Contrastive learning for the classification of varible stars

Author:
Niall MILLER

Supervised by:

Centre for Astrophysics Research
Physics, Engineering and Computer Science
University of Hertfordshire

Submitted to the University of Hertfordshire in partial fulfilment of the requirements of the degree of .

March 2024

Abstract

We demonstrate that it is possible to extract semantically meaningful fixed length representations of stochastically sampled time series data. We use a novel neural network architecture (SimCLR with a gated recurrent neural network backbone) to go about this.

Chapter 1

Introduction

The extremely large surveys typifying astronomy's Big Data Era will be impossible to parse manually (Minniti et al., 2010; Ivezić et al., 2019; Dewdney et al., 2009). If we are to consistently interrogate this data deluge at scale we need to devise reliable and robust automated methods. Deep learning has already gained a foothold in many data intensive fields, from astronomy, to particle physics, to chemistry. Deep learning is therefore a natural solution to astronomy's inherent scaling problem.

While supervised deep learning has been applied again (Storrie-Lombardi et al., 1992), again (Belokurov et al., 2003), and again (Charnock and Moss, 2017) in the quest to classify astronomical objects, its uses are limited by the availability of high quality labelled data. If there is no reliably labelled dataset one must turn to unsupervised or self-supervised methods to sort known categories of objects, and also to the find the 'unknown unknowns'—objects so obscure that they defy classification.

Self-supervised representation learning has recently exploded in popularity, with a slew of models being developed in rapid succession (i.e. Chen et al., 2020; Chen et al., 2020a; Grill et al., 2020; He et al., 2019; Chen et al., 2020b). At its core, representation learning attempts to produce semantically meaningful compressed representations (or embeddings) of complex highly dimensional data. Aside from simply being a compression device, these embeddings can also be taken and used in downstream tasks, like clustering, anomaly detection, or classification.

In recent years, pioneering work has applied self-supervised contrastive learning models to galaxy image clustering. Abul-Hayat et al. (2020) trained a simple framework for contrastive learning representations,(SimCLR; Chen et al., 2020) on multi-band galaxy photometry from the Sloan Digital Sky Survey,(SDSS; York et al., 2000). They demonstrated that the resulting embeddings capture useful information by using them directly in a training set for a galaxy morphology classification model and a redshift estimation model. Similarly, Sarmiento et al.

(2021) trained a SimCLR model on integral field spectroscopy data from galaxies in the Mapping Nearby Galaxies at Apache Point Observatory survey (MaNGA; Bundy et al., 2015). They also found that SimCLR produces semantically meaningful embeddings. With these recent successes in mind, we ask: can we also use contrastive learning to interrogate astronomical time series data? In this work we address this question and leverage self supervised contrastive learning to explore the VISTA Variables in the Vía Láctea survey (VVV; Minniti et al., 2010).

In a concurrent work Donoso-Oliva et al. (2022) approach the problem of time series representation learning from a natural language processing (NLP) perspective. They repurpose the BERT (Bidirectional Encoder Representations from Transformers) Transformer network, which was initially developed in the context of NLP (Vaswani et al., 2017; Devlin et al., 2019). They then perform a 'pretraining' task on light curves, using the network to fill in zeroed datapoints within the time series. Once this pretraining task is completed, semantically meaningful embeddings can be extracted from the transformer network. Donoso-Oliva et al. (2022) show that these embeddings are useful for the downstream task of classification.

Chapter 2

Contrastive self-supervised learning

Figure 2.1 describes a simple contrastive learning model in the vein of SimCLR (Chen et al., 2020) (this will be referred to as 'contrastive curves' throughout). This model takes as input a sample ($\mathbf{x}$) from the training set, and augments it to produce $\mathscr{A}(\mathbf{x})$. This augmentation is performed in such a way that $\mathscr{A}(\mathbf{x})$ shares enough semantically meaningful data with $\mathbf{x}$ to belong to the same class of objects. In the contrastive learning literature $(\mathbf{x}, \mathscr{A}(\mathbf{x}))$ is known as a positive pair. This positive pair is then passed to a Siamese neural network Φ, which projects the high dimensional input data onto a lower dimensional latent space. All other training set samples are assumed to belong to a different class to $\mathbf{x}$, and so can be combined with $\mathbf{x}$ to produce 'negative pairs'.

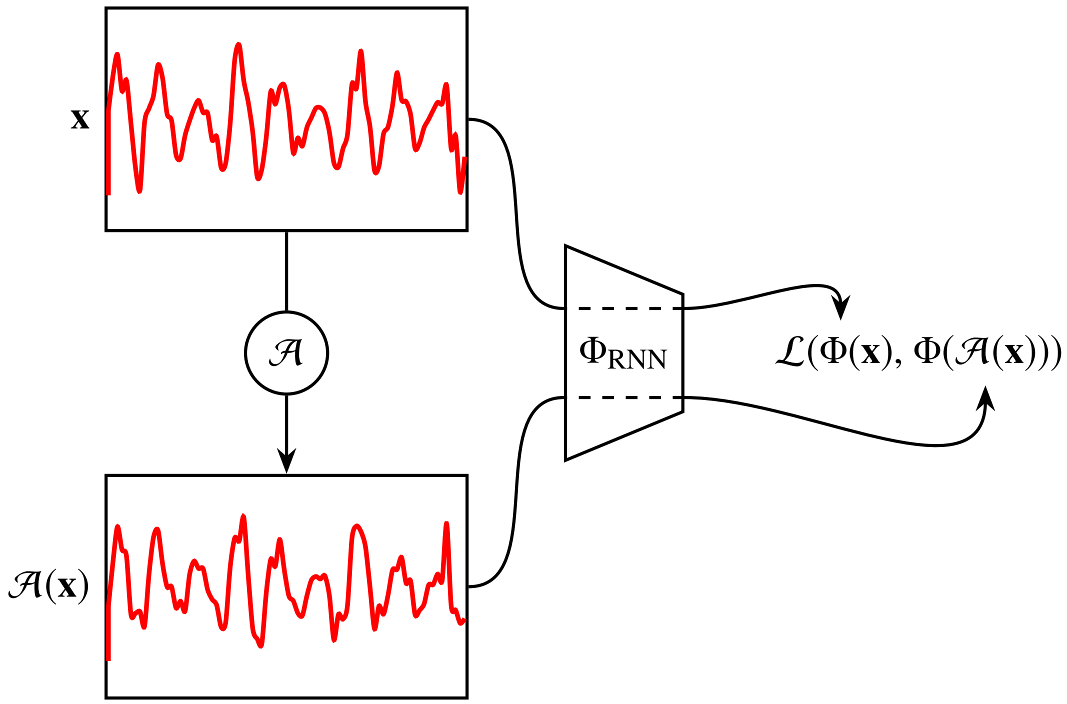
We use the normalised temperature cross entropy (NT-Xent) loss as our contrastive loss. The NT-Xent loss was first introduced in Sohn (2016), and was subsequently popularised by Chen et al. (2020). The NT-Xent loss is defined as

$$\mathscr{L}(\mathbf{z}_i, \mathbf{z}_j) = -\log\left(\frac{\exp(\mathbf{z}_i^T \mathbf{z}_j / \mathscr{T})}{\sum_{k=1}^{2N}(1-\delta_{ki})\exp(\mathbf{z}_i^T \mathbf{z}_k / \mathscr{T})}\right), \tag{2.1}$$

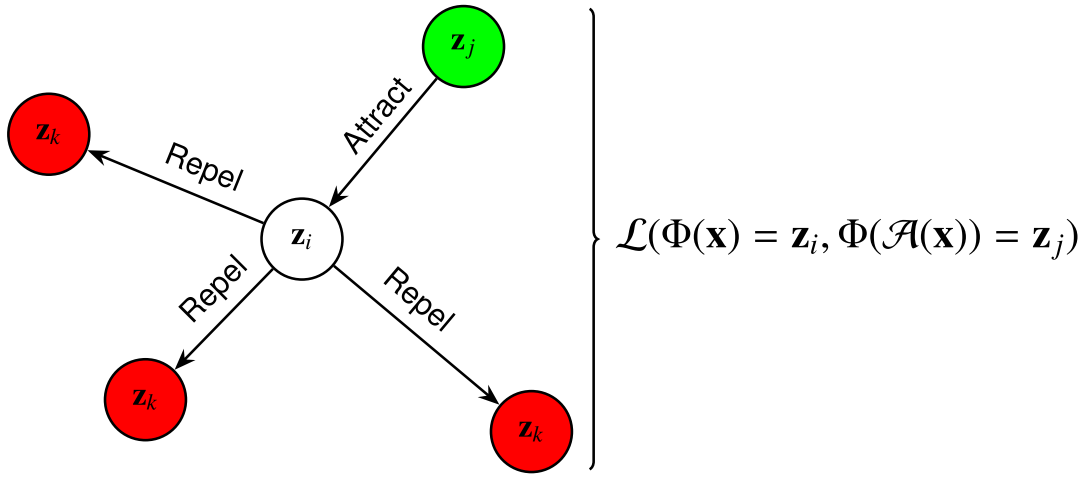
where $\mathbf{z}_i$ and $\mathbf{z}_j$ are a positive pair, and $\mathbf{z}_i$ and $\mathbf{z}_k$ are a negative pair. All embeddings are normalised. $\mathscr{T}$ is a 'temperature' hyperparameter introduced in Chen et al. (2020) to help the model learn from hard negatives. δ is the Kronecker delta.

As shown in figure 2.1b, minimising the NT-Xent loss minimises the distance in the embedding space between positive pairs while simultaneously maximising the distance between negative pairs. Therefore, once training is completed we expect to have moulded a semantically meaningful embedding space with similar vectors clustered close together.

