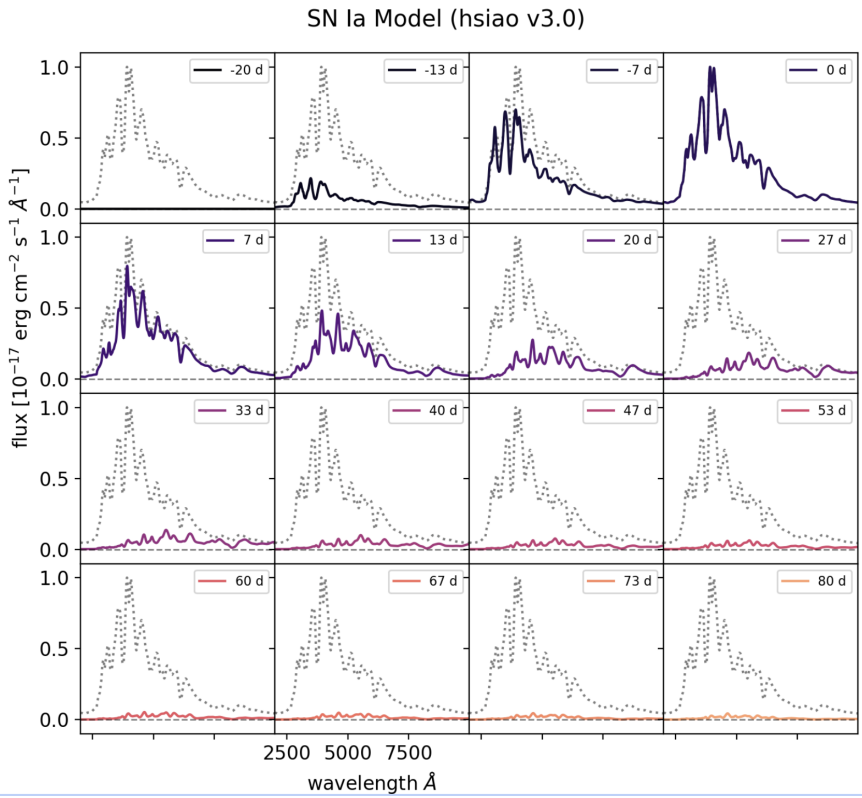


Figure 1.4: Spectral model of a SN Ia shown over the course of its evolution, from 20 days before peak brightness to 80 days after. Adapted from BenZvi, Tiwari, and Wasserman (2020).

processing (Mikolov et al. 2013). ML and DL have recently been applied to astronomy, with promising results. Gauci, Zarb Adami, and Abela (2010) used a random forest algorithm to distinguish between spiral, elliptical, and irregular galaxies, and Becker et al. (2021) used a convolutional neural network (CNN) to classify the morphology of radio galaxies. CNNs found success in working within the DESI collaboration, with Parks et al. (2018) using the architecture to detect strong emission lines. These techniques have also been applied to the classification of SNe. Möller et al. (2016) used a CNN to classify SNe into Type Ia and non-Ia SNe, and Muthukrishna, Parkinson,

and Tucker (2019) and Davison, Parkinson, and Tucker (2022) further extended the classification to identify core-collapse subtypes.

Previous attempts at using ML algorithms for transient detection specifically on DESI spectra have been conducted by Rubenzahl (2018), Wasserman (2021), and Sepeku (2022). The classifiers presented by Wasserman (2021) and Sepeku (2022) were shown to be capable of high precision (further explained in Section 2.2.1), but only after pipeline-dependent preprocessing and extensive cutting based on model certainty. An alternative model based on the vision transformer (ViT) architecture is proposed here to solve these shortcomings. Aptly named the "Spectral ViT," this model seeks to classify SNe into their respective subtypes with comparable precision and higher recall compared to previous classifiers, without dependence on the DESI redshift fit for pre-processing.

In Chapter 2, the deep learning techniques used throughout the study are explained in detail, while Chapter 3 describes the creation and training of the Spectral ViT. The results of the study are discussed in Chapter 4, with future research and applications of this model presented in Chapter 5.

2. Deep Learning Techniques

2.1 Multi-Layer Perceptrons

To aid in the classification of SNe, DL techniques can provide flexibility and scalability that have been previously unattainable using template-based methods. The simplest and most widely used architecture in DL are feed-forward fully connected neural networks, also known as multi-layer perceptrons (MLPs) (Popescu et al. 2009). These networks are composed of a series of "layers" of neurons, or nodes, that are "fully connected" to the previous layer (Figure 2.1). Given an input vector $\vec{x}^{(0)}$ with n values, a MLP applies a series of weights to each element, resulting in a linear transformation from $\vec{x}^{(0)}$ to $\vec{x}^{(1)}$, where $\vec{x}^{(1)}$ is a vector of length m. This transformation is then followed by a bias added to each element, followed by the application of an activation function. The transition from layer i to $i+1$ can be represented mathematically as

$$\vec{x}^{(i+1)} = f(\mathbf{W}\vec{x}^{(i)} + \vec{b}^{(i+1)}), \tag{2.1}$$

where $\mathbf{W}$ is an $m \times n$ matrix of weights, $\vec{b}^{(i+1)}$ is a vector of length m representing the bias, and f is the activation function. The activation function can be any function, but

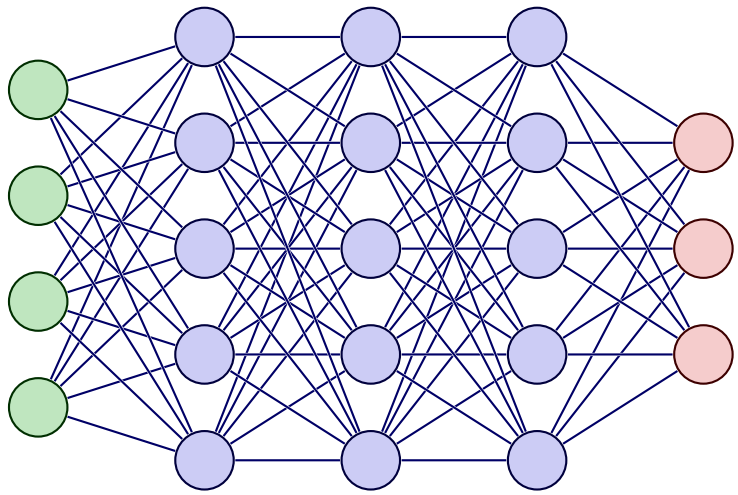


Figure 2.1: A simple MLP with an input dimension of 4 (green nodes), three hidden layers (blue nodes), and an output dimension of 3 (red nodes). Each node is connected to every node in the previous and next layer. The lines represent the weights, and each node has an associated bias. Figure adapted from Neutelings (2021).

traditionally it is chosen to be a function that is continuous, non-linear, monotonically increasing, and differentiable. The most common activation functions are the sigmoid function $f(x_i) = \sigma(x_i) = 1/(1+e^{-x_i})$, the hyperbolic tangen, $f(x_i) = \tanh(x_i)$, and the rectified linear unit $f(x_i) = \mathrm{ReLU}(x_i) \equiv \max(0, x_i)$, all of which are applied element-wise. The output layer of the network is a vector tailored for the particular purpose of the MLP. For example, if the network is used to classify input data $\vec{x}$ into a set of classes, the output is a vector of normalized probabilities of each class.

The MLP architecture, though simplistic, is very powerful. Cybenko (1989) showed that for any boolean function (e.g., 2-class classification), an MLP with a single hidden layer can approximate any function arbitrarily well[1]. This is the basis for the success of MLPs in many fields, but it is also the reason that MLPs are not

[1]This is known as the Universal Approximation Theorem.

the best choice for all problems. MLPs begin to show deficiencies when training large models, such as the exploding/vanishing gradient problem, which makes them difficult to train (Neilsen 2015). In addition, the lack of invariance under translation of data (important for many types of data, such as images), and the exponential growth of the number of trainable parameters in a fully connected network (Naskath, Sivakamasundari, and Begum 2022) pose serious problems which inhibit the use of MLPs on larger datasets and in more complex problems, including image recognition, time series analysis, and natural language processing. Since the mid-2000s, the weaknesses of MLPs have been addressed by the adoption of alternative network architectures such as convolutional neural networks (CNNs) and transformers.

2.2 Convolutional Neural Networks

Convolutional neural networks (CNNs) were originally developed to address the lack of invariance under translation in input data (Fukushima 1979). CNNs designed for classification are traditionally composed of a series of convolutional layers, usually followed by a pooling layer, and a fully connected layer. The convolutional layers are composed of a series of filters that take a certain number of inputs from the previous layer and apply a kernel to them, resulting in a single output. These filters are applied across the entire input space, and are commonly referred to as feature maps. If N filters are applied in each convolutional layer, the result is a tensor of shape $N \times x$, where x is the shape of the original input (which can be 1D or multi-dimensional). To reduce the dimensionality of the output, a pooling layer is applied. The pooling

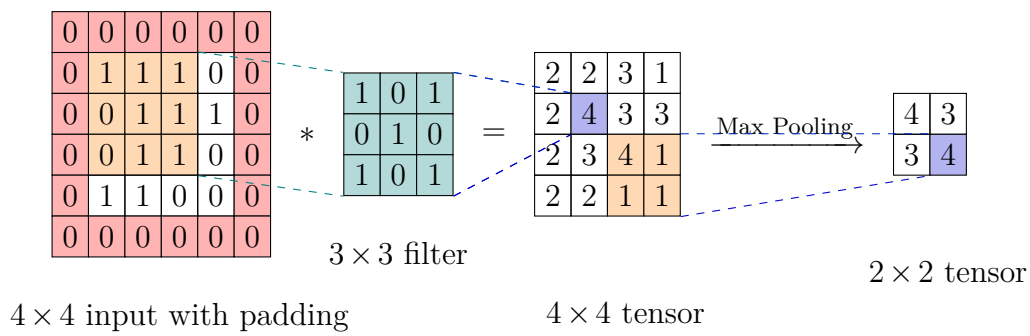


Figure 2.2: A convolutional filter and a 2×2 pooling applied to a 2D input. As the filter is passed over the input space, the kernel is applied. The input is padded with zeros to preserve dimensionality, which is then reduced by the max pooling. Figure adapted from Neutelings (2022).

layer takes a certain number of inputs from the previous layer, and reduces them to a single output, similar to the convolutional filters, but without the kernel application. The most common pooling layer is the max-pooling layer, which takes the maximum value in each set as the reduced value. The process of passing an input through a convolutional filter and pooling layer is shown in Figure 2.2. These convolution and pooling layers are applied in series until the desired output dimensionality is reached.

From here, depending on the task, the output of the pooling layer can be fed into a fully connected layer, similar to the MLP architecture described in Section 2.1, or it can be fed into another convolutional layer. An MLP is usually applied for classification tasks, while another convolutional layer is applied for segmentation tasks. An example of a CNN architecture used for classification is shown in Figure 2.3.

