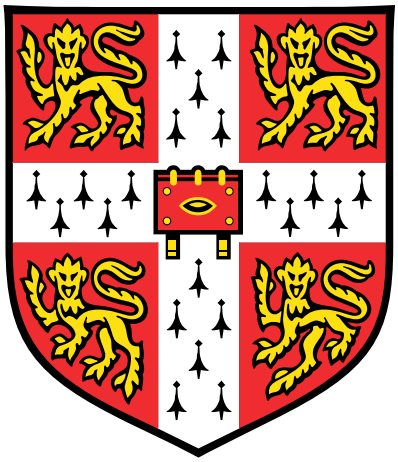
MPhil Data Intensive Science
University of Cambridge

AstroCLIP: Cross-Modal Pre-training for Astronomical Foundation Models

Reproduction and Extension

Data Analysis Project



Andreas Vrikkis

June 30, 2024

LaTeX Word count: 6105

ABSTRACT

We present a reproduction and extension of *AstroCLIP*: a powerful model that facilitates the construction of both galaxy images and spectra into a shared, cross-modal embedding space. This is achieved by fine-tuning a pair of pre-trained single-modal models under a contrastive learning framework. Despite using a smaller spectrum encoder with 2 orders of magnitude fewer parameters than the original work, we achieve similar performance on a range of downstream tasks. We apply the model to extract embeddings from the Dark Energy Spectroscopic Instrument (DESI) images and from its corresponding Legacy Imaging Survey spectra. We demonstrate that the embedding space is effective at (1) accurate semantic similarity search for both in-modality and cross-modality query-retrieval tasks and (2) zero-shot prediction of redshift and stellar mass, achieving a Spearman's rank correlation coefficient R^2 of up to 0.87. We then extend the original work by projecting the embeddings space onto a 2D plane using UMAP, and use k-Means to formulate clusters of galaxies that are semantically separable in both modalities. Ultimately, our approach demonstrates the ability of AstroCLIP to yield highly informative embeddings that have the emergent property of aligning themselves based on shared semantics and physical galaxy properties.

Contents

1 Introduction

In recent years, the field of astronomy has witnessed a significant surge in the volume and complexity of datasets. Large-scale sky surveys, such as the Dark Energy Spectroscopic Instrument (DESI) [1], have generated catalogues encompassing tens of millions of objects, with future projects targeting the inclusion of billions. As the data volume expands, so does the complexity of the required analysis, necessitating sophisticated methods by researchers to extract meaningful insights. This has led to a growing interest in leveraging data-driven methodologies, particularly machine learning (ML), to aid in the analysis of these extensive astronomical datasets [2].

Despite the diverse range of computational approaches that have been developed, most ML models in astronomy are trained using a **supervised** learning framework. These models fundamentally depend on the quality and quantity of labelled training data to identify underlying patterns. However, in the field of astronomy, labelled data are often scarce and costly to obtain, primarily relying on crowd-sourced efforts [3]. Additionally, supervised models require retraining for each new task, starting with essentially random initialisation of network weights and parameters (random priors). This necessity poses a significant limitation, constraining the scalability and generalisation of these models to novel tasks.

Recently, a new line of research has shifted towards self-supervised learning (SSL) as an alternative approach. These methods learn high-quality, rich **embeddings**: low-dimensional vector representations that capture semantically meaningful information of objects without the need for labeled training data. This is achieved by training models to perform surrogate tasks, such as identifying corrupted pairs or filling in masked sections of the input data. The extracted embeddings are versatile and can be utilised for various downstream tasks such as similarity searches, clustering and outlier removal. More importantly, they can be used as a 'foundation' for further downstream tasks, hence they are often referred to as **foundation models**. Approaches of this kind have already revolutionised ML fields, such as computer vision [4] and natural language processing [5], where they have demonstrated superior performance compared to their supervised counterparts in zero-shot and few-shot learning tasks.

Numerous SSL approaches have been deployed in the field of observational astronomy, with a notable example being the application of the Momentum Contrastive pretraining strategy (MoCo v2) to galaxy images [6]. However, these approaches have primarily been limited to embedding objects of a single modality, typically by creating augmented views of the same underlying object. Nevertheless, observational astronomy is inherently **multimodal**, as the same object can yield a variety of complementary observations. For instance, a galaxy can be observed through both images and spectra, each providing different data structures of the same underlying object. Since these modalities are correlated transformations of the same object, a common embedding space should exist where the representations of these modalities are aligned. Establishing such a universal foundation model for observational astronomy would enable cross-modal tasks such as searches, clustering, and inference.