In figure 2.2, we show a representation of our chosen model: a stacked bidirectional gated recurrent unit (GRU),(Cho et al., 2014). Due to the variable lengths of our input time series, we use a recurrent neural network. By taking the hidden states of our neural network, we convert the



(A) A simple contrastive learning model for time series data.



(B) The NT-Xent loss incentivises attraction in the latent space between similar examples while simultaneously incentivising repulsion between dissimilar examples.

FIGURE 2.1: In figure 2.1a a simple contrastive learning model is applied to time-series data. $\mathscr{A}$ is an augmentation pipeline. $\mathscr{A}$ could consist of noise addition, stochastic temporal shifting, and random data deletion. Φ is a function approximator that projects inputs onto an embedding space. Φ is typically a neural network; when processing time-series data Φ could be a recurrent neural network (RNN; McCulloch and Pitts, 1943). The loss $\mathscr{L}$ measures the distance between the embeddings $\Phi(\mathbf{x}) = \mathbf{z}_i$ and $\Phi(\mathscr{A}(\mathbf{x})) = \mathbf{z}_j$, and we train by attempting to minimise this distance while maximising the distance between dissimilar samples (figure 2.1b).

variably lengthed light curves to a fixed-length representation. Once we have this fixed-length representation, we can follow Chen et al. (2020) and use a single hidden layer fully connected neural network (i.e., $\mathbf{z} = g(\mathbf{h}) = W_2, \texttt{ReLU}(W_1\mathbf{h}))$ to project the representation onto a final 64-dimensional space. We train on the vectors in this final space.

Our model is written in PyTorch (Paszke et al., 2019) and is available under the GNU Affero General Public License v3.0 at `https://github.com/nialljmiller/contrastive-curves`.

In Deb and Singh (2009) a different method is used to create representations of light curves without any priors. Their work presents a methodology for analysing light curves of variable stars, with both Fourier decomposition and PCA as principal analytical tools. This methodology is particularly designed to address the challenges posed by the non-uniform sampling of light curves, which is a common issue in observational astronomy. The authors implement a preprocessing step that involves phase-folding the light curves based on the stars' periods, then interpolating the magnitudes to achieve uniform sampling across the phase from 0 to 1 in steps of 0.01. This process ensures that each light curve is represented by a uniformly spaced set of points, making the data compatible with Fourier decomposition, which requires uniform sampling. For the Fourier decomposition analysis, the method transforms the light curves into a sum of cosine and sine series, thus allowing for the characterisation of the light curve through the Fourier parameters. This method has limitations however, Fourier analysis is fundamentally a fitting technique. While we can increase the Fourier terms to ultimately approximate any shape of light curve, this technique is computationally expensive and difficult to tune (BAART, 1982). PCA is employed as a more scaleable solution for analysing and classifying variable stars within large datasets. By directly using the interpolated light curve magnitudes as input, PCA bypasses the need for pre-computing Fourier coefficients, offering a significant advantage in terms of computational efficiency. The PCA transforms the original dataset into a new set of uncorrelated variables (principal components), which represent the most significant patterns within the data. However, the PCA method relies on linear assumptions about the data it analyses, which might not always be suitable for variable star light curves where non-linear phenomena govern brightness variations. Contrastive curves utilises data augmentation techniques to create positive pairs from the original data. This approach is particularly adept at capturing nuanced similarities between light curves that PCA might overlook due to its linear transformation and variance-focused dimensionality reduction. In contrast, contrastive curves, especially when implemented with neural networks such as bi-directional gated recurrent units (GRUs), can capture complex non-linear relationships within the data. This capability allows for a more nuanced understanding of the underlying astrophysical processes reflected in the light curves. Contrastive curves aims to create a semantically meaningful embedding space where similar examples are clustered together while dissimilar examples are repelled from each other. This approach can be particularly advantageous for classifying variable stars into their correct classes based on the intrinsic properties of their light curves. PCA, on the other hand, might not yield an embedding

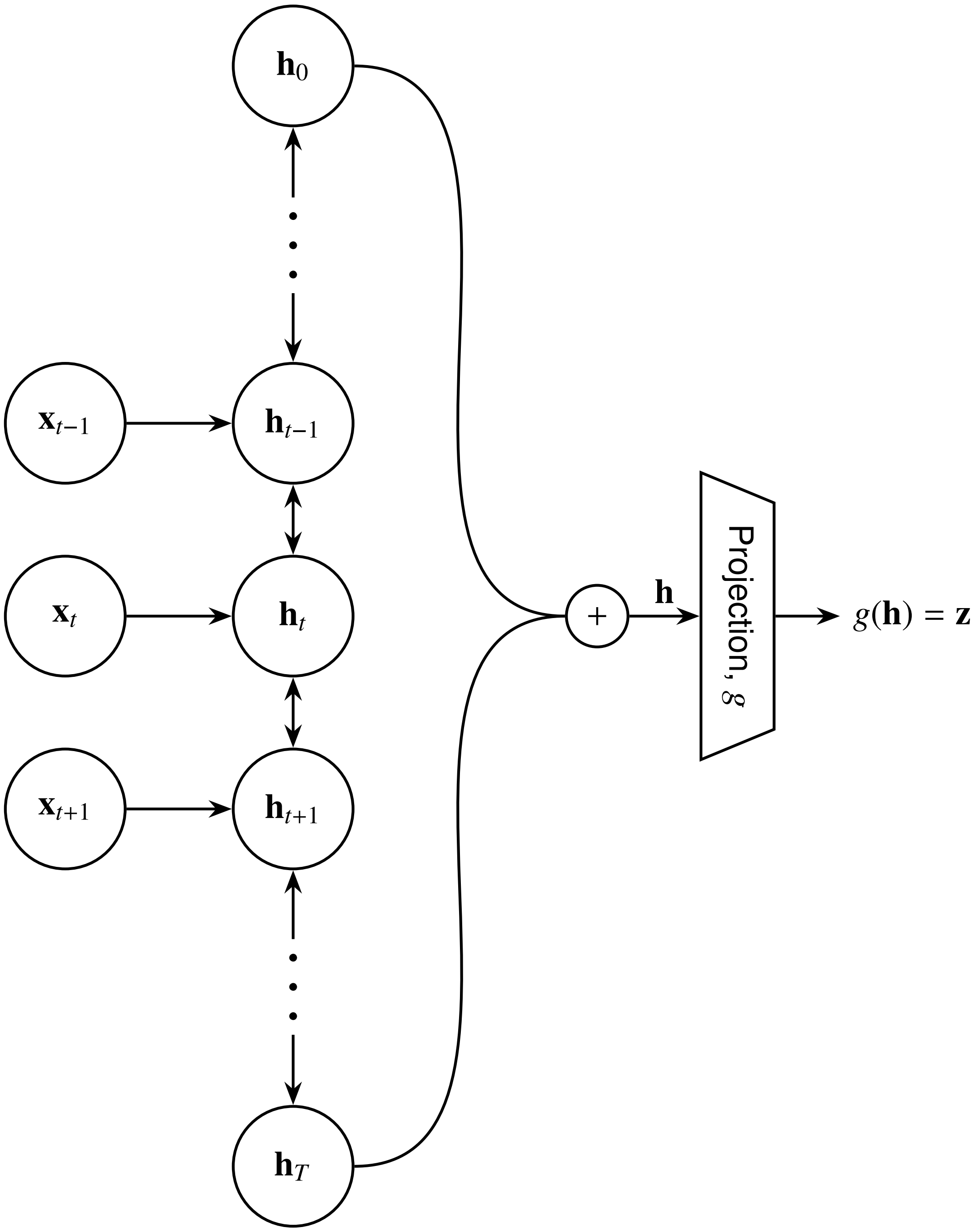


FIGURE 2.2: A variable star is input into our model. The star's time series is denoted $\mathbf{x}_t$. The time series is first passed into a bidirectional GRU network with an initial hidden state denoted $\mathbf{h}_0$ and a final hidden state denoted $\mathbf{h}_T$. The initial and final hidden states are concatenated along the channel axis, and the resulting vector $\mathbf{h}$ is passed through a linear projector. The output vector $\mathbf{z}$ is used for training (Eqn. 2.1). At inference time we follow Chen et al. (2020) and take $\mathbf{h}$ as the representation.

space that is as intuitively interpretable in terms of semantic similarity, as it primarily focuses on variance maximization.

To train our contrastive curves model, we generate slightly altered versions of each light curve. These alterations, or augmentations, include adding noise, randomly shifting time points, and cropping sections. These augmentations are chosen to represent changes to a light curve that we could see between light curves of identical classes of variable stars. The purpose of these augmentations is to teach the model to recognise the essential features of the light curves despite these changes. Each original light curve and its augmented version form a positive pair, meaning they should be recognized as similar by the neural network. Conversely, each original light curve paired with different light curves forms negative pairs, which the model should recognise as dissimilar. The GRU, which comprises the majority of the network architecture, processes the input light curves and transforms them into fixed-length vectors, or embeddings. These embeddings are lower-dimensional representations that capture the significant features of the light curves.

The NT-Xent loss function helps the network learn by making sure that the embeddings of positive pairs are close together, while the embeddings of negative pairs are far apart. After training, the model can take any new light curve and convert it into an embedding. These embeddings can then be used for various analysis tasks, such as clustering similar stars, classifying different types of variable stars, or detecting anomalies. The contrastive curves method brings several advantages to the field of astronomy; it can efficiently handle the massive datasets generated by modern astronomical surveys, it is designed to be resilient to noise and irregular sampling (given the appropriate design of augmentations), and the embeddings produced can be used for a wide range of downstream tasks.

2.1 Data sample, preparation, and training

The light curves were taken from VVV light curves in the PRIMVS catalogue (Miller et al in prep). PRIMVS contains ≈ 5 million periodic variable stars with low false alarm probability.