The CNN architecture was shown to be effective in image recognition tasks by LeCun, F. Huang, and Bottou (2004) and then popularized after the success

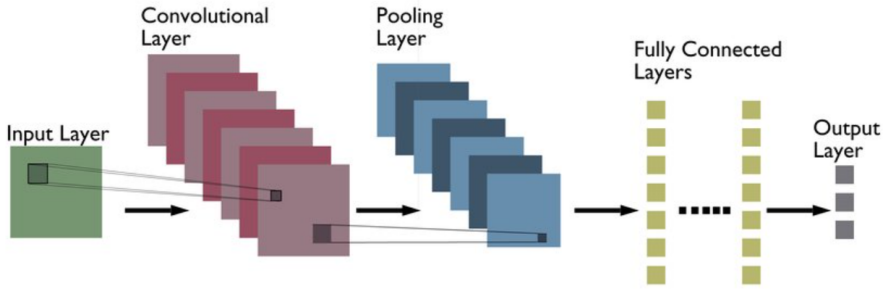


Figure 2.3: A convolutional network archeticture designed for classification (adapted from Kumar 2022). Only one convolutional/pooling layer is shown, but there can be multiple depending on the application.

of AlexNet in the ImageNet challenge (Krizhevsky, Sutskever, and Hinton 2012). AlexNet was the first of many CNN architectures popularized in the 2010s for vision classification, such as VGG (Simonyan and Zisserman 2015), ResNet (He et al. 2015), and DenseNet (G. Huang et al. 2016). This also spurred a sudden increase in deep networks designed for segmentation tasks as well, such as the FCN architecture by Shelhamer, Long, and Darrell (2016). In addition to vision tasks, it also found great success in applications such as natural language processing (Kim 2014), engineering, and medicine (Kiranyaz et al. 2021).

2.2.1 Using CNNs on Spectroscopic Data in Astronomy

CNNs excel at identifying patterns throughout their input space, stemming from their spacial invariance. Theoretically, this should translate to spectral classification quite well. Currently running at the National Energy Research Scientific Computing Center (NERSC) is a 1D CNN developed by Wasserman (2021) for DESI transient detection. This model, while proven to be effective on training data, is limited by

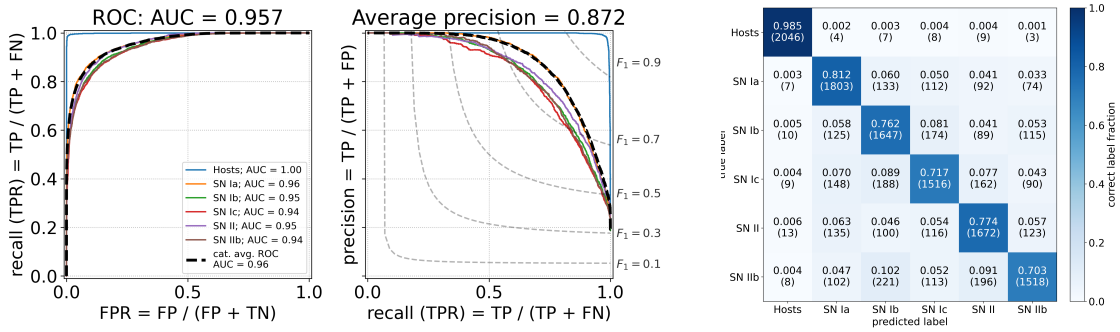


Figure 2.4: CNN Diagnostics: ROC Curve (left) and Confusion Matrix (right)

the number of false positives it produces. In order to perform well on DESI data, Wasserman (2021) created a preprocessing pipeline that converts the spectra into a format that is more easily interpretable by the CNN. This preprocessing pipeline includes de-redshifting the spectra into the observed galaxies' rest frames, applying a log-binning to the spectra, normalizing the flux to values between 0 and 1, and padding the spectra to a fixed length.

The work by Wasserman (2021) provided a foundation for the use of CNNs on SNe classification which was continued by Sepeku (2022), who introduced the use of a 2D CNN for SNe classification, adding a second dimension to the spectra by re-shaping the input from an $N \times 1$ array to an array of size $N/2 \times N/2$. The resulting 2D 'image' was then fed to a more traditional vision CNN architecture (see Appendix A.1 for a complete diagram of the architecture). This architecture was shown to train quickly (less than 1 hour on a single GPU), and was able to classify a validation set of 12888 spectra with an average precision of 0.872 in a 6-label categorical classification of host galaxies and supernova sub-types.

Figure 2.4 shows the ROC curve and confusion matrix for the 2D CNN trained

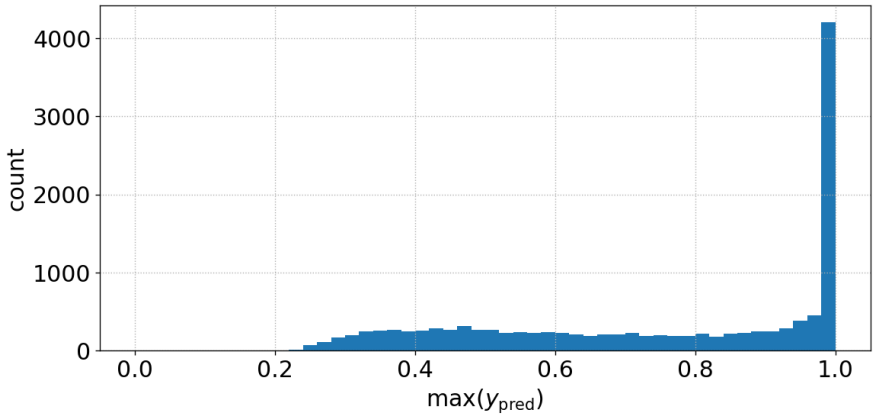


Figure 2.5: Max value of the output vector from the CNN.

on synthetic spectra.

The output of the CNN is a 6D "prediction" vector $\vec{y}_{\rm pred} = (y_0, y_1, y_2, y_3, y_4, y_5)^{\mathsf{T}}$, where $\sum_i y_i = 1$. Each element corresponds to the probability that the input spectrum belongs to a host galaxy or includes one of five supernova sub-types. As shown in Figure 2.5, the maximum value of the output vector varies between 1, indicating a perfectly confident classification, and 0.2, indicating a high degree of confusion

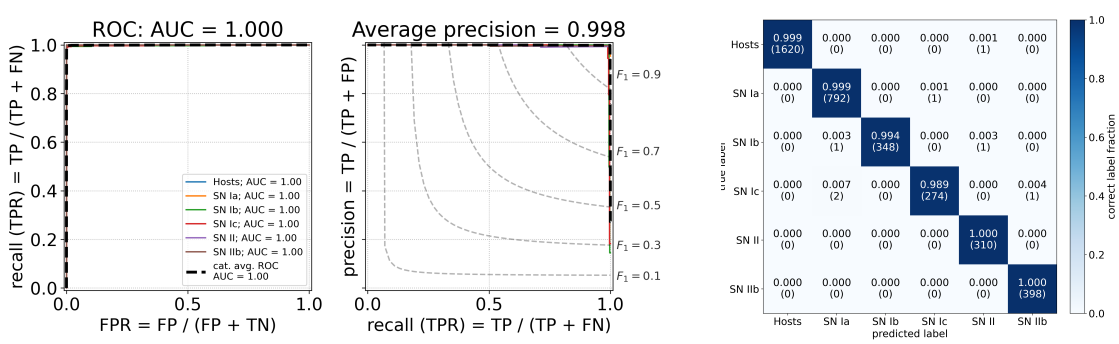


Figure 2.6: CNN Diagnostics: ROC Curve (left) and Confusion Matrix (right) with a 99% confidence cut

among the host and supernova spectra. Figure 2.5 shows the distribution of the maximum value found for an independent validation sample of spectra. To ensure that the CNN is not misclassifying spectra with low confidence, a cut on the maximum value of the output vector can be applied. Figure 2.6 shows the ROC curve and confusion matrix for the CNN trained on synthetic spectra with a 99% confidence cut $\max\{\vec{y}_{\text{pred}}\} > 0.99$, which preserves 28.8% of the original validation set.

A nearly perfect ROC curve and confusion matrix are produced by the CNN when evaluated on only highly confident classifications. This indicates that the CNN is able to remedy the issue of false positives by only classifying spectra that it is confident in. However, this cut is not ideal, with only 28.8% of the data, sacrificing efficiency for purity.

The next step in training would be to either further increase the accuracy of the CNN or move to a different architecture, with the hopes of increasing not only the overall accuracy of the networks, but also the number of confident classifications.

2.3 From CNNs to Transformers

Transformers are a relatively new DL architecture introduced in 2017 by Vaswani et al. (2017) for natural language processing (NLP) tasks. The original architecture consists of an encoder-decoder system (Figure 2.7). The encoder accepts a series of tokens and produces a series of vectors representing the input data via a series of self-attention layers and feed-forward layers. The decoder then takes the output of the encoder, and produces a series of output tokens, one for each input token.

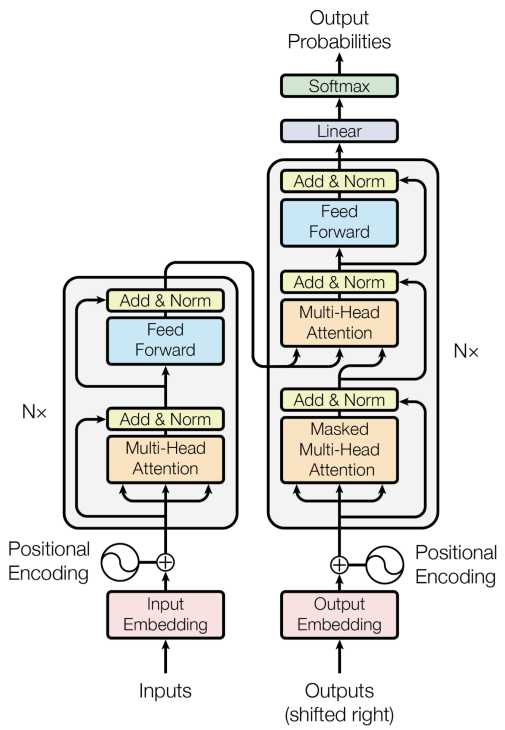


Figure 2.7: Traditional transformer architecture consisting of an enocoder and decoder. Image from Vaswani et al. (2017)

Once transformers were shown to have remarkable success in NLP tasks, they were quickly adapted to other fields, such as vision and image classification. Dosovitskiy et al. (2020) developed a vision transformer (ViT) architecture that differed from the original transformer encoder by replacing the tokenized input with a more image-appropriate alternative. In short, the input $n \times n$ image is broken into a series of $m \times m$ patches where $m < n$ and $m|n$, which are then flattened into a vector. These vectors, along with positional encodings, are then fed into the transformer architecture, as shown in Figure 2.8. For classification tasks, the first input token is replaced with a class token. After passing through the transformer, the class token is then

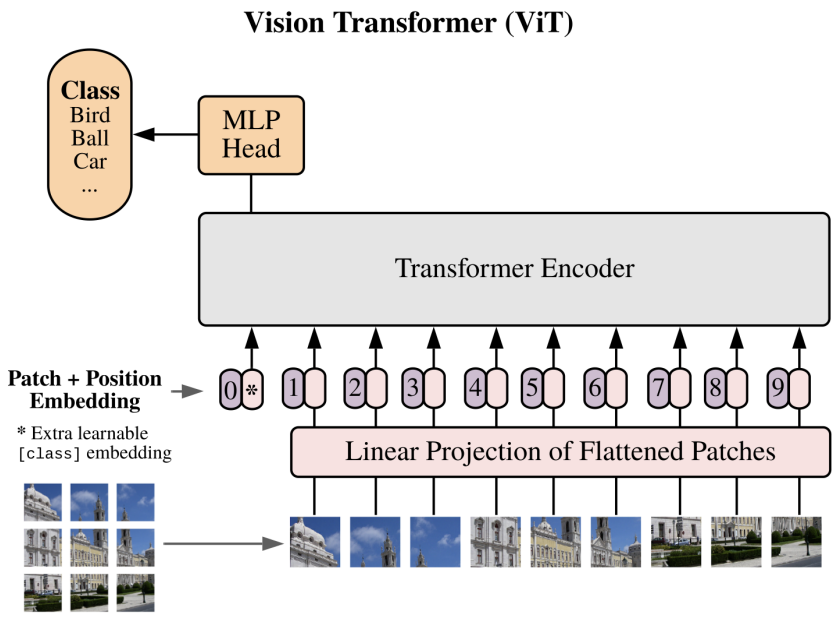


Figure 2.8: Vision transformer (ViT) architecture. Purple ovals indicate positional embeddings, while pink ovals indicate the resulting tokens from the patches. Image from Dosovitskiy et al. (2020).

run through a fully connected layer to produce the final categorization probabilities. Further details are provided in Section 3.1.