In this work, we reproduce and extend *AstroCLIP* - the first cross-modal foundation model for galaxies. Specifically, we aim to reproduce the results of version 1 of the AstroCLIP paper [7], which we refer to as the 'original work'. Our reproduction focuses on the following tasks:

- Deploy pre-trained single-modal models for galaxy images and spectra for the DESI Legacy Survey [1].
- Train a cross-modal model under a contrastive learning framework to align the embeddings of the two modalities in a shared latent space.
- Reproduce the original work's model performance on a range of downstream tasks, including query retrieval, zero-shot prediction of physical properties.

We then extend the original work by further analysing the embedding space structure using clustering algorithms and dimensionality reduction techniques. In doing so, we empirically demonstrate that our cross-modal embeddings can capture semantically meaningful information about galaxies, which are aligned in both modalities.

ML research is currently facing a reproducibility crisis, with many studies proving difficult or even impossible to replicate [8]. This often arises due to unpublished data or source code, or due to subtle sensitivities in the training process that are not well-documented. Furthermore, ML code tends to be complex and lacks the standardisation seen in other scientific fields. As the paradigm of predictive modelling using ML gains traction, it is more crucial than ever to undertake reproducibility studies like this one. Here, we discuss the various challenges encountered during the reproduction process and the steps taken to overcome them.

2 Methodology

2.1 Transfer Learning & Self-Supervised Learning

When labelled training data are scarce, other datasets can be exploited to improve performance. In *transfer learning* a model is first pre-trained to perform a related secondary task for which we have more (potentially labelled) data [9]. The resulting model is then adapted to the primary task of interest. This is typically done by removing the final layer(s) of the network and adding new layers (heads) that produce the desired output. Further training is then performed on the primary task. The pre-trained part of the network can be frozen (i.e., its weights are not updated during further training) or it can be fine-tuned. The key principle behind this approach is that the pre-trained model has built a good internal representation of the data from the secondary task, which can be useful for the primary task. It can also be seen as a form of sensible weight initialisation for most of the final network.

For transfer learning to be effective, the secondary task should contain a large amount of data. It is often the case, however, than even the secondary task has no labels. This is where self-supervised learning (SSL) has seen a wide use, in which the secondary task is a pretext task where the labels are generated from the data itself. In the process, the model learns to extract rich, low-dimensional representations from data without the need for human labelling. The pretext task is often chosen to be an artificial surrogate task on the input data. Recently, numerous such tasks have been developed, including autoregressive prediction of the next word in a sequence [5], masked language modelling [10] and contrastive learning [11]. Adding to this, pretext tasks can be of different types; for example, a model can be made to predict the rotation angle of a given image [12]. These techniques have shown success in generating versatile and informative representations across natural language processing and computer vision.

2.2 Contrastive Learning

Contrastive learning is a self-supervised learning (SSL) technique that derives meaningful representations from unlabelled datasets. The fundamental concept of contrastive learning involves distinguishing between semantically similar (positive) and dissimilar (negative) pairs of data points. This technique encourages the representations of positive pairs to be closely aligned in the embedding space, while those of negative pairs to be more orthogonal [13]. This process be done in either a single-modal or multi-modal (cross-modal) setting, where 'modality' refers to the type of data input, such as images or text. In single-modal contrastive learning, stochastic data augmentation is applied to a data example to create two correlated views of the same instance [12]. These views form a positive pair, whereas all other examples constitute negative pairs.

We can extend this concept to cross-modal contrastive learning by considering that different observational modalities provide correlated views of the same underlying physical object. Consequently, instead of artificially generating positive and negative pairs through stochastic augmentation, embeddings from different modalities of the same object are inherently positive pairs, while all other embeddings are negative pairs. Separate embedder networks are employed for each modality to create these embeddings, which are then jointly trained using a contrastive loss function. A notable example of aligning embeddings across modalities is Contrastive Language-Image Pre-training (CLIP) [11], which unifies images and text within a shared embedding space.

The cross-modal framework is illustrated in Fig.1. Here, $\mathbf{x}_i \in \mathbb{R}^N$ and $\mathbf{y}_i \in \mathbb{R}^M$ represent two modalities of the same object i. These modalities are processed through encoder networks $f_\theta : \mathbb{R}^N \to \mathbb{R}^D$ and $g_\phi : \mathbb{R}^M \to \mathbb{R}^D$, with $\{\theta, \phi\}$ as trainable parameters. The encoders extract representation vectors in a shared embedding space, where contrastive loss is applied. These representation vectors are denoted as $\mathbf{z}_i^x \in \mathbb{R}^D$ and $\mathbf{z}_i^y \in \mathbb{R}^D$, with D being the dimensionality of the embedding space and the superscripts x and y indicating the originating modality. While not strictly necessary, it was shown in Ref.[14] that by augmenting the data before passing it through the encoder with a stochastic data augmentation $\mathcal{T}$, the model can learn more robust representations.