Due to the unique nature of the VVV survey, cross matching with pre-existing periodic variable catalogues was limited. Each light curve's period was calculated via phase folding and Fourier based techniques with a false alarm probability assigned from a machine learning technique. This allows for multiple periods to be calculated via fundamentally different methods and the period with the lowest false alarm probability to be used. This is performed in an attempt to remove any selection bias for periodic variable stars with previously unknown phase folded structures such as the difference between AGB and EB, even though these are known.

For each input light curve a phase was calculated from the best period. A time resolution of 0.25 days was used such that any datapoints within this distance to each other were combined and their photometric error reduced as a function of $1/\sqrt{N}$. This is required to minimise the longest light curve length within a batch; if we have a single very long light curve within a batch, the neural network will pad every curve within the batch to the length of the longest light curve. In extreme cases this requires more VRAM than is available on the GPU machine and halts training. After this reprocessing, the number of datapoints per light curve varies between 40 and 2000, with a median of ~ 150. Further cuts were used to ensure each light curve only contained reliable data-points that exclusively feature the photometry of the target star. This involved selecting for both photometric and astrometric error. A selection of the following was used:

- $m_{error} < 0.5$
- $m_{error} < 3 \times \bar{m_{error}}$
- 'ast_res_chisq' < 100
- 'chi' < 10
- 'ambi-match' $= 0$

Where 'ast_res_chisq', 'chi', and 'ambi-match' are astrometric values taken from the `DoPHOT` point spread function (PSF) fitting code. 'ast_res_chisq' and 'chi' characterise the goodness of fit for the PSF. 'ambi-match' is a boolean flag which signifies if the source appears blended with a neighbour.

We normalise our magnitudes as

$$\bar{\mathbf{x}}_m = 2\left(\frac{\mathbf{x}_m - A}{B - A}\right) - 1 \tag{2.2}$$

where apparent magnitude is denoted x_m. A is set as the 90% completeness limit of the VVV survey (16.8 mag), and B is set as the VVV survey's saturation point (12 mag). This scaling ensures that all magnitudes are roughly scaled between -1 and 1. Since the light curves are already phase folded, we can pair our magnitudes with their phases. All the phases are originally scaled between 0 and 1. Since the phase is cyclical, we embed it as a two channel vector

$$\bar{\phi} = (\sin(\tau\phi), \cos(\tau\phi)). \tag{2.3}$$

The final light curves as seen by the model have three channels: magnitude and the two channels of encoded phase. We select the following augmentations for our model:

- We want the learnt features to be invariant to the telescope's sampling schedules. To this end we apply a random datapoint deletion of our incoming sequences as an augmentor. In practice we apply dropout (Srivastava et al., 2014) at a 10% rate on our sequences.

- We also do not want the representations to be dependent on the light curve length, and so we also always apply a random crop along the time axis.

- To enforce phase invariance in the light curves we apply a randomised phaseshift on the phase folded light curves. In practice we sample a phase from $\alpha \sim \mathscr{U}(0, \tau)$, and rotate the phase channels via the trigonometric identities $\sin(\tau\phi + \alpha) = \sin(\tau\phi)\cos\alpha + \cos(\tau\phi)\sin\alpha$, and $\cos(\tau\phi + \alpha) = \cos(\tau\phi)\cos\alpha - \sin(\tau\phi)\sin\alpha$.

- The data is affected by instrumental noise. As we do not want the model to use this information in its representations, we apply a random noise addition in our augmentation pipeline. This noise is sampled from $\mathscr{N}(0, \lambda\Delta m)$, where Δm is the median magnitude error of the time series. λ is a hyperparameter. We set λ to 1 to take into account error sources that are not represented in Δm.

- We apply an amplitude jitter to the magnitude channel. This is of the form of a random resample within a flat distribution between 1-1.05 of the amplitude. Without prior classification of the light curves we cant know the expected amplitude range for the source. Instead, we use a conservative amplitude jitter as being a reasonable alternative.

We always apply random phase shift and random cropping. All other augmentations are applied at a 50% rate.

The fina1 model is trained for 50 000 iteration steps on a single NVIDIA Tesla V100. Training completes in a wall time of roughly 18.5 hours.

2.2 Tuning

Due to the novelty of this method it is not trivial to decide on input parameter values for the architecture and training of this network. Table 2.1 shows the hyperparameters used and their justification. As it can take on the order of days to fully train the model, two separate grid searches[1] were performed to determine all of the hyperparameters.

The grid search method used for this network was not as simple as deciding the iteration which produced the lowest loss and highest accuracy. This is because self supervised learning does not provide a loss or accuracy measure which can be used to directly determine the effectiveness of

[1]Training the network multiple times with an array of different hyperparameters

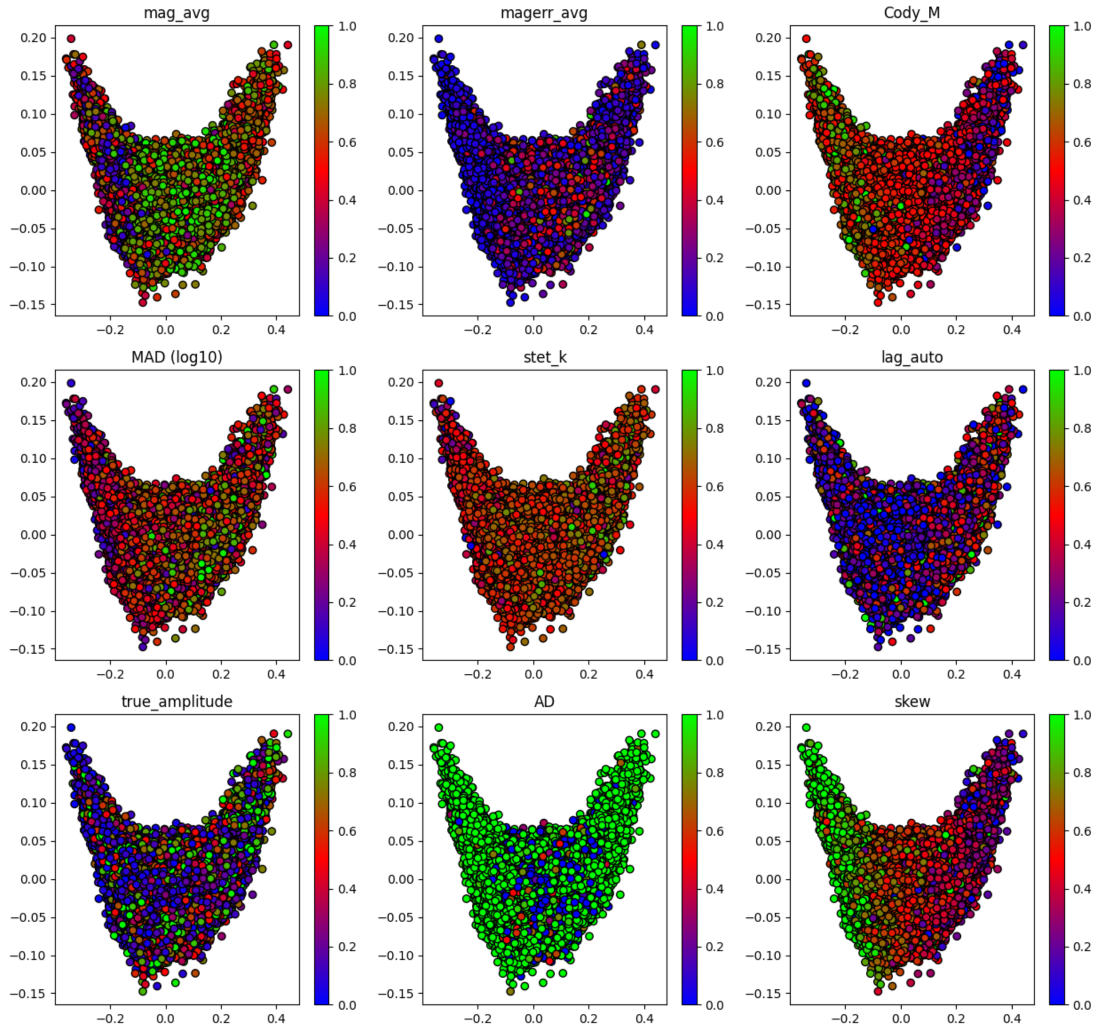


FIGURE 2.3: A PCA representation of the latent space trained with; learning rate = 0.001, tau = 0.05 and gamma = 0.7.

the neural network. The NT-Xent loss is used to construct a semantically meaningful embedding space, to determine the effectiveness of the network we must visually inspect the embedding space. Our metric for determining the '*best*' latent space representation was to inspect the structure of both the UMAP (McInnes et al., 2018) and PCA (F.R.S., 1901) projections. We also inspect the relationship between these projections and features from the PRIMVS catalogue.

Figures 2.3 -2.4 show the variety of projections we receive with relatively minimal changes to the input parameters. Each subplot is colour coded with the normalised values from the PRIMVS catalogue, these are (from top left to bottom right): Average magnitude, Average magnitude error, M (Cody et al., 2014), Median Absolute Deviation, Stetson-k index (Stetson, 1996), Lag-1 autocorrelation, Amplitude, Anderson Darling (Anderson and Darling, 1952) and Skew.