2.3.1 ViT on Spectroscopic Data

Previous implementations of transformers have been shown to have characteristics beneficial to the classification of spectral data. A transformer's ability to learn contextual information is essential in spectral classification. A broad absorption line, for example, may be indicative of a Type Ia supernova if in one part of the spectrum but may be indicative of a Type II supernova if in another part of the spectrum. This

contextual information is not easily learned by a CNN, as the convolutional layers are not able to learn the importance of certain parts of the input space. Attention can also play a role in identifying the purpose of features that are not in a standard location. For example, a spectrum in the observer frame might have a continuum pattern at different locations along the spectrum than in the object's rest frame, but the overall shape would be similar (minus features redshifted out of the observer's wavelength range). This change in sizing would be difficult to learn with fixed filters in a CNN, but would be identified based on their positional importance by a transformer. In addition to this, ViTs have been shown to outperform CNNs on vision tasks, which shows they are capable of focusing on learned features.

3. Creation and Training of the Spectral ViT

In order to examine the effectiveness of the transformer architecture for spectral classification, the following series of steps were taken. First, a custom transformer architecture was created based on the traditional ViT architecture (Dosovitskiy et al. 2020), and trained on a simple synthetic dataset to validate our approach. Next, a synthetic dataset using DESI spectra was created under a variety of preprocessing conditions. The previously created CNN and new transformer architectures were trained on the same preprocessed data. Finally, these training samples were used to determine the optimal training conditions for non-redshift corrected data.

3.1 Creation of Spectral ViT

The Spectral ViT architecture was coded using the `PyTorch` deep learning framework (Paszke et al. 2019). The complete architecture is shown graphically in Appendix A.2. The Spectral ViT is composed of three main components: the preprocessor, the encoder, and the classifier.

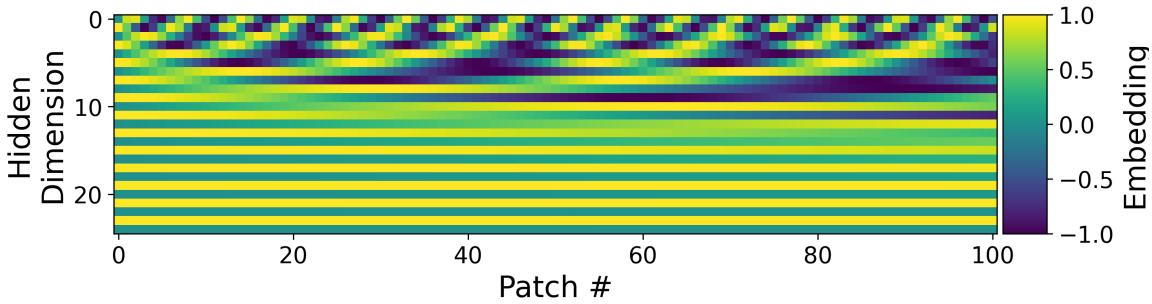


Figure 3.1: Visual representation of the embeddings for a sample of spectra. The x-axis represents the index of the patch, the y-axis represents the index of the hidden dimension, and the color represents the value of the embedding.

The pre-processing component is responsible for taking the input spectra extracted from the DESI CCDs and converting them into a series of vectors that the transformer can interpret. Consider a sample of N spectra, each with 10000 pixels corresponding to a bin containing a wavelength between 3600Å and 9800Å. Each group is split into a set number of patches (approximately 100 pixels in length). Each patch is then linearly mapped via a fully connected network to a vector 4 times smaller than the patch size. The sample is now a set of size $N \times 100 \times 25$. A classification token of the same dimension as the patches is initialized with uniformly distributed random values between 0 and 1, and then concatenated to the beginning of each sample.

In order for the transformer to properly understand the positional relationship between each patch, an embedding is added to each patch. This embedding is a scalar function based on the size of the patch and is calculated as follows:

$$\text{Embedding}_{ij} = \begin{cases} \sin\left(\frac{i}{10000^{(j/\text{patch size})}}\right) & \text{if } j \text{ is even} \\ \cos\left(\frac{i}{10000^{((j-1)/\text{patch size})}}\right) & \text{if } j \text{ is odd} \end{cases}, \tag{3.1}$$

where i is the patch index, j is the index in the hidden dimension, and the patch size is the number of pixels defined in each patch (Vaswani et al. 2017). In a CNN, the positional relationship between each pixel is understood by the convolutional layers and the pooling layers. However, in a transformer without the positional embeddings, there is no difference between the first and last patch in the sample. The sinusoidal embeddings we use are based on the original transformer architecture (Vaswani et al. 2017), but also provide a simple way to encode the positional information. A visual representation of the embeddings for the sample is shown in Fig. 3.1. Once the embeddings are added element-wise to the patches, the resulting tensor of size $N \times 101 \times 300$ is passed through the encoder.

The encoder is the main component of any ViT architecture, as it contains the multi-head attention and feed-forward layers. The encoder is composed of a user-defined number of transformer blocks, each of which contains layer normalization, multi-head attention, another layer normalization, and finally a feed-forward layer. The multi-head attention layer used in this work is a scaled dot-product attention layer found in Vaswani et al. (2017), with the adjustments suggested by Xiong et al. (2020) for efficient training. Each patch is passed through three separate linear layers, each with a different set of weights, resulting in three sets of vectors: the query (Q), key (K), and value (V).

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{Q \cdot K^T}{\sqrt{d_k}}\right) \cdot V \tag{3.2}$$

The dot product between the query and key vectors is then calculated, scaled by the square root of the dimensionality of the query vector, and then passed through a

softmax function,

$$\text{softmax}(x_i) = \frac{e^{x_i}}{\sum_{j=1}^{N} e^{x_j}}. \tag{3.3}$$

The resulting attention weights are then multiplied by the value vectors (Equation 3.2). This is simultaneously done for each head in the multi-head attention layer. Each result is concatenated together and then passed through a linear layer to reduce the dimensionality to the size of the query vector. This resulting vector is then added

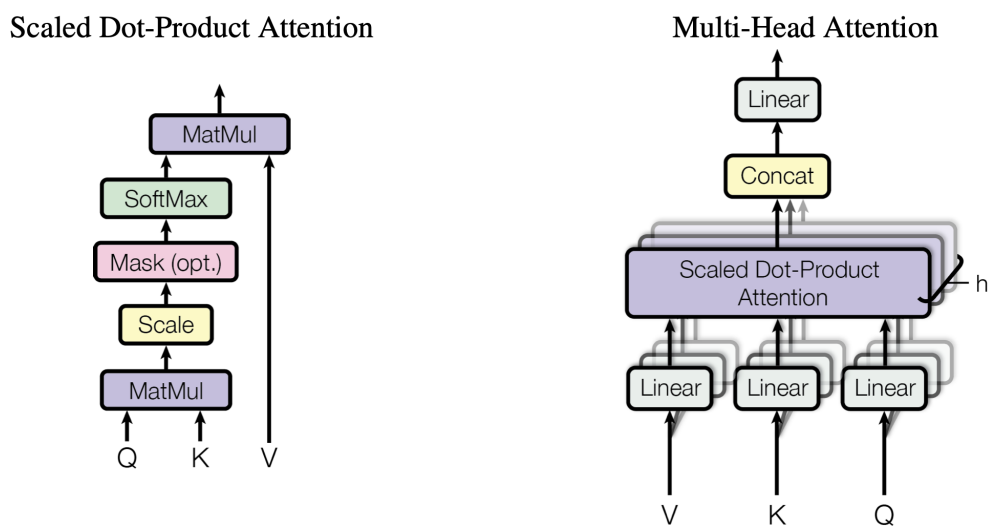


Figure 3.2: Graphical Depiction of Multi-Head Attention Layers inside a transformer architecture. Image from Vaswani et al. (2017)

element-wise to the original patches, normalized, and then passed through a MLP, resulting in a tensor of size $N \times 101 \times 300$. This again is added element-wise to the original patches. A visual example of this is shown in Figure 3.2. Each step is repeated for a user-defined number of transformer blocks, resulting in a tensor of size $N \times 101 \times 300$. A diagram of the ViT block used the Spectral ViT is shown in Figure A.3.

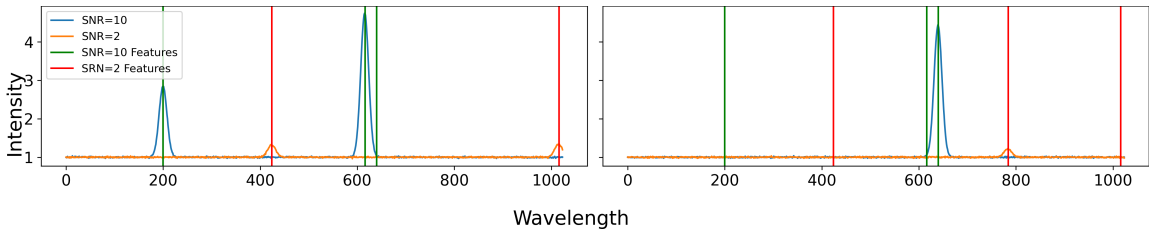


Figure 3.3: Synthetic spectra used to verify the performance of Spectral ViT. Red and green vertical lines represent features used to separate the data into distinct classes for SNR=2 and SNR=10, respectively. Two classes are shown for each SNR.

The final component of the Spectral ViT architecture is the classifier. The classifier takes in only the first patch of each sample, which has been designated as the classification token. The classification token is passed through a linear layer to reduce the dimensionality to the number of classes, and then passed through a softmax function to produce a probability distribution over the classes. Therefore, the resulting tensor is of size $N \times 1 \times 6$, for our 6 classes of galaxy + SNe sub-types.