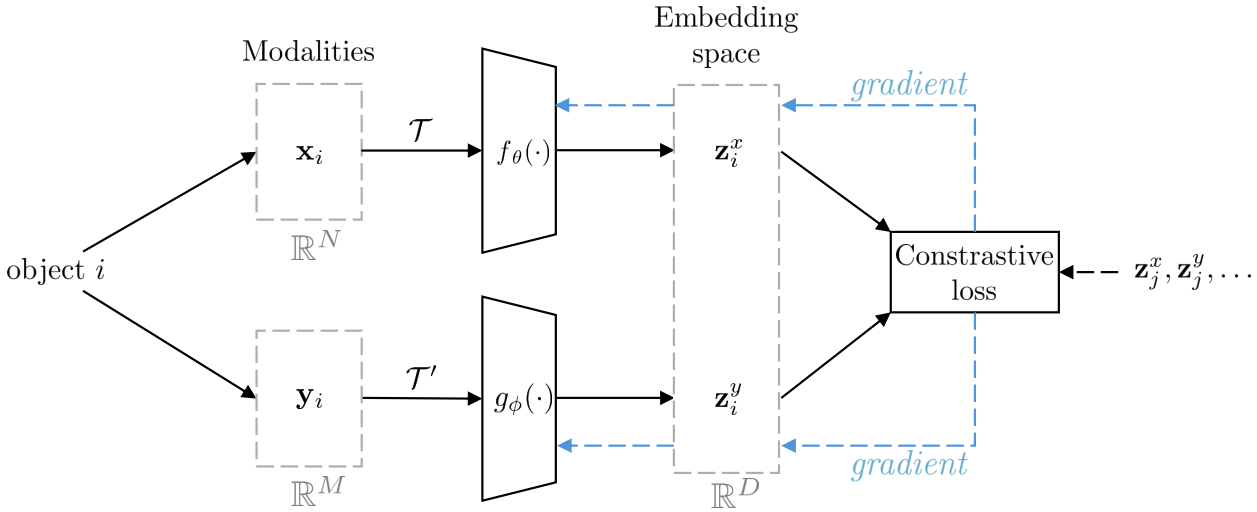


Figure 1: Cross-modal contrastive learning framework. An underlying physical object i is observed in two different modalities, $\mathbf{x}_i \in \mathbb{R}^{\mathbb{N}}$ and $\mathbf{y}_i \in \mathbb{R}^{\mathbb{M}}$. An augmented version of each modality undergoes a transformation $\mathcal{T}, \mathcal{T}'$ and are passed through encoder networks f_θ, g_ϕ which compress them into representations $\mathbf{z}_i^x, \mathbf{z}_i^y$ in a shared embedding space. The contrastive loss (InfoNCE) is applied to these representations along with the negative pairs denoted by j, with the gradients backpropagated to jointly train the encoder networks.

We want this embedding space to maximise the mutual information $I(f_\theta(\mathbf{x}), g_\phi(\mathbf{y}))$ between these two representations. However, calculating the mutual information directly is intractable for finite data [15]. Instead, we approximate each modality as a noisy transformation of the same underlying object and use an Information Noise-Contrastive Estimation (InfoNCE) loss function [16] which maximises a variational bound on the mutual information. The InfoNCE loss is defined as:

$$\mathcal{L}_{\text{InfoNCE}}(\mathbf{z}^x, \mathbf{z}^y, \tau) = -\frac{1}{K}\sum_{i=1}^{K} \log \frac{\exp\left(S_C(\mathbf{z}_i^x, \mathbf{z}_i^y)/\tau\right)}{\sum_j^K \exp\left(S_C(\mathbf{z}_i^x, \mathbf{z}_j^y)/\tau\right)} \tag{1}$$

where $\mathbf{z}_i^x = f(\mathbf{x}_i)$ and $\mathbf{z}_i^y = g(\mathbf{y}_i)$, $\tau > 0$ denotes a smoothing parameter (referred to as temperature) , $S_C(\mathbf{z}_i^x, \mathbf{z}_i^y)$ is a similarity metric between the two representations, and j represent the indices of negative examples (i.e., representations of different objects to object i). For the similarity metric in CLIP, we use the cosine similarity given by:

$$S_C(\mathbf{z}_i^x, \mathbf{z}_j^y) = \frac{(\mathbf{z}_i^x)^T \mathbf{z}_j^y}{\|\mathbf{z}_i^x\|^2 \|\mathbf{z}_j^y\|^2}. \tag{2}$$

InfoNCE is biased, but it represents a stable, low variance bound on the mutual information that is widely used in contrastive methods [14].