The observed sensitivity in latent space with respect to the selected hyperparameters means it is likely we have not selected the most optimal values. However, the values we have selected

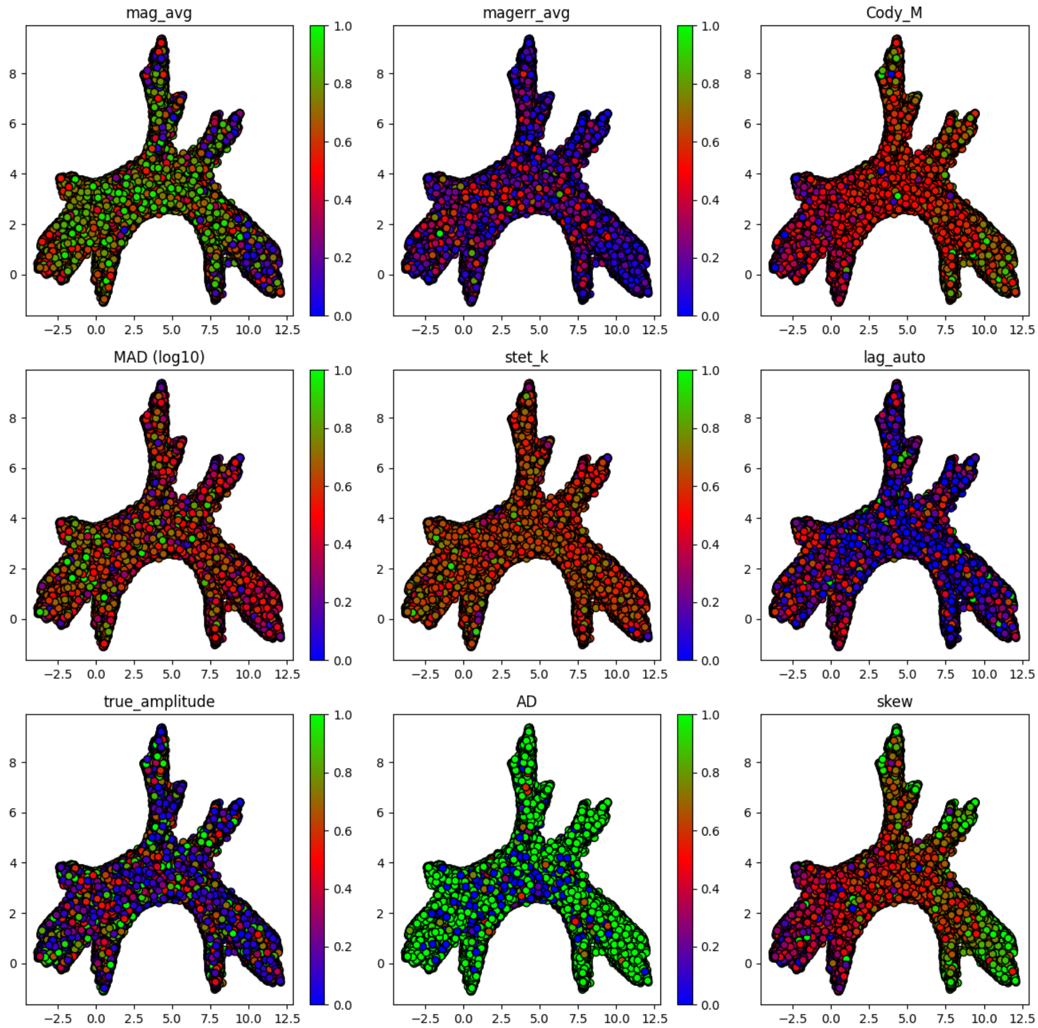


FIGURE 2.4: A PCA representation of the latent space trained with; learning rate = 0.01, tau = 0.05 and gamma = 0.9

TABLE 2.1: Tuning parameters and their justifications

Hyperparameter	Value	Justification
Batch Size	4096	GPU memory limited
Drop out rate	10%	Grid search #1
Hidden Dimensions	64	Grid search #1
Learning rate	0.0001	Grid search #2
Tau	0.05	Grid search #2
Gamma	0.7	Grid search #2
Output Dimensions	64	Final tuning

produce a semantically meaningful latent space projection and further tuning will require a more intelligent approach.

The first grid search "Grid search # 1" was used to determine parameters that do not specifically pertain to the training loop - drop out rate and the number of hidden dimensions. The drop out rate determines the probability of any point in a light curve being removed. This has proven to

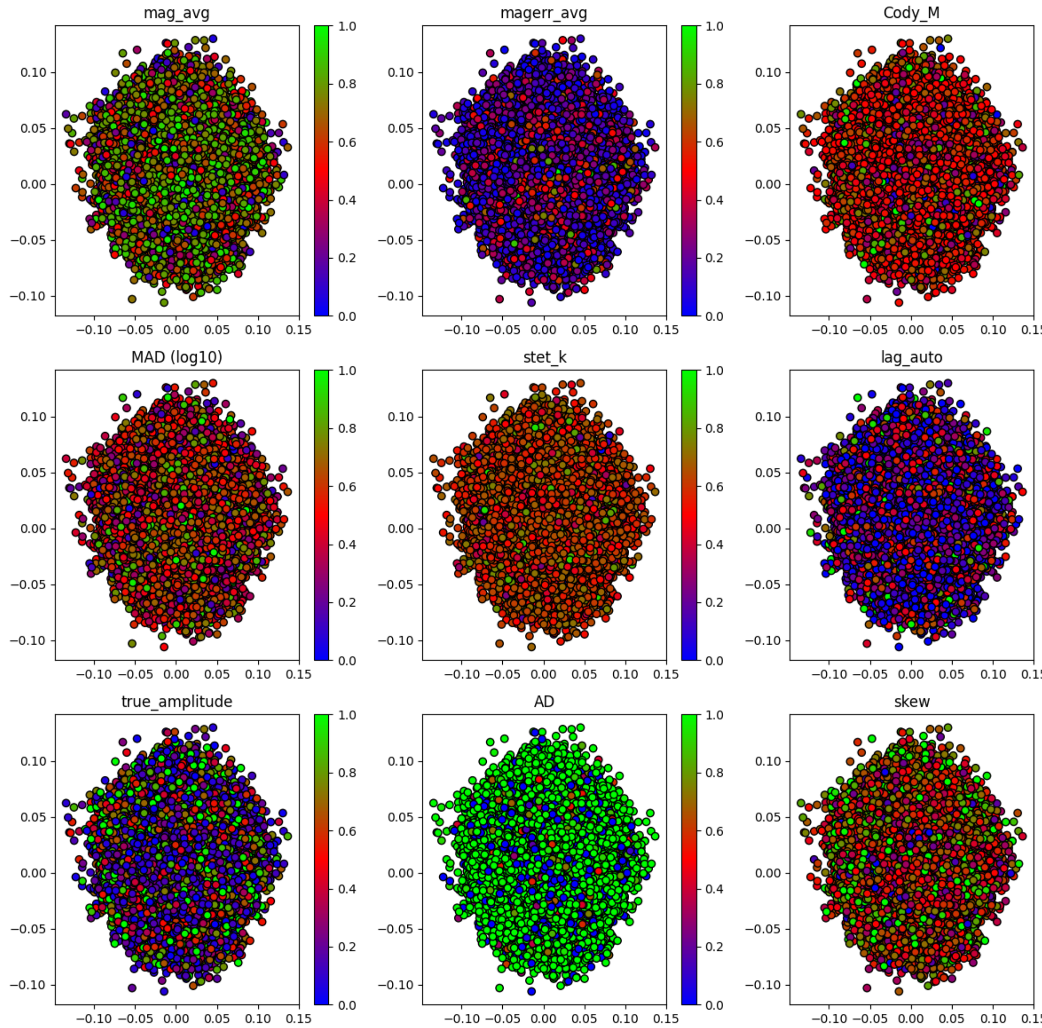


FIGURE 2.5: A PCA representation of the latent space trained with; learning rate = 0.001, tau = 0.01 and gamma = 0.7

be an effective technique for destroying highly correlated relationships between neurons(Hinton et al., 2012). The hidden dimensions (amount of neural layers between input and output) were chosen as the smallest power of 2 which did not visually impact latent space clustering.

During the first grid search values of Learning rate = 0.01, Tau = 0.5 and Gamma = 0.9 were used. The second grid search 'Grid Search #2' was used to determine the hyperparameters used for the training loop. Where 'Tau' is the 'temperature parameter' from the NT-Xent loss, it is analogous to the learning rate with a larger value amplifying gradients through the network. 'Gamma' determines the rate at which the learning rate decays during training. The number of output dimensions was determined as the final parameter. This value was iteratively halved from 256 until the PCA and UMAP representations of both hidden states (h) and latent representations (z) noticeably declined in complexity. Interestingly, this appeared to be 64 dimensions, the same as the hidden dimensions.

Chapter 3

Results

For the classification of stellar light curves, the utilisation of both hidden states (h) and latent representations (z) proves to be advantageous. Hidden states encapsulate the temporal dynamics inherent to light curves. This is crucial for capturing patterns such as periodicity and trends over time. Latent representations are hidden states passed through a feed-forward network, in our case a projection layer consisting of linear transformations, `LeakyReLU` activation, and dropout regularisation. Latent representations offer a condensed version of the input data, with the goal of emphasising the key features that are essential for classification.

Given the distinct characteristics of stellar light curves, the combined use of h and z can significantly improve model efficacy. h leverages the sequential nature of the data to capture dynamic changes, while z distills this information into a feature-rich representation ideal for classification.

Figures 3.1,3.2,3.3, and 3.4 show the PCA representation of the hidden states with the colour axis representing different features from the PRIMVS catalogue.

Figure 3.1 shows the PCA representation of the hidden states as a function of VVV colours. There is a slight indication of a correlation with clumps of colours loosely forming. This is a weak correlation however and the correlation is dominated by noise. The plot demonstrates the model's potential to identify previously unlabelled stellar classes that feature a correlation with VVV colours, despite this colour information not being a factor in the latent space construction. This ability to correlate with known physical properties, despite the model not being explicitly trained on them, hints at a correlation with stellar class. This suggests that the clustering based on light curve morphology is likely real because morphology is related to class and, independently, colour is also related to class.