3.1.1 Validation of Spectral ViT Architecture

After creating the Spectral ViT architecture, we tested our ability to train it effectively. A synthetic dataset, consisting of a continuum with broad Gaussian peaks placed at predetermined locations, was created. Certain combinations of peak locations were chosen to represent an arbitrary 'class' of SNe. These peaks, our synthetic "emission lines," were given random amplitudes and widths to simulate variability based on the maximum 'signal' allowed for a given sample. To ensure the performance of the model, two sample sets were made: one with particularly noisy data (a maximum signal only extending past the Gaussian noise by a factor of 2), and one with very prominent features (a maximum signal extending past the Gaussian noise by a factor

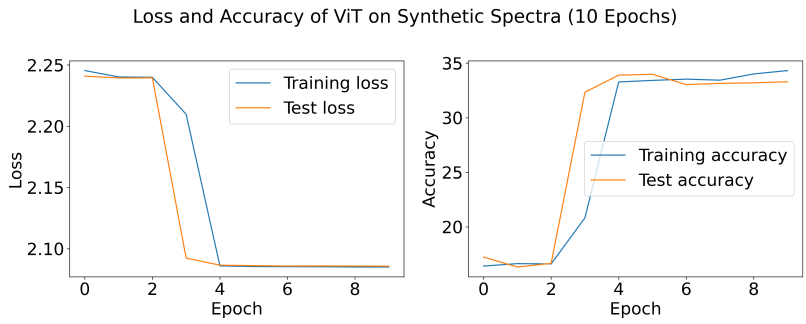
of 10). For convenience, these datasets were labeled to have a signal-to-noise ratio (SNR) of 2 and 10, respectively. Examples of the synthetic spectra with different continuum profiles are shown in Fig. 3.3.

These datasets with SNR=10 were trivially separable by a Spectral ViT architecture with two transformer blocks, two transformer heads, eight hidden layers, and a patch size of 16 (spectra were 1024 pixels long). An accuracy of 100% on the testing set was achieved within 5 epochs of training (Figure 3.4b). Trained on the same architecture, the data with SNR=2 was more difficult to separate, only achieving a 33.975% test accuracy (Figure 3.4a).

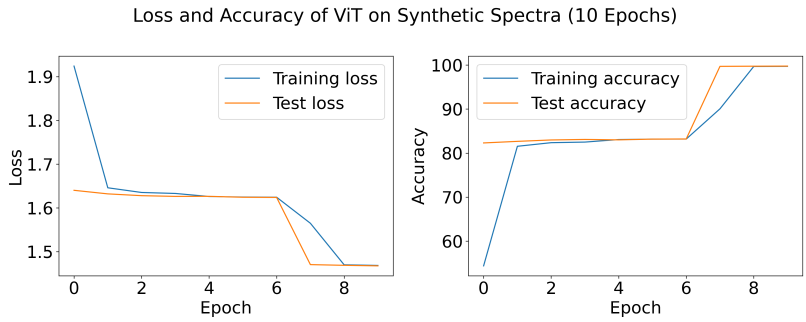
3.2 Creation of Synthetic DESI Spectra

Once we demonstrated the Spectral ViT architecture was training effectively on simple synthetic data, the architecture was trained on DESI data. In order to create a large enough training set, observed DESI spectra were used as a base to create synthetic spectra. The process of generating the spectra is explained in detail by Sepeku (2022), but a brief description is provided below.

Observed DESI BGS spectra were used to construct the training and validation samples. Type Ia supernovae were simulated using the `sncosmo` package (Barbary et al. 2016), and the templates provided by Hsiao et al. (2007). For core-collapse SNe, the spectra were simulated based on models produced by Vincenzi et al. (2019). Measured BGS spectra were randomly drawn from DESI observations, and the Ia and core-collapse templates were simulated in `desisim` using the same observing



(a) SNR = 2



(b) SNR = 10

Figure 3.4: Training of Spectral ViT on synthetic datasets. The model was able to reach 100% accuracy on the test set of SNR=10 (top), but failed to reach an accuracy of 50% on the test set of SNR=2 (bottom).

conditions (seeing, transparency, moon illumination, airmass) as the measured spectra. The simulated transient spectra were also allowed to vary in the r band relative to the "host" galaxy spectrum to simulate a variety of observing conditions, such as the transient being located off the fiber axis (resulting in a faint signal) or close to the fiber axis (resulting in a relatively strong signal). The simulated transients were then coadded to the corresponding "host" spectra and saved to a FITS file in the DESI spectrum format.

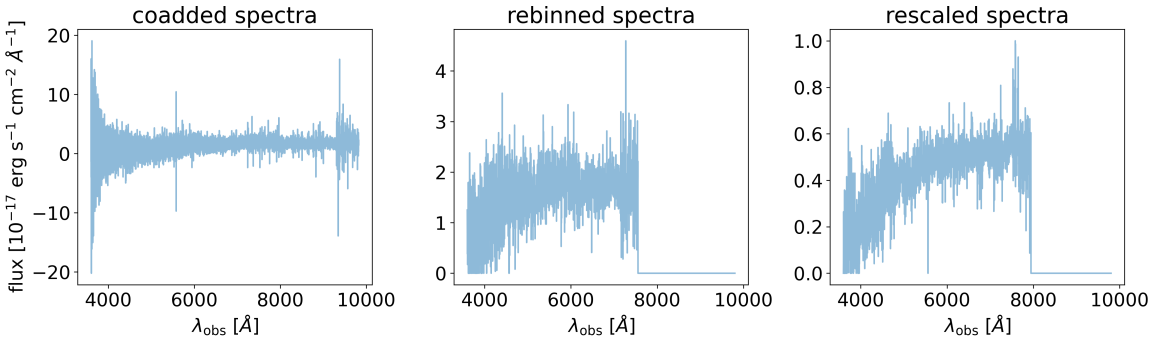


Figure 3.5: DESI synthetic spectra undergoing pre-processing. The different bands are first coadded together (left image). This is then corrected to the galaxies rest frame and separated into 3600 log-scale bins (center image). Finally, the sample is normalized between 0 and 1 (right image)

3.2.1 Pre-Processing of DESI Spectra

Once the spectra were created and saved as DESI files, they needed to be extracted, preprocessed, and split into training, testing, and validation sets. Then they could be saved and used to train the Spectral ViT. The preprocessing method developed by Wasserman (2021) and Sepeku (2022) was used, and includes four main steps:

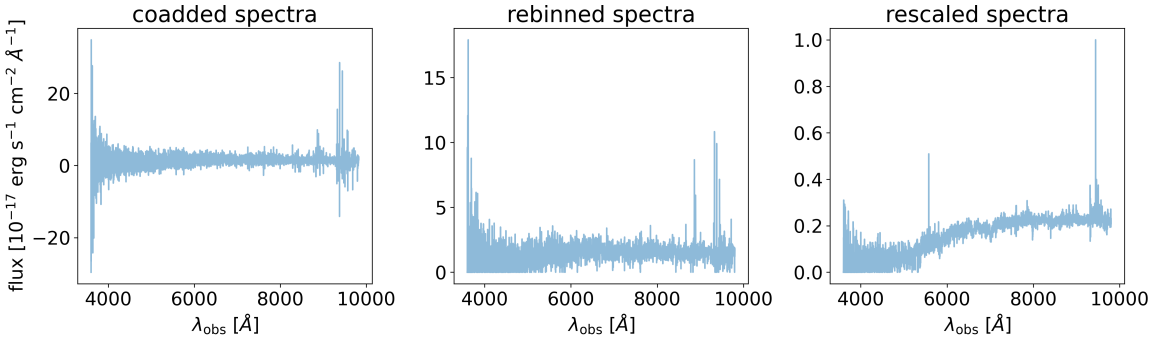


Figure 3.6: DESI synthetic spectra undergoing pre-processing. The different bands are first coadded together (left image). This is then separated into 3600 log-scale bins without any redshift correction (center image). Finally, the sample is normalized between 0 and 1 (right image)

Hyperparameter	Value
Learning Rate	0.00004135238172950965
Regularization	0.06
Dropout Rate	0.40904759925886294
Batch Size	50

Table 3.1: Hyperparameters of the CNN used to classify DESI spectra. CNN adapted from Sepeku (2022).

camera coaddition, redshift correction, rebinning/downsampling, and flux normalization/rescaling.

The reduced spectra, a table of best-fit redshifts from redrock, and a set of redrock templates and coefficients are provided by the DESI pipeline in the form of a FITS file (Pence et al. 2010; Guy et al. 2023). After the spectra between the different cameras are coadded, the best redshift identified by redrock is used to shift the spectra back into the "host" galaxies' rest frames. Following de-redshifting, all artifacts in the spectra (masks, bad pixels, etc.) are removed. The spectra are then downsampled to a user-defined resolution (default is 3600 bins) and normalized to rescale the minimum

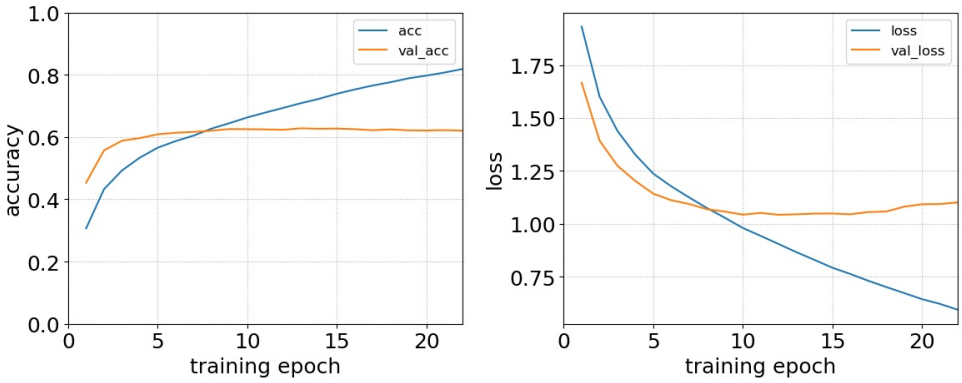


Figure 3.7: Training of the CNN on synthetic DESI spectra.

Hyperparameter	Smaller Architecture	Larger Architecture
Learning Rate	0.0001	0.01
Batch Size	16	16
Blocks	4	4
Heads	5	12
Hidden Dimension	25	72

Table 3.2: Hyperparameters of the smaller and larger Spectral ViT architecture used to classify DESI spectra.

and maximum flux values to 0 and 1. Each of these steps is shown in succession in Figure 3.5 for a redshift-corrected spectrum with 3600 bins, and in Figure 3.6 for a non-redshift corrected spectrum, also with 3600 bins.

3.2.2 Transformer Training

For the CNN datasets, the spectra were split into 2D arrays of equal height and width (i.e. for spectra of length 3600, the arrays would have a shape of 60×60). After this, the spectra were divided into training, testing, and validation samples comprising 60%, 20%, and 20% of the total dataset.