Training cross-modal models from scratch under CLIP loss has been shown to often underperform compared to single-modal problems. However, by leveraging transfer learning through the addition of pre-trained single-modal models as initialisation, we can significantly improve performance. In this work, we utilise this approach by deploying pre-trained single-modal models which we appended with a projection head that is trained under the InfoNCE loss.

2.3 Embedding Space Clustering

Since the objective of InfoNCE is to bring the embeddings of the same object close together while pushing embeddings from different objects apart, it results in a structured space where semantically similar objects are proximal, while semantically dissimilar objects are more distant. Thus, the embedding space can be viewed as a multi-dimensional space of point-objects, where the distances between them carry semantic meaning. Consequently, these embeddings are inherently useful for a variety of tasks, referred to as *downstream tasks*, which happen after the training process. Downstream tasks on the embedding space can include similarity searches or even the prediction of physical properties of the objects based solely on the spatial structure. These are known as *zero-shot* predictions, as the model has not been explicitly trained on the specific task.

The embedding space is multi-dimensional, but can be visualised by projecting the representations to a 2D space using dimensionality reduction techniques. One such technique is UMAP (Uniform Manifold Approximation and Projection) [17], which aims to preserve the global structure and local relationships of data when applying the mapping. It does so by constructing a high-dimensional graph representation of the data, which it then optimises to create a low-dimensional projection that maintains as much of the original data's structure as possible.

The objects in the embedding space can then be grouped into clusters, with each cluster representing a group of semantically similar objects. This can be done both in the original high-dimensional space or in the low-dimensional projection. Clustering algorithms include DBSCAN (Density-Based Spatial Clustering of Applications with Noise) [18]: an unsupervised ML algorithm designed to identify clusters in large spatial datasets by examining the local density of data points. The algorithm categorises points into core points, border points, and noise points, governed by two parameters: ϵ (epsilon) and minPts (minimum points). These are defined as follows:

- **Core Points:** A point is considered a core point if it has at least minPts points within ϵ radius, including the point itself. This criterion ensures that core points are those with a high density of neighbouring points.
- **Border Points:** Border points are not core points but are located in the neighborhood of a core point. These points have fewer than minPts within their ϵ neighborhood but are reachable from core points.

- **Noise Points:** Noise points are data points that are neither core points nor border points. These points do not belong to any cluster.

DBSCAN starts by arbitrarily selecting a point and assessing whether it is a core point. If it is, the algorithm then iteratively explores and includes all directly reachable points, thereby expanding the cluster. This process continues until no new points can be added to the cluster. Points that are reachable from a core point via other core points are also included in the same cluster.

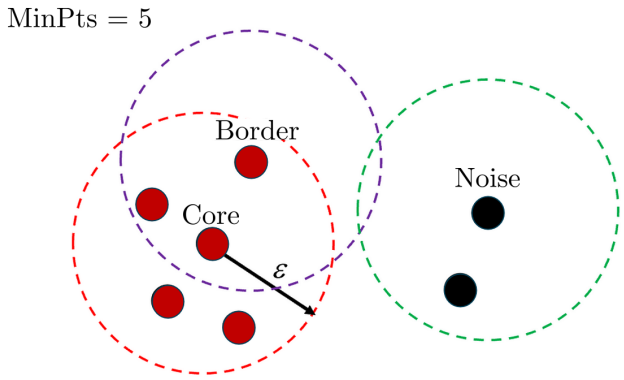


Figure 2: DBSCAN illustration: in this diagram, minPts = 5. The red points form a cluster, with 4 core points because the area surrounding these points in an ϵ radius contain at least 5 points (including the point itself). A border point is also included in the red cluster because it is reachable from a core point. The black points are noise as they are neither core points nor directly-reachable.

An illustration of the DBSCAN algorithm is shown in Fig.2. DBSCAN is particularly effective in identifying isolated collections of points ('islands') in 2D projections, such as those produced by UMAP, due to its density-based clustering approach. This method allows for the detection of irregularly shaped clusters without the need for predefined cluster numbers.