Figure 3.2 shows the PCA representation of the hidden states as a function of basic statistics for the light curves: skewness, kurtosis (in $\log_{10}$), amplitude, and period. We observe a strong

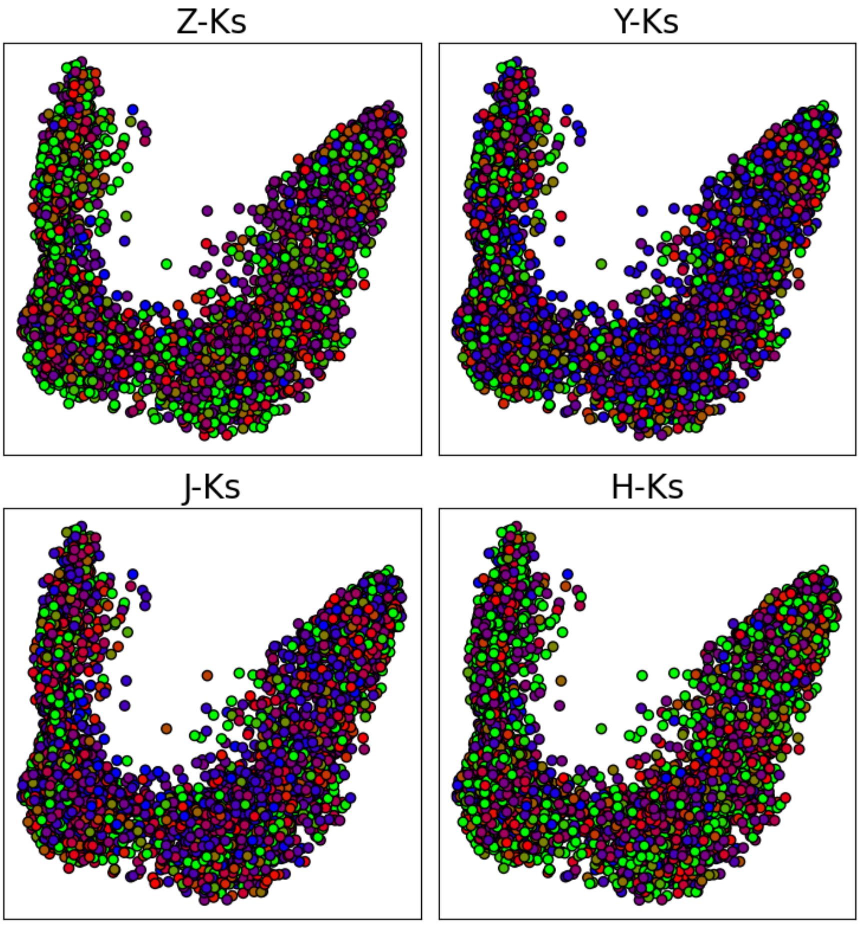


FIGURE 3.1: A 2 dimensional PCA representation of the hidden states (h) colour coded with respect to VVV colours. The plot demonstrates the ability of the model to identify previously unlabelled stellar classes which feature a correlation with VVV colours, despite this colour info not being a factor of the latent space construction.

correlation with skewness, indicating that the contrastive curves method is effectively creating representations based on the shape of the light curve. This is a crucial validation that our model is sensitive to morphological features, which are essential for distinguishing different types of variable stars. In contrast, we do not see a strong correlation with kurtosis, which may be due to the challenging nature of representing the distribution shape accurately. Kurtosis measures the tails of the distribution, and its weaker correlation might indicate that the model does not prioritise these features as strongly as skewness or other statistics. Encouragingly, there is a visible correlation with both amplitude and period, which are commonly used together to form a Bailey diagram to aid in stellar classification. This suggests that the model captures key periodic characteristics and the extent of brightness variation in the light curves. The independence of these correlations from skewness further supports the robustness of the model, demonstrating its ability to consider multiple dimensions of variability simultaneously. This independence hints at the deeper complexity of the latent space representations, indicating that the model can discern and encode various aspects of the light curves. By clustering light curves based on

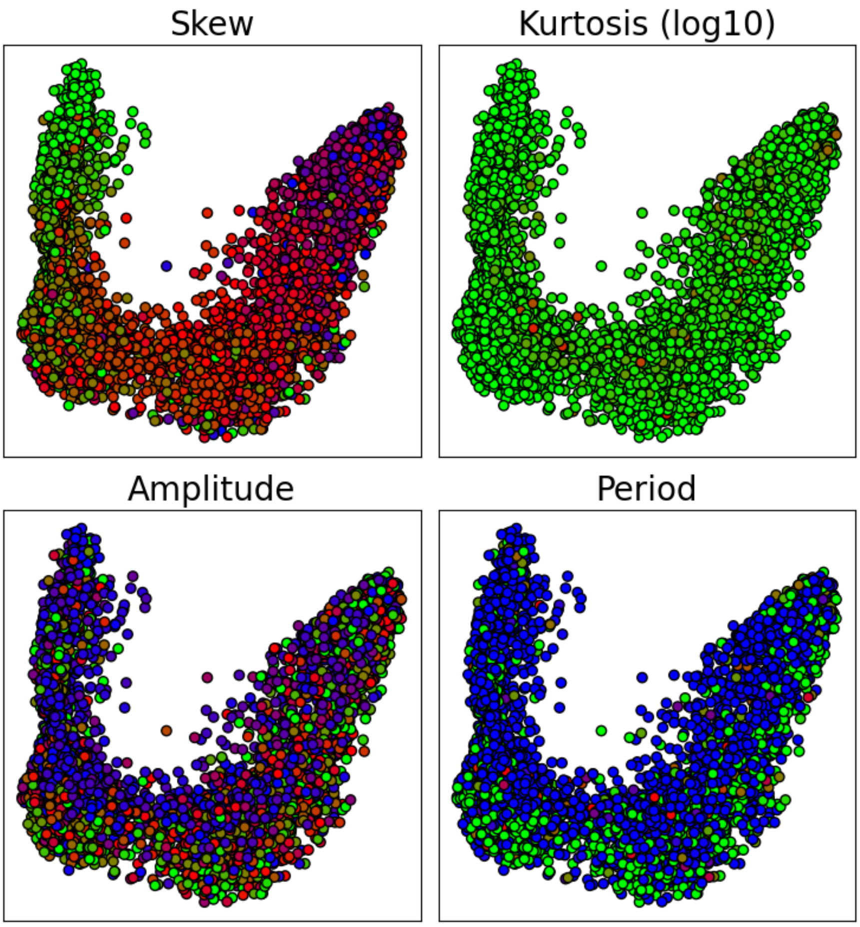


FIGURE 3.2: A 2 dimensional PCA representation of the hidden states (h) colour coded with respect to Skew, Kurtosis, Amplitude and Period (from top left to bottom right).

shape, amplitude, and period without explicit input on these statistics, the model demonstrates its capability to uncover intrinsic patterns in the data. This alignment with known statistical properties underscores the reliability of the model's representations, providing confidence in its application to other astronomical datasets.

Figure 3.3 shows the PCA representations as a function of more statistical features extracted from the light curve. We see a relatively strong correlation with every feature shown. A.M Cody's 'M' value measures the asymmetry in the light curve, with higher values indicating more pronounced asymmetry, a key parameter for identifying eclipsing binaries. The median buffer range percentage captures the proportion of points within a small range around the median magnitude. The range of cumulative sum assesses the overall variability, with larger values indicating greater variability. The maximum slope identifies the steepest change in brightness, highlighting rapid variability. The independent correlations observed in the PCA representations further highlight the model's ability to capture more nuanced details of light curve variability.

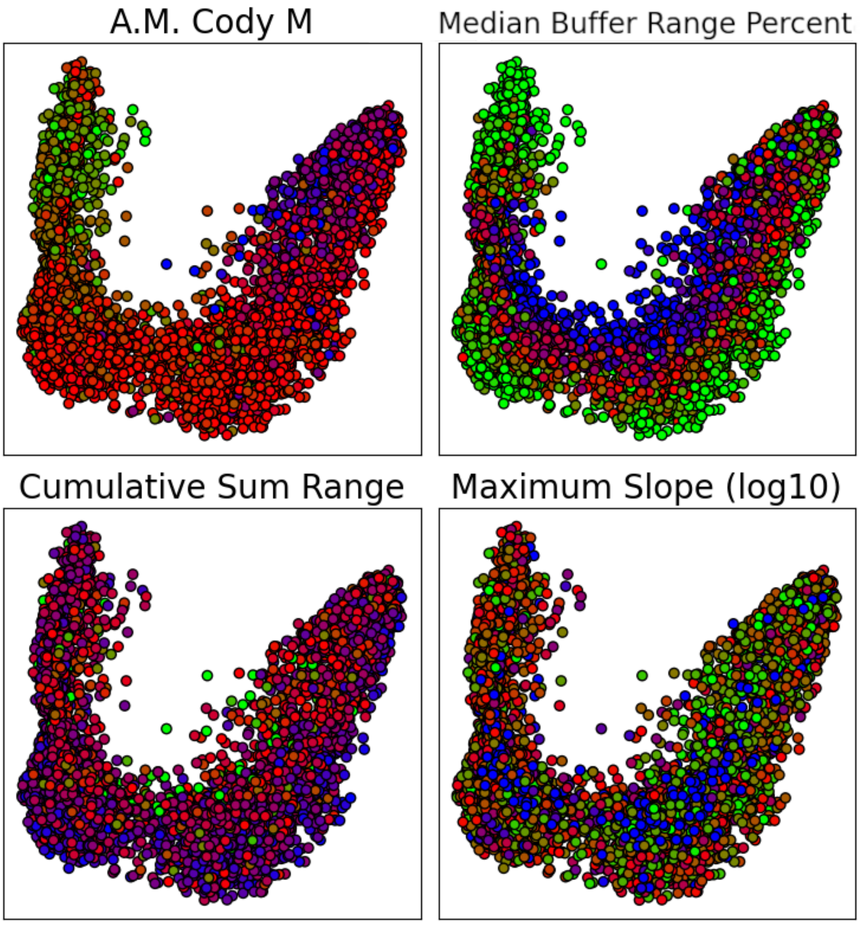


FIGURE 3.3: A 2 dimensional PCA representation of the hidden states (h) colour coded with respect to 'M', the median buffer range percentage, the range of cumulative sum, and the maximum slope (from top left to bottom right).