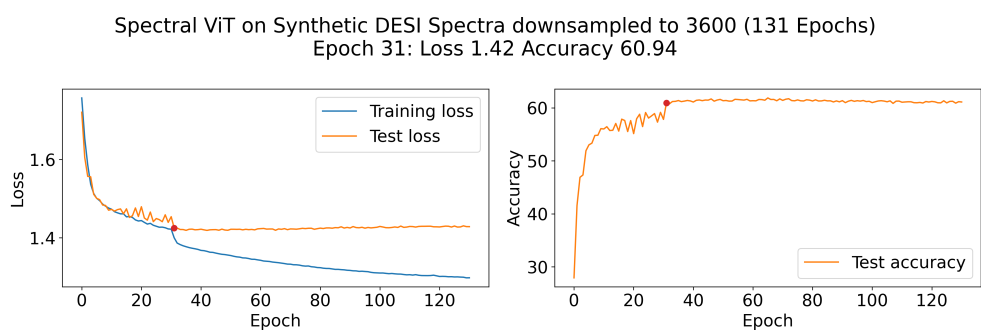


Figure 3.8: Training of the small architecture on redshift-corrected spectra downsampled to 3600 bins. Over-fitting was determined to have occurred by Epoch 31.

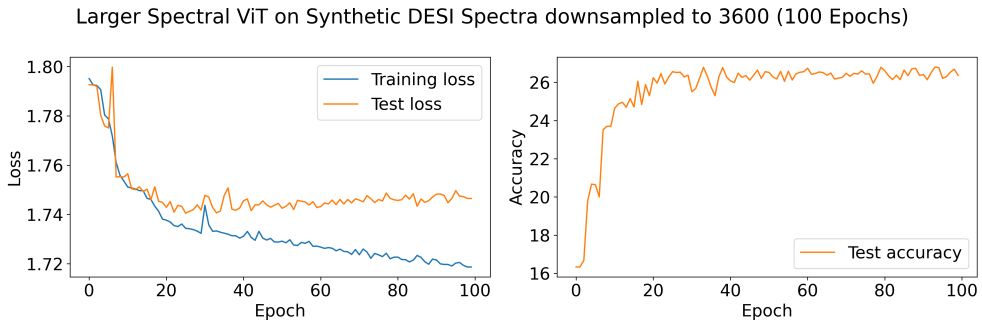


Figure 3.9: Training of the large architecture on Redshift-corrected spectra downsampled to 3600 bins. No convergence above random guessing was observed.

Two Spectral ViT architectures were trained on the same synthetic DESI spectra as the CNN. The first Spectral ViT architecture was smaller than the second. The properties of both are shown in Table 3.2.

3.3 Training of Neural Networks

3.3.1 CNN Training

The CNN architecture was developed by Sepeku (2022), and has properties shown in

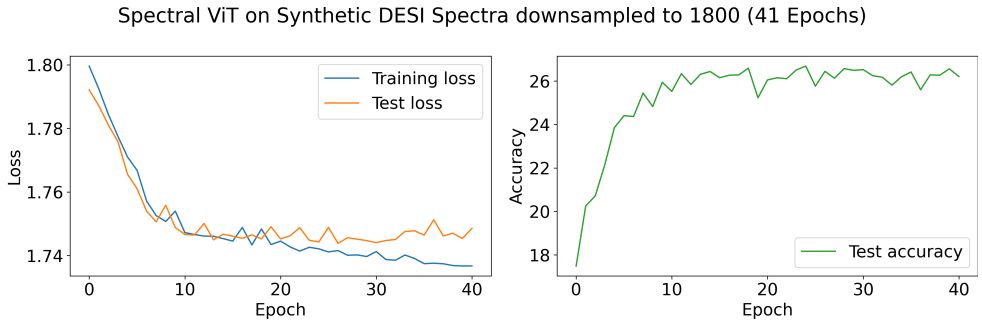


Figure 3.10: Training of the small architecture on Redshift-corrected spectra downsampled to 1800 bins. No convergence above random guessing was observed.

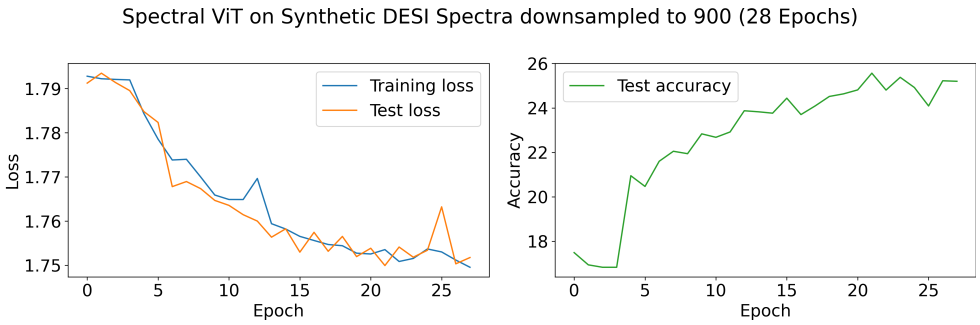


Figure 3.11: Training of the small architecture on Redshift-corrected spectra downsampled to 900 bins. No convergence above random guessing was observed.

Table A.1. This CNN was trained for a maximum of 50 epochs, with a batch size of 50, and non-variable hyperparameters (Table 3.1). The timeline of the training of the CNN architecture is shown in Fig. 3.7.

Both architectures were initially trained on spectra downsampled to 3600 bins. As shown in Figure 3.8, the smaller architecture was able to train effectively, reaching a test accuracy of 60.94% after approximately 31 epochs. The larger architecture did not converge as well, only reaching a test accuracy of around 26% before overfitting, as shown in Figure 3.9. Therefore, the smaller architecture was chosen for further testing.

To explore the effect of the dimensionality of the input data on the performance of the Spectral ViT, the smaller architecture was trained on spectra downsampled to 1800 and 900 bins. As shown in Figure 3.10 and Figure 3.11, neither variation was able to train as effectively as the original binning.

The smaller architecture, using input spectra with 3600 bins, was then trained on the non-redshift-corrected spectra. As shown in Figure 3.12, the model was able to train effectively, reaching a test accuracy of 49.82%. This lack of dependence on DESI

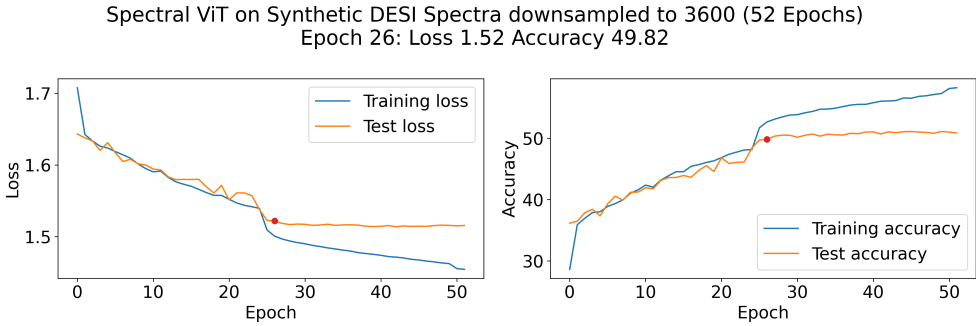


Figure 3.12: Training of the small architecture on non Redshift-corrected data downsampled to 3600 bins. Over fitting was determined to have occurred by Epoch 26.

redshift fitting is promising, as it means that the model could be used to classify spectra without the need for the redshift fit.

4. Spectral ViT Model Performance

4.1 Spectral ViT V1: A suitable replacement

The Spectral ViT's performance was evaluated on a test set of 12,888 validation spectra. These spectra were not used in the training, nor the test set of the model. The resulting ROC curve and confusion matrix are shown in Figure 4.1. The ROC curve shows the model is capable of differentiating between SNe and non-SNe with an AUC of 0.99. However, differentiating between the different types of SNe is more difficult, as the confusion matrix indicates.

In contrast to the CNN previously used by Sepeku (2022), the Spectral ViT

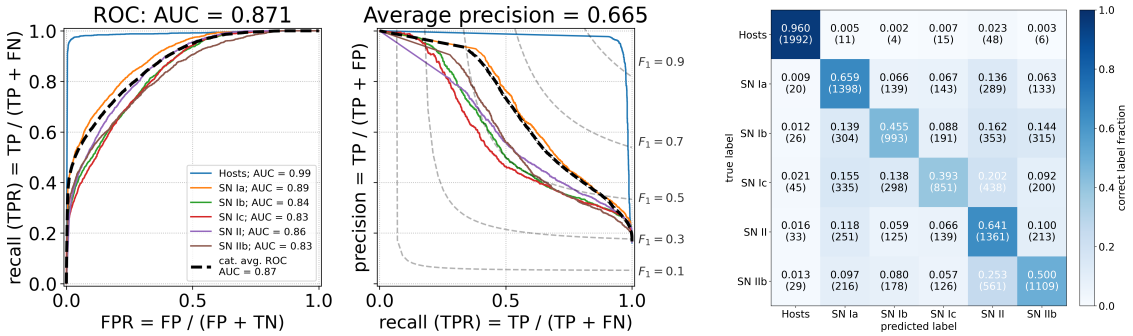


Figure 4.1: Spectral ViT V1 Diagnostics: ROC Curve (left) and Confusion Matrix (right)

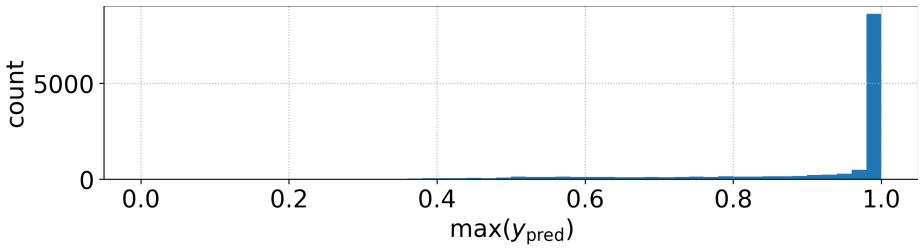


Figure 4.2: Max value of the output vector from the Spectral ViT V1.

V1 was more sure of its predictions, as indicated by Figure 4.2. By filtering out predictions with a maximum value less than 99%, the model's average precision was increased from 0.665 to 0.783 (Figure 4.3). This cut included 63.4% of the original predictions. Further increasing this cut to 99.9999% reduced the sample size to 5156 (40% of the original validation size), but further increased the average precision to 86.2%, as shown in Figure 4.4.