3 Implementation

In AstroCLIP, we consider two modalities of galaxies: images and spectra. We obtain their aligned embedding space in a two step process:

1. We employ two pre-trained single-modal models, one for images and one for spectra, which were initially trained using SSL, to leverage the benefits of transfer learning. For the pre-trained image embedder, we use a galaxy image encoder from Stein et al. (see Ref.[6]), based on MoCo v2 [19]. For galaxy spectra, we use the encoder part of the Spender model [20].

2. We append a simple Multilayer Perceptron (MLP) to each of the pre-trained models to compress output to $d = 128$ dimensions. We then proceed to train (or fine-tune) both MLP projection heads jointly under the InfoNCE loss, to align the embeddings of the two modalities within a shared latent space.

We keep the pre-trained encoders frozen during training and only update the MLP heads under InfoNCE loss, rather than training the entire model from scratch. This approach is consistent with previous studies showing that employing pre-trained the single-modal models significantly improves performance in cross-modal tasks [14].

It is important to note a deviation from the original work: in the original AstroCLIP (v1) paper, the authors pre-train a transformer model, structured similarly to GPT-2, to embed the

spectra [7]. This model is substantially larger than Spender, comprising approximately 43.2 million parameters. However, the transformer model is not publicly available, and the authors do not provide details on the pre-training process. Consequently, we opt to use the Spender model based on the recommendation of one of the authors. We provide details of the two models, the data used, and the training process in the following sections.

3.1 Data

3.1.1 DESI Legacy Survey Images

We use the same data as the original work. For galaxy images, we utilise the DESI Legacy Survey Data Release 9 imaging data from January 2021 [1], as prepared by Stein et al. [6]. The g and r band data for the northern galactic cap (NGC) were captured by the Beijing-Arizona Sky Survey, while the z band data were obtained from the Mayall Legacy Survey. For the southern galactic cap (SGC), the data were collected by the Dark Energy Camera Legacy Survey (DECaLS). We exclude images identified as stars by the Legacy Survey team and impose a cutoff for z-bands above 20. This results in an initial dataset of 41 million (g, r, z) images of size 256×256, which we centre-crop to 96×96 for training. The cropping is necessary because the vast majority of galaxies occupy less area than the total image size and often include overlapping regions of the sky.

3.1.2 DESI Spectra

To pair the images with spectra, we cross-match the galaxy spectra from the DESI Early Data Release [21], which contains spectra observed during the Survey Validation campaign. This cross-matching is performed using the target IDs associated with each galaxy, resulting in a total subset of 197,976 pairs of images and spectra. During the training process, we Z-score normalise each individual spectrum to have zero mean and unit variance. We then split the data into training and validation sets, with 90% of the data used for training and 10% for validation.

3.1.3 Data Catalogue for Downstream Tasks

For experiments involving the prediction of physical properties, we further cross-match the image-spectrum pairs with the PRObabilistic Value-Added Bright Galaxy Survey (PROVABGS) catalogue [22]. Specifically, we extract the estimates of the redshift Z_{HP} and stellar mass $M_\star$ for each galaxy ID present in the cross-matched catalogue. We then perform the same filtering process as outlined in the original AstroCLIP paper: selecting entries where $M_\star > 0$ and mag_g, mag_r, $\text{mag}_z > 0$, thereby removing spurious entries. This process yields a total of 105,159 entries, which we split into training and validation sets using a 90/10 ratio.

3.2 Pre-trained Image Embedder

The pre-trained image embedder used in this reconstruction was developed by Stein et al. (2021a) [6]. It is based on the MoCo v2 framework [19], utilising a ResNet50 backbone as the encoder network. The model was pre-trained on a curated subset of 3.5 million galaxies, sampled uniformly by z-band magnitude from the DESI Legacy Survey. The authors employed a single-modal SSL framework, where each image underwent multiple stochastic augmentations to create two views of the same image. These augmentations included galactic extinction, random cropping, random rotation, size scaling, point-spread function blur, jittering, and Gaussian noise addition. The model was then trained using a contrastive loss function to align the

representations of the two views in the embedding space. An overview of our image embedding procedure is shown in Fig.3.