Figure 3.4 shows the PCA representations as a function of features that might indicate data quality or non-desirable correlations. Ideally, the latent space should exhibit minimal dependence on factors such as Galactic position, apparent magnitude, or photometric uncertainty. In these representations, we observe no clear correlation with Galactic position, indicating that the model is effectively capturing intrinsic properties of the light curves rather than spatial biases. However, a correlation with both magnitude and its associated error is evident. The augmentations used in this method aim to mitigate these dependencies, but they are fundamentally constrained by the assumption that photometric uncertainty is uncorrelated scatter. This assumption is likely not always true, leading to residual correlations. Addressing this limitation, future iterations of this work should focus on developing more sophisticated models for photometric uncertainty. Such models would account for correlated noise and other systematic effects, enhancing the robustness of the latent space representations by removing unwanted dependencies.

Figure 3.5 shows the PCA representation of the latent representations, z. It can be seen that these representations provide less useful representations of the latent space. This is likely because the

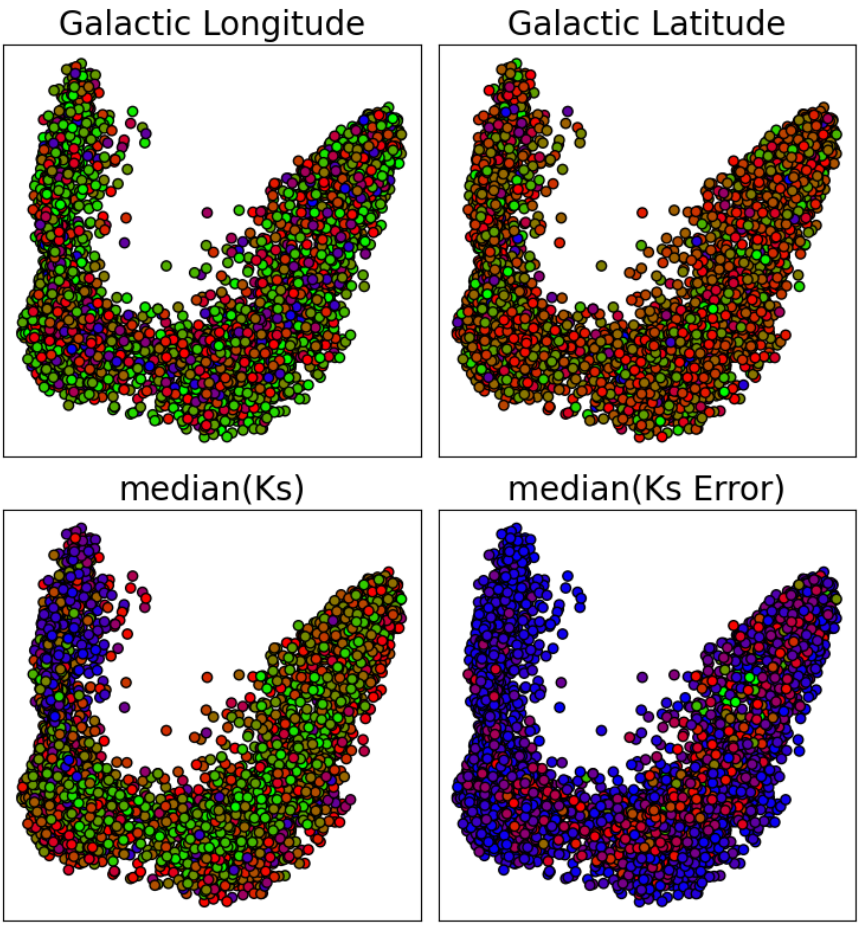


FIGURE 3.4: A 2 dimensional PCA representation of the hidden states (h) colour coded with respect to Galactic latitude, longitude, madian magnitude and median magnitude error (from top left to bottom right).

latent representations are the hidden states which have been passed through an auto-encoder. Ablative testing has shown 64 dimensions to be the minimum at which the auto-encoder still preserves all apparent semantically useful information. It makes sense that further reducing this information via PCA is not useful. This does not mean the latent representations are useless; rather, they are hard to properly visualise in lower dimensions.

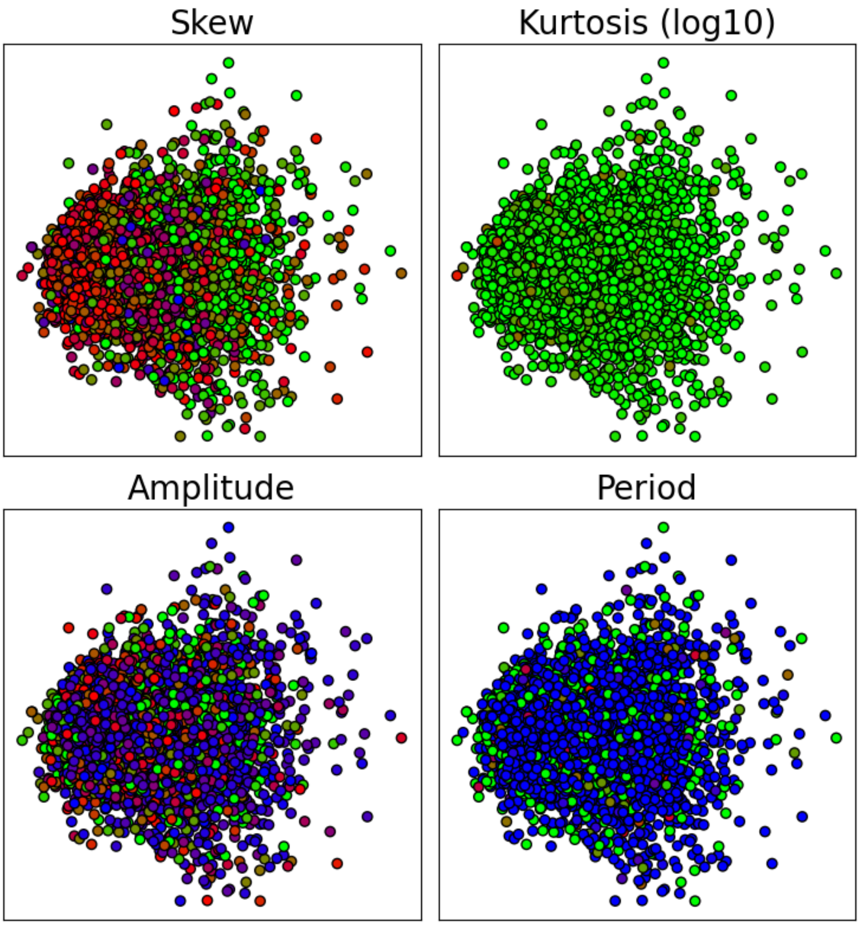


FIGURE 3.5: A 2 dimensional PCA representation of the latent representation (z) colour coded with respect to Skew, Kurtosis, Amplitude and Period (from top left to bottom right).

3.1 Latent Space Exploration

We can use the classifications obtained in via Gaia as a way of verifying this method. A decision tree based is employed on a Gaia training set and extracted astrometric, colour and time series features. This means that it is almost entirely independent of the contrastive curves method. Figure 3.6 shows two views of the 3 dimensional PCA representation of the hidden states. A clear separation can be seen between stellar class predicted by the decision tree in the latent space. This further suggests that we are indeed generating semantically meaningful representations of phase folded light curves.

Given that we see distinct clusters of labelled classes in figure 3.6, we can look for unlabelled objects within these clusters. We are effectively using these externally labelled classes to trace the classes from our contrastive curves representations. We expect the unlabelled objects within each cluster to be of the class that the cluster represents. The centre of each cluster is defined as the mean value (in terms of the 3-dimensional latent space projection) of all sources with a probability > 0.7, entropy < 0.2, and confidence metric > 0.9. Figures 3.7, 3.8, 3.9 and, 3.10 show the phase folded light curves of the 4 nearest neighbours to the centre of the Eclipsing Binary, RR Lyrae, Cepheid, and Ellipsoidal clusters, respectively. These figures highlight the ability of the model to accurately classify and identify new members of these stellar classes.