While the accuracy score didn't increase significantly with the second confidence cut, the important metric is the number of positive predictions made. All AUC scores are above 0.95, and the number of supernovae surviving the confidence cut is significantly higher than the previously-used CNN. For example, Figure 4.4 correctly

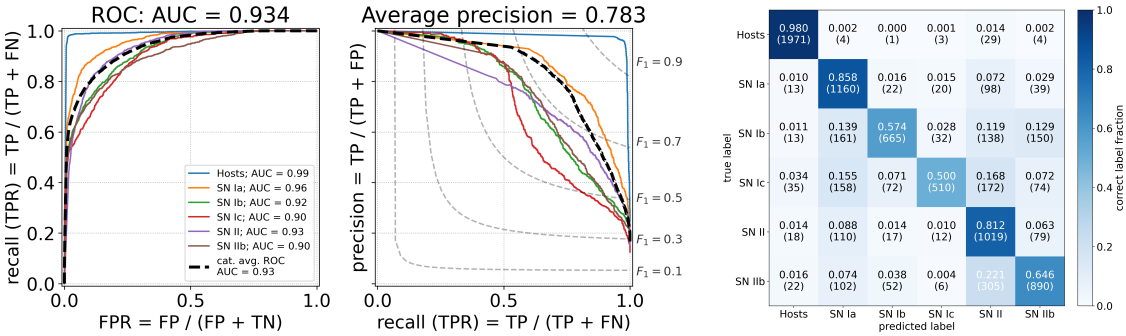


Figure 4.3: Spectral ViT V1 Diagnostics: ROC Curve (left) and Confusion Matrix (right) with a 99% confidence cut

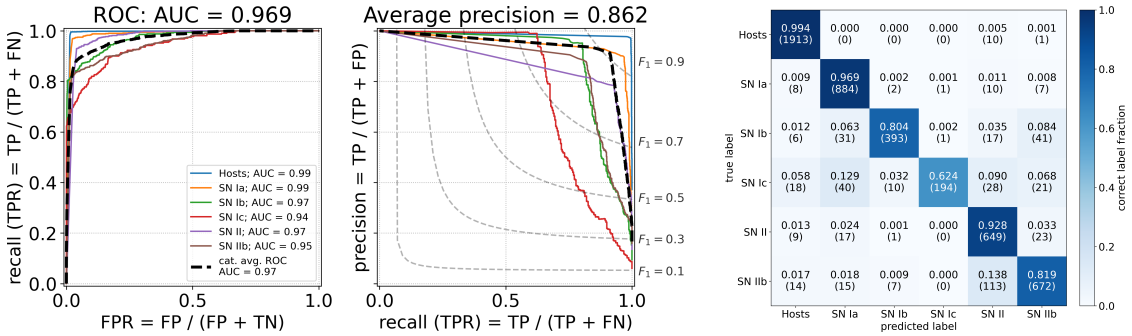


Figure 4.4: Spectral ViT V1 Diagnostics: ROC Curve (left) and Confusion Matrix (right) with a 99.9999% confidence cut

classifies 4705 targets, while Figure 2.6 only correctly classifies 3742 targets under a less strict confidence cut. This represents a 26% increase in the number of correctly classified targets with a higher degree of confidence in the classification.

4.2 Spectral ViT V2: Specialized Classifier

The goal of the Spectral ViT V2 was to see if the dependence on the redshift fit provided by the DESI pipeline could be removed. The validation set was the same size as the V1 validation set: 12,888 spectra. The ROC curve and confusion matrix

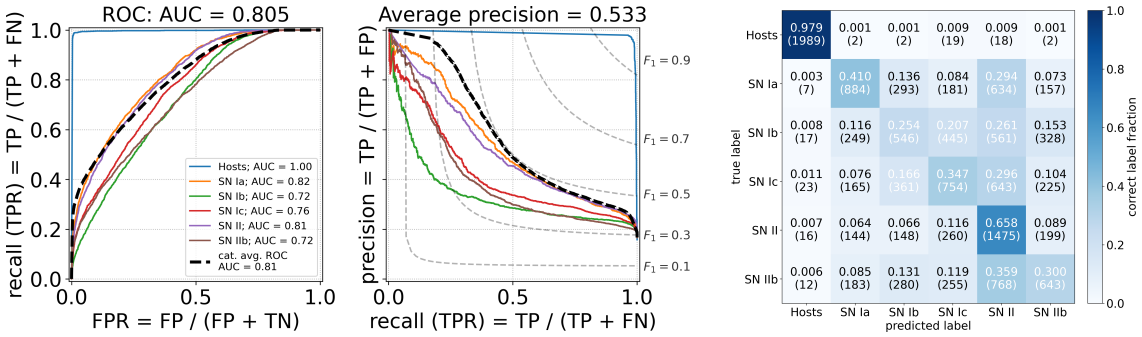


Figure 4.5: Spectral ViT V2 Diagnostics: ROC Curve (left) and Confusion Matrix (right)

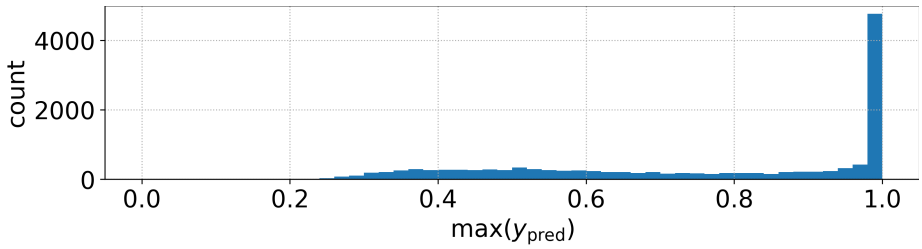


Figure 4.6: Max value of the output vector from the Spectral ViT V2.

are shown in Figure 4.5. Classification of SNe versus non-SNe is still very good, with an AUC of 1.0, but the sub-types are not classified well, resulting in an overall AUC of 0.805 and a lower average precision of 53.3%.

Similar to the CNN, if we apply a 99% confidence cut $\max\{\vec{y}_{\text{pred}}\} > 0.99$, only 34% of the original spectra survive (Figure 4.6). The AUC score and average precision both increase, as shown in Figure 4.7, but the important metric is the number of positive predictions made. The ViT, despite not de-redshifting spectra to the hosts' rest frames, is still able to identify more Host and SN II spectra than the CNN (which does require de-redshifting of the spectra). However, the confusion between the supernova subtypes remains relatively large.

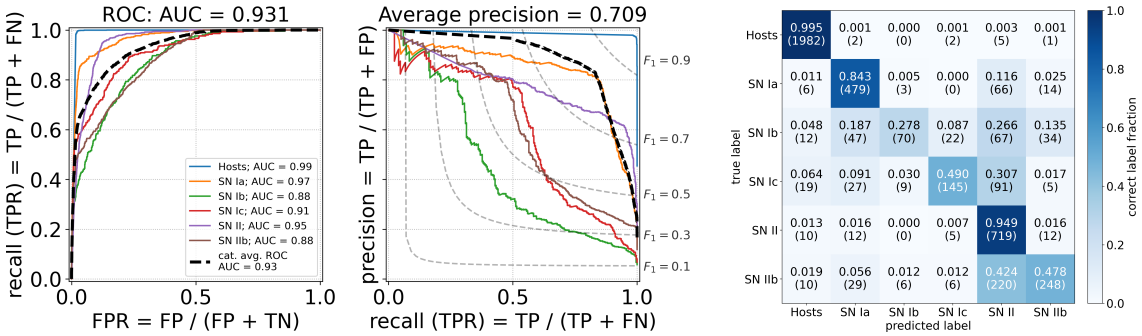


Figure 4.7: Spectral ViT V2 Diagnostics: ROC Curve (left) and Confusion Matrix (right) with a 99% confidence cut

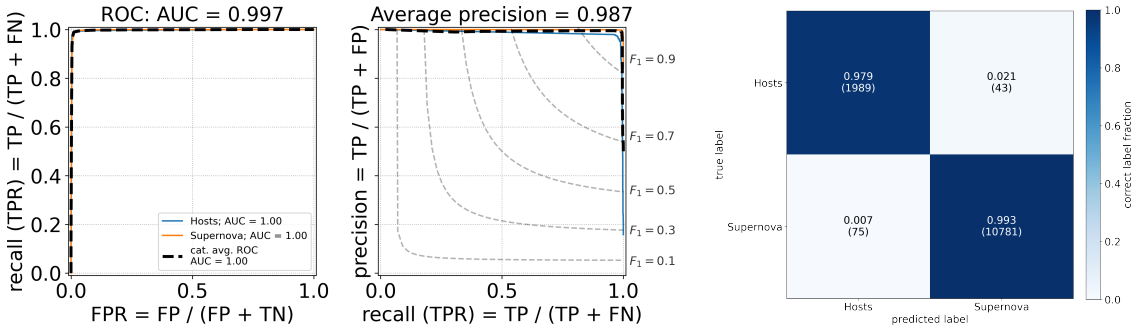


Figure 4.8: Spectral ViT V2 Binary Diagnostics: ROC Curve (left) and Confusion Matrix (right)

This confusion, combined with the lower AUC score, indicates that the ViT V2 is not a suitable replacement for the ViT V1 for the purposes of SNe sub-type classification. However, the ViT V2 is still able to differentiate SNe and non-SNe spectra with high accuracy, so alternative uses can be derived from the existing classifier.

4.2.1 Alternative Applications of the V2 Classifier

Binary Classifier

The most obvious use for a classifier capable of identifying the presence, or lack thereof, of a supernova with a high degree of accuracy is a binary classifier that simply identifies the presence of a supernova in the spectrum without sub-typing. To alter the predictions presented by the ViT for binary classification, the probabilities of all classes apart from non-SNe were summed. This resulted in a very high AUC score of 0.997, with a nearly perfect average precision of 98.7% (Figure 4.8). This, combined with the over 55% of sample classifications having a prediction confidence of over

99.9999%, indicates the ViT V2's outstanding ability to act as a binary classifier for the presence of SNe.

Progenitor and Type Classification

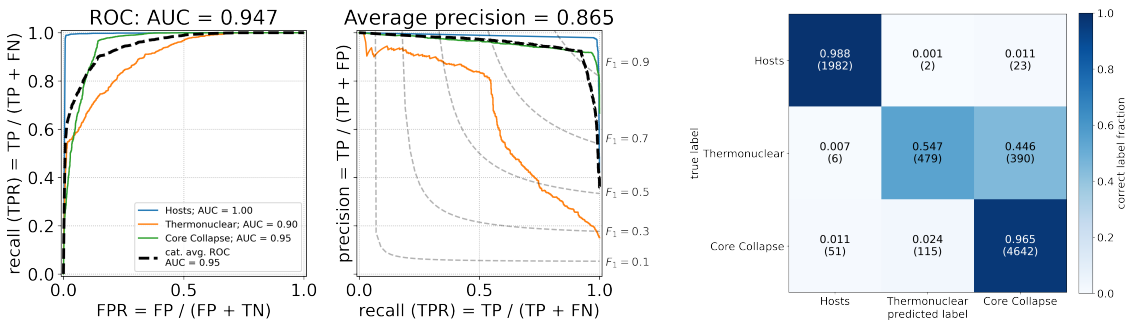


Figure 4.9: Spectral ViT V2 Progenitor Classifier Diagnostics: ROC Curve (left) and Confusion Matrix (right) with a 99% confidence cut

Beyond binary classification, the Spectral ViT V2 shows the capability to differentiate between thermonuclear SNe (Type Ia) and core collapse SNe (Types Ib,c, and Type II). With a 99% confidence cut, the classifier is able to retain 60% of the original sample, shown in Figure 4.9. Despite the great success in differentiating the core collapse and non-SNe spectra, there is some discrepancy between the two SNe categories. This, however, still results in an average precision of 86.5% which is slightly below the Spectral ViT V1 classifier on all subtypes, while recovering much of the original sample. Another classification scheme using three final classes would be to seperate the sample into the 2 main classifications: Type I vs Type II. Figure 4.10 shows the 99% cut on this classifications, yielding 44% of the original sample classified at an average precision of 91.8%. Therefore, the Spectral ViT V2 has many classification applications past just a binary classifier, regardless of its

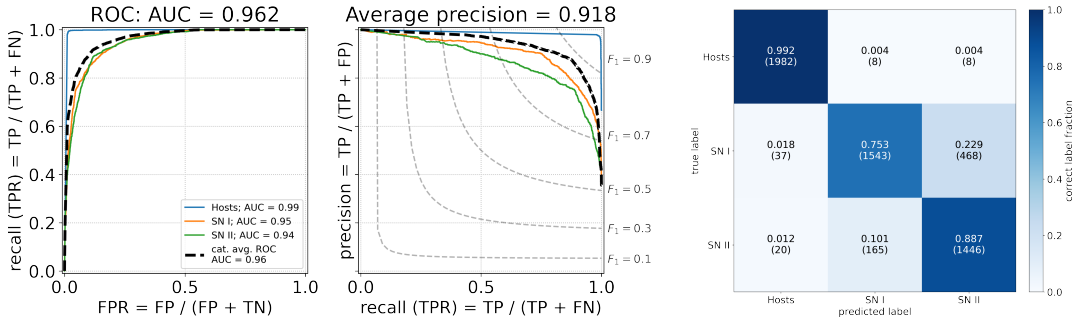


Figure 4.10: Spectral ViT V2 Type Classifier Diagnostics: ROC Curve (left) and Confusion Matrix (right) with a 99% confidence cut

inability to produce precise classification of SNe subtypes.