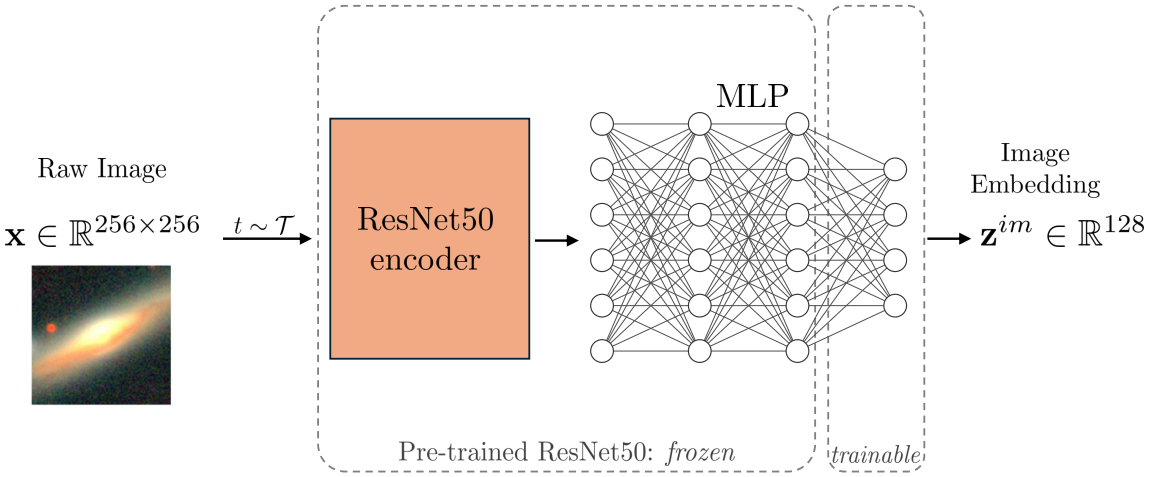


Figure 3: Overview of our image embedding procedure. Each raw galaxy image of size 256×256 undergoes an augmentation and is fed into the ResNet50 based pre-trained image encoder. During our contrastive training phase, we keep all the original weights frozen except the final layer of the MLP. After the training phase, we obtain the image embedding $\mathbf{z}^{\text{im}}$ of vector size $d = 128$ that is now in the shared embedding space.

The encoder part of the network is a ResNet50 backbone adapted to extract embeddings from astronomical images. The encoding process begins by passing the images through a series of convolutional blocks, each containing a convolutional layer, followed by batch normalisation, ReLU activation, and max pooling. Some blocks include downsampling layers to adjust the dimensions of the feature maps. The resulting embeddings are then passed through an MLP with 2048 input features and 2048 output features.

During our cross-modal contrastive training phase, we keep all encoder layers frozen except the final layer of the MLP. We adapt that final fully connected layer to output a $d = 128$-dimensional output, allowing the layer's weights and biases to be trainable under the InfoNCE loss. This results in 4.5 million trainable parameters. To enhance model performance, we artificially create more positive pairs by applying basic stochastic transformations $t \sim \mathcal{T}$ to the images. These transformations include random rotations, random horizontal and vertical flips, and random Gaussian noise.

3.3 Pre-trained Spectrum Embedder

We choose to use the Spender model [20], an autoencoder network, to extract latent parameters from the observed spectra. The overall spectrum embedder is shown in Fig.4. The encoder part of the network consists of three convolutional blocks (ConvBlocks), each containing a convolutional layer followed by trainable ReLU activation functions and max pooling. The convolutional layers have progressively wider kernel sizes (5, 11, 21), which translates the 3921 spectral components into 512 channels for 72 wavelength segments. The model then applies attention by splitting the channels into two: $\mathbf{h}$ and $\mathbf{k}$, each of dimension 256×72. These channels are then combined as follows:

$$\mathbf{e} = \mathbf{h} \cdot \text{softmax}(\mathbf{k}) \equiv \mathbf{h} \cdot \mathbf{a}, \tag{3}$$

where the dot product and the softmax are applied on the last dimension, denoting the wavelength. The attention weights in vector $\mathbf{a}$ indicate the presence and location of relevant signals, allowing the corresponding values to be enhanced into the attended features $\mathbf{e}$. This approach

efficiently accommodates the apparent shift of spectral features in galaxies at varying redshifts. The attended features are then fed into a series of fully connected layers, compressing the output into a $s = 6$ dimensions. This latent representation is subsequently passed to the decoder part of the network, which reconstructs the original spectrum. As usual in autoencoder networks, it is trained end-to-end with an MSE loss function.