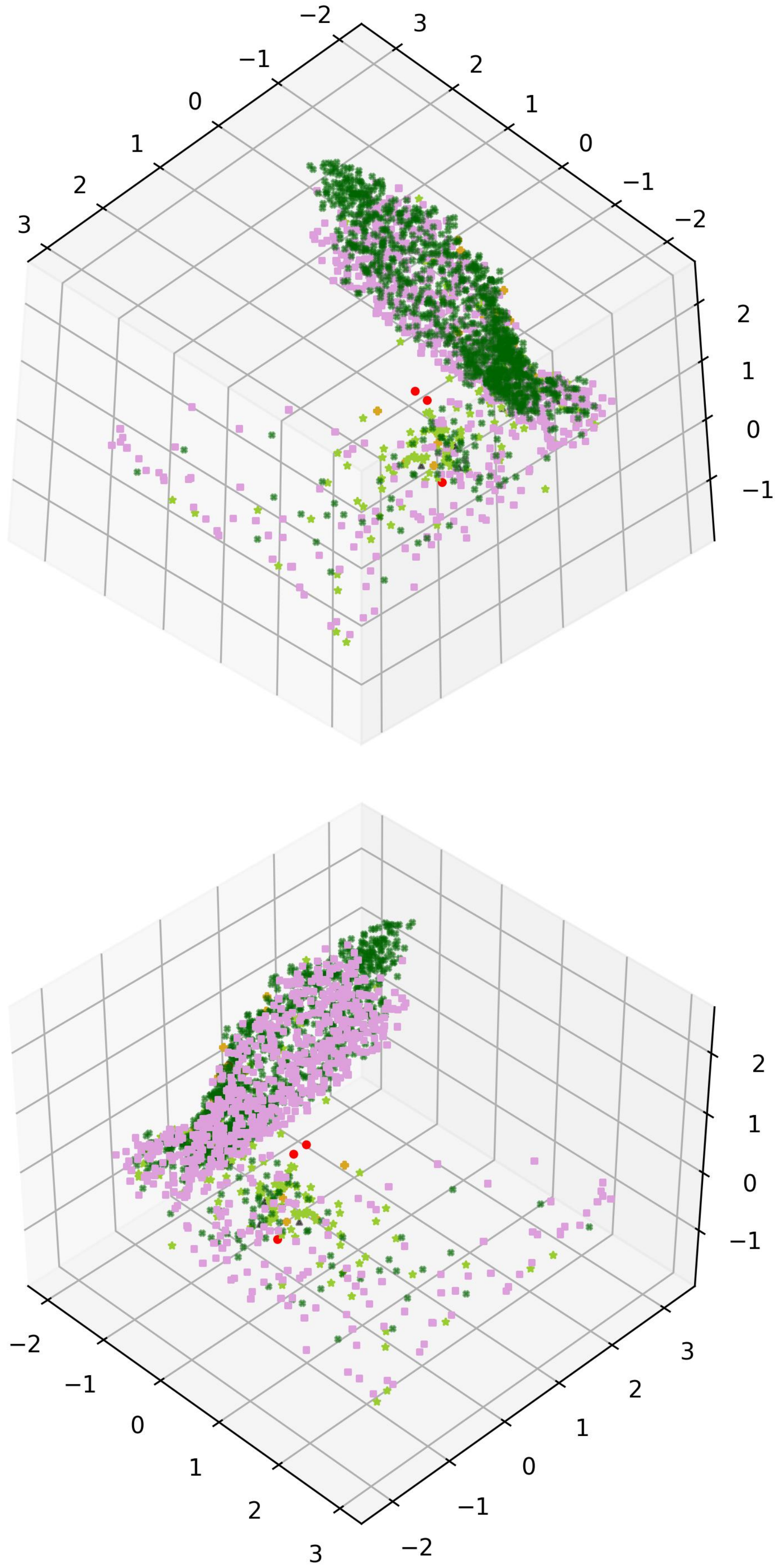


FIGURE 3.6: Two different points of view for the 3 dimensional PCA representation of the hidden states (h). The dark green is eclipsing binary, pink is long-period variables, yellow is rr lyrae, red is cepheids, and light green is ellipsoidals.

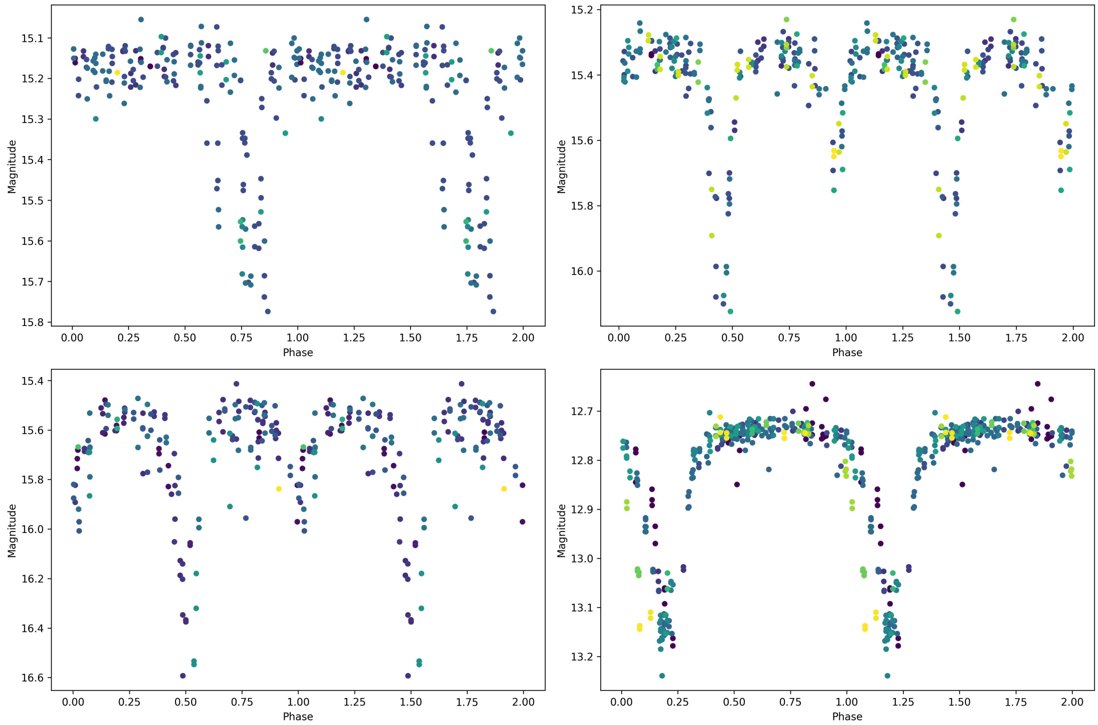


FIGURE 3.7: The phase folded light curves of the 4 nearest neighbours to the centre of the Eclipsing Binary cluster.

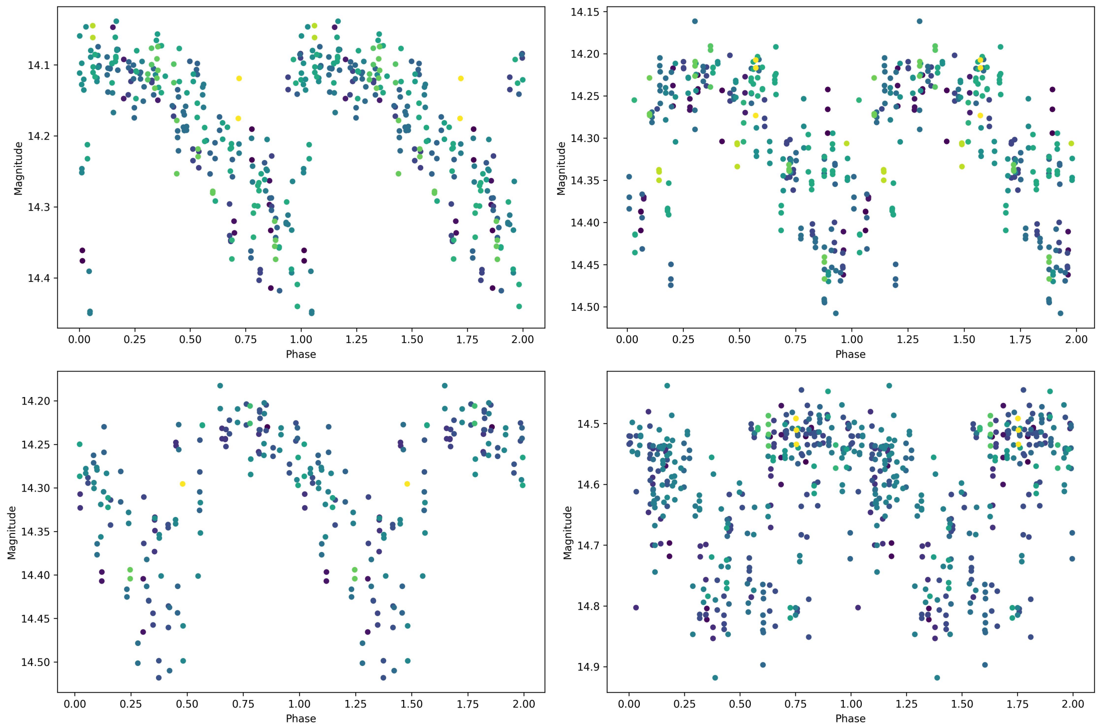


FIGURE 3.8: The phase folded light curves of the 4 nearest neighbours to the centre of the RR Lyrae cluster.

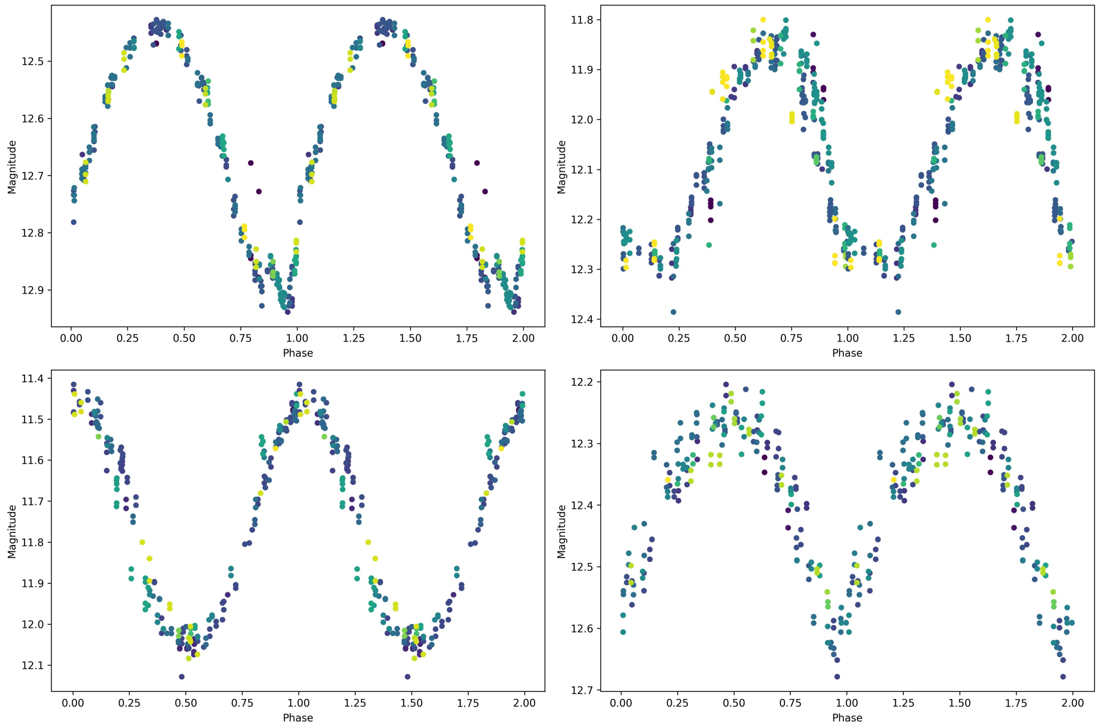


FIGURE 3.9: The phase folded light curves of the 4 nearest neighbours to the centre of the Cepheid cluster.