5. Conclusions

Supernovae classification is traditionally either a time-intensive or computationally expensive process. Possible candidates must almost always be re-visited by spectrographic analysis to confirm its origin: Type Ia for thermonuclear events, or Type Ib,c, and Type II for the death of massive stars via core collapse. For spectroscopic instruments, such as DESI, new techniques developed to classify transients quickly and efficiently would prove very useful, as no follow-up observations, nor human interaction, would ideally be necessary. In addition, due to the bright nature of these cataclysmic events, improper handling of targets with transients could lead to erroneous fits, leading to inaccurate data.

The field of deep learning encapsulates the creation and training of different neural network architectures designed to learn and identify patterns in data efficiently by front-loading computational costs via training. These frameworks can be applied to the categorical classification of spectra, as the set patterns and rules governing the different sub-categories are mutually exclusive, and synthetic training data can be created with relative ease.

We present a variation of the Vision Transformer architecture, named the Spectral

ViT, to be the successor to the 1D CNN currently utilized for spectral classification. After the model's ability to train was verified on a trivial dataset, the Spectral ViT was trained on both redshift-corrected (V1) and non-corrected (V2) spectra.

Using the same prepossessing methodology as its predecessors, the Spectral ViT V1 is able to recall 26% more correctly classified targets than the 2D CNN while also applying a stronger prediction cut. Compared to the 2D CNN, the average precision of the Spectral ViT is 13.3% worse. This performance hit is a result of the misclassification of supernova sub-types, as both architectures are able to differentiate between SNe and non-SNe spectra almost perfectly. This network, due to its higher recall, could be used to recover SNe missed by the 2D CNN. Further testing on visually confirmed, non-synthetic, DESI SNe is the next logical step.

Very bright SNe are capable of outshining their host galaxies, so it is possible that redshift failures in the DESI pipeline could occur in these targets, rendering the classifier, and most downstream tasks, impaired. The Spectral ViT V2 was trained for the purpose of eliminating dependence on the redrock fit. While it is unable to match the sub-type classification power of the Spectral ViT V1, the model exhibits high recall, high precision binary classification, and good precision in the classification of the spectra of host galaxies, Type Ia supernovae, and core-collapse supernovae. The V2 classifier could be implemented as a quick filter on top of an existing classification scheme, avoiding catastrophic redshift failures in case of bright supernovae.

Bibliography

Baade, W. and F. Zwicky (1934). "Cosmic Rays from Super-novae". In: *Proceedings of the National Academy of Sciences of the United States of America* 20.5, pp. 259–263. ISSN: 00278424. URL: http://www.jstor.org/stable/86841 (visited on 05/03/2023).

Barbary, Kyle et al. (2016). *sncosmo/sncosmo: v1.4.0*. DOI: 10.5281/ZENODO.168220. URL: https://zenodo.org/record/168220.

Baron, Dalya and Dovi Poznanski (Mar. 2017). "The weirdest SDSS galaxies: results from an outlier detection algorithm". In: *MNRAS* 465.4, pp. 4530–4555. DOI: 10.1093/mnras/stw3021. arXiv: 1611.07526 [astro-ph.GA].

Becker, Burger et al. (May 2021). "CNN architecture comparison for radio galaxy classification". In: *MNRAS* 503.2, pp. 1828–1846. DOI: 10.1093/mnras/stab325. arXiv: 2102.03780 [astro-ph.GA].

BenZvi, S., V. Tiwari, and A. Wasserman (Dec. 2020). *Discovering Transients (& other interesting things) in DESI Spectra*. Presentation.

Cybenko, G. (Dec. 1, 1989). "Approximation by superpositions of a sigmoidal function". In: *Mathematics of Control, Signals and Systems* 2.4, pp. 303–314. DOI: 10.1007/BF02551274. URL: https://doi.org/10.1007/BF02551274.

Davison, William, David Parkinson, and Brad E. Tucker (Feb. 2022). "STag: Supernova Tagging and Classification". In: *ApJ* 925.2, 186, p. 186. DOI: 10.3847/1538-4357/ac3422. arXiv: 2108.10497 [astro-ph.SR].

de Vaucouleurs, G. and Jr. Corwin H. G. (Aug. 1985). "S Andromedae 1885: A Centennial Review". In: *ApJ* 295, p. 287. DOI: 10.1086/163374.

DESI Collaboration et al. (2016). *The DESI Experiment Part I: Science,Targeting, and Survey Design*. arXiv: 1611.00036 [astro-ph.IM].

Dosovitskiy, Alexey et al. (2020). "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale". In: *CoRR* abs/2010.11929. arXiv: 2010.11929. URL: https://arxiv.org/abs/2010.11929.

Filippenko, Alexei V. (Jan. 1997). "Optical Spectra of Supernovae". In: *Ann. Rev. Astron. Astrophys.* 35, pp. 309–355. DOI: 10.1146/annurev.astro.35.1.309.

Fukushima, Kunihiko (1979). "Self-Organization of a Neural Network Which Gives Position-Invariant Response". In: *Proceedings of the 6th International Joint Conference on Artificial Intelligence - Volume 1*. IJCAI'79. Tokyo, Japan: Morgan Kaufmann Publishers Inc., pp. 291–293. ISBN: 0934613478. DOI: 10.5555/1624861.1624928.

Gauci, Adam, Kristian Zarb Adami, and John Abela (May 2010). "Machine Learning for Galaxy Morphology Classification". In: *arXiv e-prints*, arXiv:1005.0390, arXiv:1005.0390. DOI: 10.48550/arXiv.1005.0390. arXiv: 1005.0390 [astro-ph.GA].

Graur, Or and Dan Maoz (Apr. 2013). "Discovery of 90 Type Ia supernovae among 700 000 Sloan spectra: the Type Ia supernova rate versus galaxy mass and star formation rate at redshift ∼0.1". In: *MNRAS* 430.3, pp. 1746–1763. DOI: 10.1093/mnras/sts718. arXiv: 1209.0008 [astro-ph.CO].

Guy, J. et al. (Mar. 2023). "The Spectroscopic Data Processing Pipeline for the Dark Energy Spectroscopic Instrument". In: *The Astronomical Journal* 165.4, p. 144. DOI: 10.3847/1538-3881/acb212. URL: https://doi.org/10.3847/1538-3881/acb212.

Hahn, ChangHoon et al. (2022). *DESI Bright Galaxy Survey: Final Target Selection, Design, and Validation*. arXiv: 2208.08512 [astro-ph.CO].

He, Kaiming et al. (2015). *Deep Residual Learning for Image Recognition*. DOI: 10.48550/ARXIV.1512.03385. URL: https://arxiv.org/abs/1512.03385.

Hsiao, E. Y. et al. (July 2007). "K-Corrections and Spectral Templates of Type Ia Supernovae". In: *The Astrophysical Journal* 663.2, pp. 1187–1200. DOI: 10.1086/518232. URL: https://doi.org/10.1086/518232.

Huang, Gao et al. (2016). *Densely Connected Convolutional Networks*. DOI: 10.48550/ARXIV.1608.06993. URL: https://arxiv.org/abs/1608.06993.

Kim, Yoon (2014). *Convolutional Neural Networks for Sentence Classification*. DOI: 10.48550/ARXIV.1408.5882. URL: https://arxiv.org/abs/1408.5882.

Kiranyaz, Serkan et al. (2021). "1D convolutional neural networks and applications: A survey". In: *Mechanical Systems and Signal Processing* 151, p. 107398. ISSN: 0888-3270. DOI: https://doi.org/10.1016/j.ymssp.2020.107398. URL: https://www.sciencedirect.com/science/article/pii/S0888327020307846.

Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E Hinton (2012). "ImageNet Classification with Deep Convolutional Neural Networks". In: *Advances in Neural Information Processing Systems*. Ed. by F. Pereira et al. Vol. 25. Curran Associates, Inc. URL: https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf.

Kumar, Ajitesh (Apr. 2022). *Different types of CNN Architectures explained: Examples*. URL: https://vitalflux.com/different-types-of-cnn-architectures-explained-examples/.

LeCun, Y., F. Huang, and L. Bottou (2004). "Learning methods for generic object recognition with invariance to pose and lighting". In: *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004*. Vol. 2, II–104 Vol.2. DOI: 10.1109/CVPR.2004.1315150.

Madgwick, Darren S. et al. (Dec. 2003). "Spectroscopic Detection of Type Ia Supernovae in the Sloan Digital Sky Survey". In: *ApJ* 599.1, pp. L33–L36. DOI: 10.1086/381081. arXiv: astro-ph/0310887 [astro-ph].

Mikolov, Tomas et al. (Jan. 2013). "Efficient Estimation of Word Representations in Vector Space". In: *arXiv e-prints*, arXiv:1301.3781, arXiv:1301.3781. DOI: 10.48550/arXiv.1301.3781. arXiv: 1301.3781 [cs.CL].

Minkowski, R. (Aug. 1941). "Spectra of Supernovae". In: *PASP* 53.314, p. 224. DOI: 10.1086/125315.

Möller, A. et al. (Dec. 2016). "Photometric classification of type Ia supernovae in the SuperNova Legacy Survey with supervised learning". In: *Journal of Cosmology and Astroparticle Physics* 2016.12, pp. 008–008. DOI: 10.1088/1475-7516/2016/12/008. URL: https://doi.org/10.1088/1475-7516/2016/12/008.

Muthukrishna, Daniel, David Parkinson, and Brad E. Tucker (Nov. 2019). "DASH: Deep Learning for the Automated Spectral Classification of Supernovae and Their Hosts". In: *ApJ* 885.1, 85, p. 85. DOI: 10.3847/1538-4357/ab48f4. arXiv: 1903.02557 [astro-ph.IM].