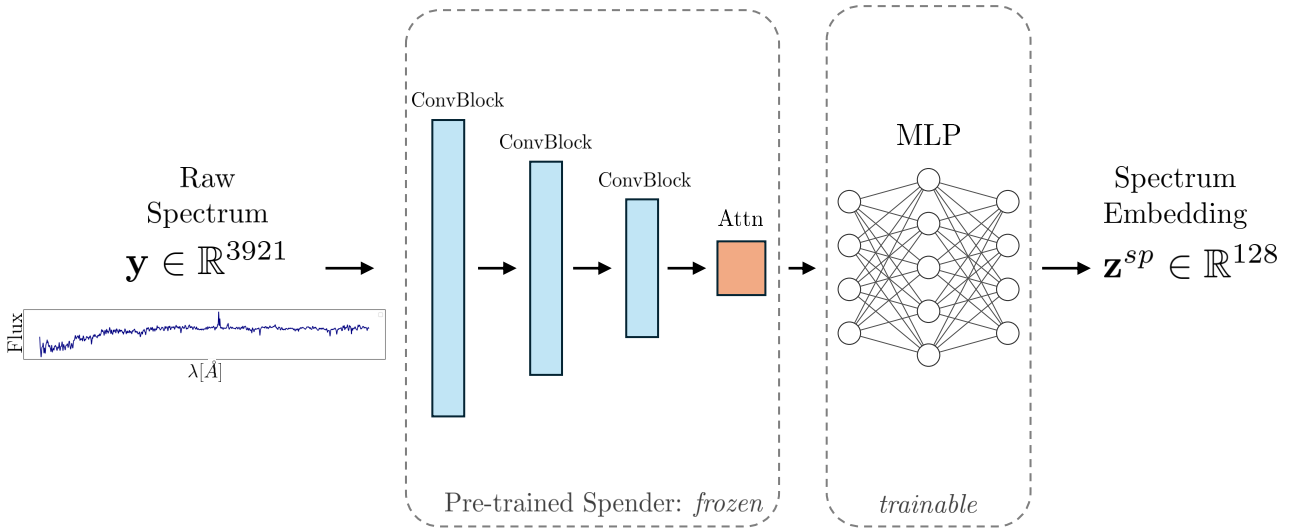


Figure 4: Overview of our spectrum embedding procedure. A raw, 1-dimensional galaxy spectrum of size 3921 is fed through the pre-trained encoder part of the Spender model, the weights of which are kept frozen during contrastive training. This consists of three convolutional blocks (ConvBlock) followed by an attention mechanism. The output of the encoder is then passed through a trainable MLP head which compresses the representation into a $d = 128$ dimensional vector representation. The output $\mathbf{z}^{\text{sp}}$ is in the shared embedding space.

For AstroCLIP, we discard the decoder part of the network and use the convolutional layers and attention mechanism of the encoder. We replace the MLP, which originally compressed the representation into a 6-dimensional latent space, with a new fully connected MLP comprising three hidden layers of sizes [256, 128, 128] and a final output layer of size $d = 128$ with ReLU activation functions. This is then trained under the InfoNCE loss function, while the convolutional layers and attention mechanism are kept frozen. These frozen layers retain semantic information about the spectra, which is beneficial through transfer learning. Overall, this results to 230,272 trainable parameters, two orders of magnitude fewer than the original work's spectrum encoder, which totalled 43.2 million parameters [7].

3.4 Contrastive Training Phase

The pre-trained models are used as components of our unified AstroCLIP model. We further train both models jointly under the InfoNCE loss defined in Eq.(1) using the training set of image-spectrum pairs as obtained in section 3.1. Embeddings originating from the same galaxy are considered positive pairs, while all others are considered negative pairs. Before an image is passed through the image embedder, we apply basic augmentations as follows: a fixed crop to the central 96×96 pixels, random rotation, random horizontal flip, and Gaussian blur. We normalise the embeddings for the similarity metric of Eq.(2) using the L2 norm.We set the batch size to $K = 512$ image-spectrum pairs, as larger batch sizes often correlate with better performance, consistent with the findings of Ref.[11]. We train the models using the Adam optimiser [23] for 80 epochs on a single NVIDIA A100-SXM-80GB GPU on the Cambridge Wilkes3 cluster. We use an adaptable learning rate scheduler that reduces the learning rate by

a factor of 2 (chosen by trial and error) if the validation loss does not improve for 5 epochs, using `PyTorch`'s `ReduceLROnPlateau` [24]. Similar to results observed in other studies, such as Ref.[25], our performance improves when we keep the temperature parameter $\tau = 15.5$ constant rather than allowing it to vary. This training process takes roughly 2 hours to complete.