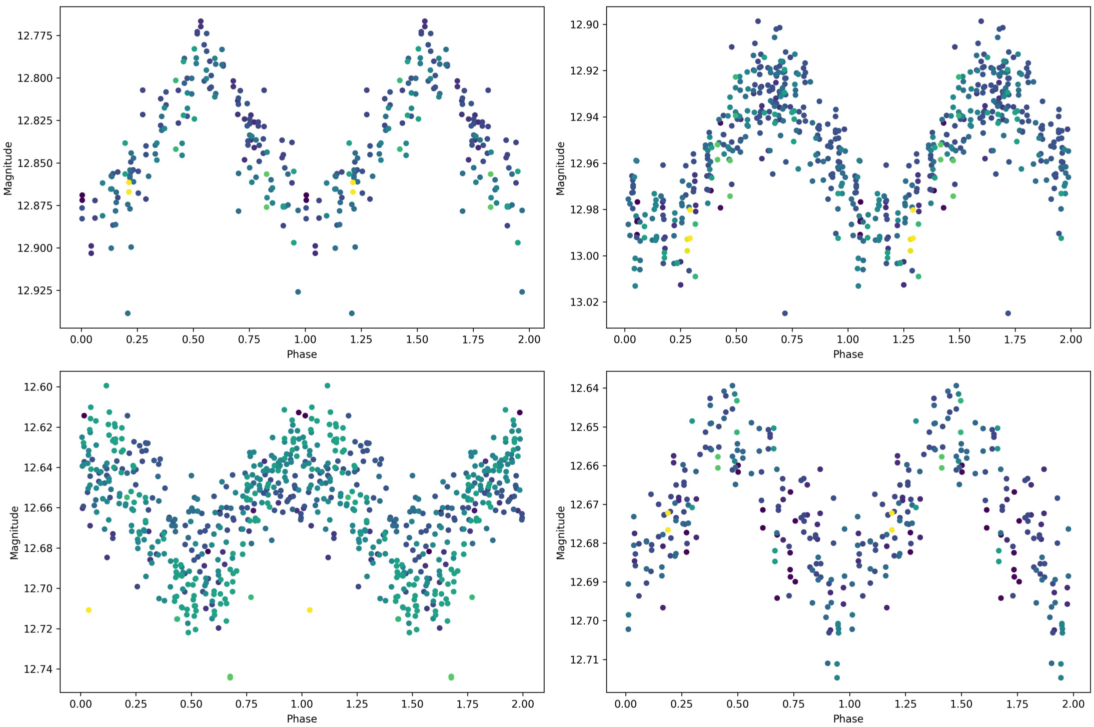


FIGURE 3.10: The phase folded light curves of the 4 nearest neighbours to the centre of the Ellipsoidal cluster.

Chapter 4

Conclusions

The paper demonstrates the effective use of a novel neural network architecture, which leverages contrastive learning with a gated recurrent neural network backbone, to generate semantically meaningful representations of stochastically sampled time series data. This approach proves to be effective for analysing astronomical time series data, capturing the complex, dynamic behaviours characteristic of variable stars.

Training the model presented unique challenges, primarily due to the novelty of the approach and the complexity of the time series data. Through extensive experimentation, including two rounds of grid searches, some optimal hyperparameters were identified.

The analysis presented here is by no means exhaustive and only seeks to prove the efficacy of this method. We have shown with figures 3.6- 3.10 that there is good agreement between the Gaia trained decision tree based classification and the contrastive curves method. It follows that using the output of contrastive curves in the decision tree would likely improve classification accuracy.

This work showcases the potential of contrastive learning models to revolutionise our understanding of astronomical time series data. By effectively capturing the essence of variable stars in fixed-length embeddings, this approach opens up new possibilities for automated classification and analysis in the era of Big Data in astronomy.

Bibliography

Abul-Hayat, M., Stein, G., Harrington, P., et al., 2020. Self-Supervised Representation Learning for Astronomical Images. *arXiv e-prints*, arXiv:2012.13083.

Anderson, T.W. and Darling, D.A., 1952. Asymptotic Theory of Certain "Goodness of Fit" Criteria Based on Stochastic Processes. *The Annals of Mathematical Statistics*, 23(2):193 .

BAART, M.L., 1982. The Use of Auto-correlation for Pseudo-rank Determination in Noisy III-conditioned Linear Least-squares Problems. *IMA Journal of Numerical Analysis*, 2(2):241.

Belokurov, V., Evans, N.W., and Du, Y.L., 2003. Light-curve classification in massive variability surveys - I. Microlensing. *Monthly Notices of the Royal Astronomical Society*, 341(4):1373.

Bundy, K., Bershady, M.A., Law, D.R., et al., 2015. Overview of the SDSS-IV MaNGA Survey: Mapping nearby Galaxies at Apache Point Observatory. *ApJ*, 798(1):7.

Charnock, T. and Moss, A., 2017. Deep Recurrent Neural Networks for Supernovae Classification. *The Astrophysical Journal Letters*, 837(2):L28.

Chen, T., Kornblith, S., Norouzi, M., et al., 2020. A Simple Framework for Contrastive Learning of Visual Representations. *arXiv e-prints*, arXiv:2002.05709.

Chen, T., Kornblith, S., Swersky, K., et al., 2020a. Big self-supervised models are strong semi-supervised learners. *CoRR*, abs/2006.10029.

Chen, X., Fan, H., Girshick, R.B., et al., 2020b. Improved baselines with momentum contrastive learning. *CoRR*, abs/2003.04297.

Cho, K., van Merrienboer, B., Bahdanau, D., et al., 2014. On the properties of neural machine translation: Encoder-decoder approaches. *CoRR*, abs/1409.1259.

Cody, A.M., Stauffer, J., Baglin, A., et al., 2014. CSI 2264: Simultaneous Optical and Infrared Light Curves of Young Disk-bearing Stars in NGC 2264 with CoRoT and Spitzer—Evidence for Multiple Origins of Variability. *AJ*, 147(4):82.

Deb, S. and Singh, H.P., 2009. Light curve analysis of variable stars using Fourier decomposition and principal component analysis. *A&A*, 507(3):1729.

Devlin, J., Chang, M.W., Lee, K., et al., 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics, Minneapolis, Minnesota.

Dewdney, P.E., Hall, P.J., Schilizzi, R.T., et al., 2009. The Square Kilometre Array. *IEEE Proceedings*, 97(8):1482.

Donoso-Oliva, C., Becker, I., Protopapas, P., et al., 2022. ASTROMER: A transformer-based embedding for the representation of light curves. *arXiv e-prints*, arXiv:2205.01677.

F.R.S., K.P., 1901. Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559.

Grill, J., Strub, F., Altché, F., et al., 2020. Bootstrap your own latent: A new approach to self-supervised learning. *CoRR*, abs/2006.07733.

He, K., Fan, H., Wu, Y., et al., 2019. Momentum contrast for unsupervised visual representation learning. *CoRR*, abs/1911.05722.

Hinton, G.E., Srivastava, N., Krizhevsky, A., et al., 2012. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv e-prints*, arXiv:1207.0580.

Ivezić, Ž., Kahn, S.M., Tyson, J.A., et al., 2019. LSST: From Science Drivers to Reference Design and Anticipated Data Products. *ApJ*, 873(2):111.

McCulloch, W.S. and Pitts, W., 1943. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115.

McInnes, L., Healy, J., and Melville, J., 2018. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. *arXiv e-prints*, arXiv:1802.03426.

Minniti, D., Lucas, P., Emerson, J., et al., 2010. Vista variables in the via lactea (vvv): The public eso near-ir variability survey of the milky way. *New Astronomy*, 15(5):433.

Paszke, A., Gross, S., Massa, F., et al., 2019. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

Sarmiento, R., Huertas-Company, M., Knapen, J.H., et al., 2021. Capturing the physics of manga galaxies with self-supervised machine learning. *The Astrophysical Journal*, 921(2):177.

Sohn, K., 2016. Improved deep metric learning with multi-class n-pair loss objective. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29 of *Advances in Neural Information Processing Systems*. Curran Associates, Inc.

Srivastava, N., Hinton, G., Krizhevsky, A., et al., 2014. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929.

Stetson, P.B., 1996. On the Automatic Determination of Light-Curve Parameters for Cepheid Variables. *PASP*, 108:851.

Storrie-Lombardi, M.C., Lahav, O., Sodré, L., J., et al., 1992. Morphological Classification of galaxies by Artificial Neural Networks. *Monthly Notices of the Royal Astronomical Society*, 259(1):8P.

Vaswani, A., Shazeer, N., Parmar, N., et al., 2017. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6000–6010. Curran Associates Inc., Red Hook, NY, USA.

York, D.G., Adelman, J., Anderson, J. E., J., et al., 2000. The Sloan Digital Sky Survey: Technical Summary. *AJ*, 120(3):1579.