Naskath, J., G. Sivakamasundari, and A. Alif Siddiqua Begum (Oct. 2022). "A Study on Different Deep Learning Algorithms Used in Deep Neural Nets: MLP SOM and DBN". In: *Wireless Personal Communications* 128.4, pp. 2913–2936. DOI: 10.1007/s11277-022-10079-4. URL: https://doi.org/10.1007/s11277-022-10079-4.

Nassif, Ali Bou et al. (2019). "Speech Recognition Using Deep Neural Networks: A Systematic Review". In: *IEEE Access* 7, pp. 19143–19165. DOI: 10.1109/ACCESS.2019.2896880.

Neilsen, Michael A. (2015). *Neural Networks and Deep Learning*. Determination Press.

Neutelings, Izaak (Sept. 26, 2021). *Neural networks*. URL: https://tikz.net/neural_networks/.

— (Apr. 9, 2022). *Neural networks*. URL: https://tikz.net/conv2d/.

Parks, David et al. (Jan. 2018). "Deep learning of quasar spectra to discover and characterize damped Lyα systems". In: *Monthly Notices of the Royal Astronomical Society* 476.1, pp. 1151–1168. ISSN: 0035-8711. DOI: 10.1093/mnras/sty196.

eprint: https://academic.oup.com/mnras/article-pdf/476/1/1151/24261051/sty196.pdf. URL: https://doi.org/10.1093/mnras/sty196.

Paszke, Adam et al. (2019). "PyTorch: An Imperative Style, High-Performance Deep Learning Library". In: *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., pp. 8024–8035. URL: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

Pence, W. D. et al. (Nov. 2010). "Definition of the Flexible Image Transport System (FITS), version 3.0". In: *A&A* 524, A42. DOI: 10.1051/0004-6361/201015362. URL: https://doi.org/10.1051/0004-6361/201015362.

Perlmutter, S. et al. (Feb. 1995). "A Supernova at Z = 0.458 and Implications for Measuring the Cosmological Deceleration". In: *ApJl* 440, p. L41. DOI: 10.1086/187756. arXiv: astro-ph/9505023 [astro-ph].

Popescu, Marius-Constantin et al. (July 2009). "Multilayer perceptron and neural networks". In: *WSEAS Transactions on Circuits and Systems* 8.

Rubenzahl, Ryan (May 2018). "Identifying Type Ia Supernovae in Extragalactic Spectra". Bachelor's Thesis. Rochester, NY: University of Rochester.

Schaefer, Bradley E. (July 1995). "The Peak Brightness of SN 1895B in NGC 5253 and the Hubble Constant". In: *ApJ* 447, p. L13. DOI: 10.1086/309549.

Sepeku, E. (May 2022). "Spectroscopic Identification of Supernova In DESI". Bachelor's Thesis. Rochester, NY: University of Rochester.

Shelhamer, Evan, Jonathan Long, and Trevor Darrell (2016). *Fully Convolutional Networks for Semantic Segmentation*. DOI: 10.48550/ARXIV.1605.06211. URL: https://arxiv.org/abs/1605.06211.

Simonyan, Karen and Andrew Zisserman (2015). "Very Deep Convolutional Networks for Large-Scale Image Recognition". In: *International Conference on Learning Representations*.

Transient Name Server (2015). URL: https://www.wis-tns.org/stats-maps.

Turatto, M. (2003). "Classification of Supernovae". In: *Supernovae and Gamma-Ray Bursters*. Ed. by K. Weiler. Vol. 598, pp. 21–36. DOI: 10.1007/3-540-45863-8_3.

Vaswani, Ashish et al. (2017). "Attention is All you Need". In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon et al. Vol. 30. Curran Associates, Inc. URL: https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

Vincenzi, M et al. (Sept. 2019). "Spectrophotometric templates for core-collapse supernovae and their application in simulations of time-domain surveys". In: *Monthly Notices of the Royal Astronomical Society* 489.4, pp. 5802–5821. ISSN: 0035-8711. DOI: 10.1093/mnras/stz2448. eprint: https://academic.oup.

com/mnras/article-pdf/489/4/5802/30097806/stz2448.pdf. URL: https://doi.org/10.1093/mnras/stz2448.

Wasserman, A. (May 2021). "Using Machine Learning to Identify Transients in the DESI Survey". Bachelor's Thesis. Rochester, NY: University of Rochester.

Xiong, Ruibin et al. (Feb. 2020). "On Layer Normalization in the Transformer Architecture". In: *arXiv e-prints*, arXiv:2002.04745, arXiv:2002.04745. DOI: 10.48550/arXiv.2002.04745. arXiv: 2002.04745 [cs.LG].

A. Diagrams of Neural Networks

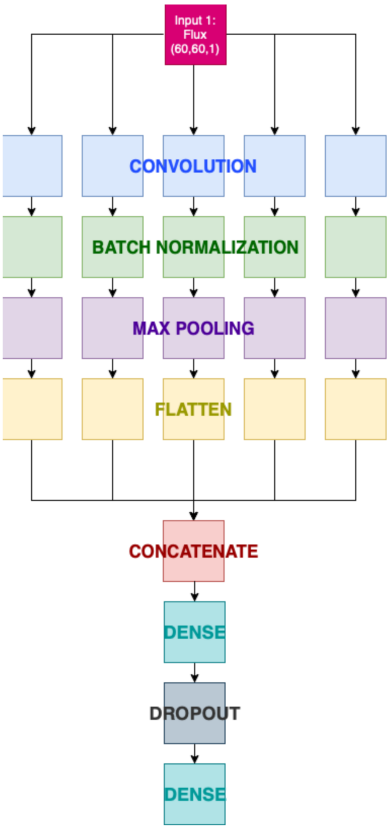


Figure A.1: CNN architecture. Figure from Sepeku (2022).

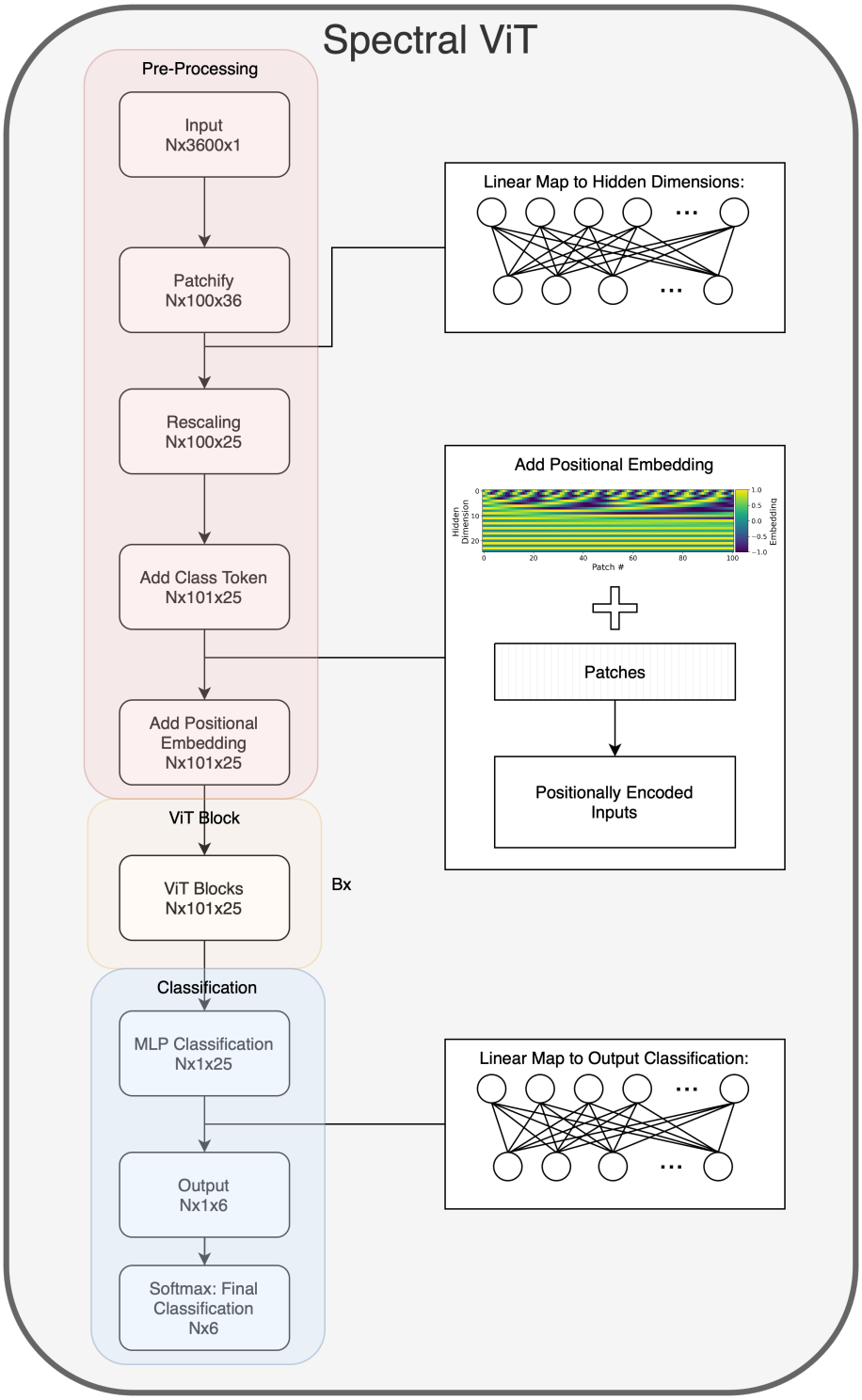


Figure A.2: Spectral ViT Architecture

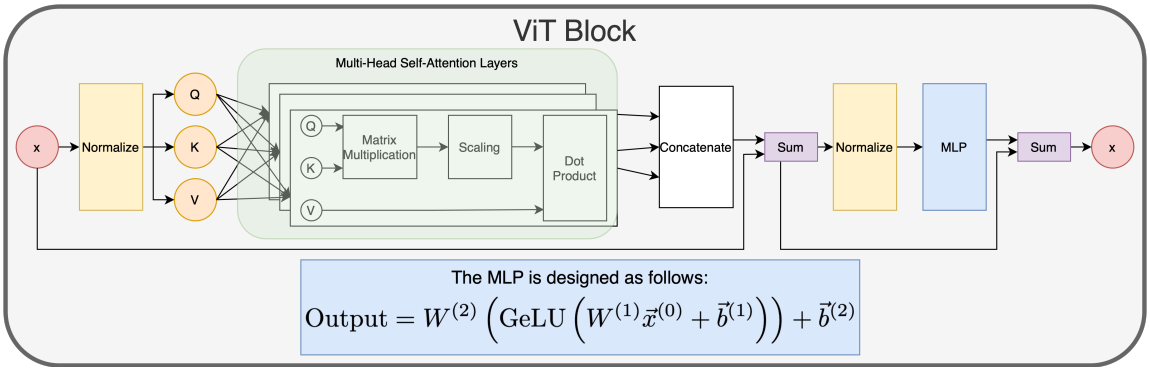


Figure A.3: The ViT Block Subarchitecture includes the mutli-head self attention layers, a normalization layer, and an MLP.