3.5 Downstream Tasks

To demonstrate the capabilities of AstroCLIP's embedding space, we evaluate its performance across a range of tasks for which it was neither explicitly trained nor fine-tuned. Firstly, we embed all galaxy images and spectra in the validation set (obtained as outlined in Section 3.1) using our trained AstroCLIP model as follows:

$$\text{Initial Modality Representation} : (\mathbf{x}, \mathbf{y}) \xmapsto{\text{AstroCLIP}} \text{Embeddings} : \left(\mathbf{z}^{\text{im}}, \mathbf{z}^{\text{sp}}\right) \in \mathbb{R}^{128}. \tag{4}$$

where $\mathbf{x}, \mathbf{y}$ are the raw galaxy images and spectra, and $\mathbf{z}^{\text{im}}, \mathbf{z}^{\text{sp}}$ are the corresponding image and spectrum embeddings. We then normalise both image and spectrum embeddings as:

$$\bar{\mathbf{z}}^{\text{im}} = \frac{\mathbf{z}^{\text{im}}}{||\mathbf{z}^{\text{im}}||_2}, \quad \bar{\mathbf{z}}^{\text{sp}} = \frac{\mathbf{z}^{\text{sp}}}{||\mathbf{z}^{\text{sp}}||_2} \tag{5}$$

and these to perform the tasks outlined below.

3.5.1 Query Similarity Search

We randomly select a galaxy from the validation set and retrieve similar galaxies purely based on the embedding space structure. Specifically, caclculate the cosine similarity (Eq. (2)) between the query galaxy $\bar{\mathbf{z}}_q$ and all other galaxies in the validation set. We then rank the galaxies based on the similarity score and display the top 5 most similar galaxies.

For instance, to search for galaxy images similar to a specific query spectrum $\mathbf{y}_q$, we calculate the cosine similarity between the query spectrum embedding $\mathbf{z}_q^{\text{sp}}$ and all image embeddings $\{\mathbf{z}_j^{\text{im}}\}$ in the validation set. We denote this similarity score as $S_C(\mathbf{z}_q^{\text{sp}}, \mathbf{z}_j^{\text{im}})$. The target images with the highest similarity values are then returned. This process requires no additional transformations or alterations.

This allows us to search for both image and spectrum embeddings, which is unique to this cross-modal framework. We present examples for both *in-modality* $S_C(\mathbf{z}_q^{\text{im}}, \mathbf{z}_j^{\text{im}})$ or $S_C(\mathbf{z}_q^{\text{sp}}, \mathbf{z}_j^{\text{sp}})$, where the query and target are of the same modality, and *cross-modality* $S_C(\mathbf{z}_q^{\text{im}}, \mathbf{z}_j^{\text{sp}})$ or $S_C(\mathbf{z}_q^{\text{sp}}, \mathbf{z}_j^{\text{im}})$, where the query and target are of different modalities.

3.5.2 Zero Shot Prediction of Physical Properties

To quantify AstroCLIP's inference abilities, we consider the task of predicting the redshift Z_{HP} and stellar mass $M_\star$ of galaxies purely from their extracted embeddings. Each galaxy has a corresponding redshift and stellar mass value provided by the PROVABGS catalogue. These values are used as the ground truth for the prediction task but were not used in the training process.

We perform a simple k-Nearest Neighbour (k-NN) regression on the embeddings to predict the redshift and stellar mass values. k-NN regression works by averaging the output values of the k-nearest data points to predict the value for a new data point. To that end, we further split the 30,000 pairs in the validation set into a new training set and a held-out set. In k-NN regression, 'training' means simply storing the image embeddings and their corresponding redshift values. We then use the train set to predict a redshift value for each data point in the

held-out set, and evaluate the performance of the model by comparing the predicted redshift values to the true redshift values. The `KNeighborsRegressor` is instantiated with parameters set to `weights="distance"` and the number of neighbours is set to 16. By setting `weights` to `distance`, the algorithm ensures that closer neighbours have a greater influence on the prediction, as their contributions are inversely proportional to their distances from the query point.

3.6 Embedding Space Analysis

We extend the original work by providing an analysis of the embedding space structure. Specifically, we begin by projecting the space to a 2D space using UMAP [17]. By doing this for image embeddings in isolation, spectra embeddings in isolation and the combined image-spectrum embeddings.

First, we examine whether the model can separate the galaxy embeddings based on specific characteristics. To that end, we search for isolated collections of galaxies in the projection space, referred to as 'islands', and investigate if these galaxies share common traits. Secondly, we examine whether the model forms meaningful clusters in the original, high-dimensional embedding space before we map them to 2-dimensions. As this structure is informed by both image and spectrum representations, we expect to find clusters of galaxies that are separable in both modalities.

To detect these islands on the UMAP projection, we use the DBSCAN from `scikit-learn`. We also perform clustering on the original 128-dimensional embeddings before applying any dimensionality reduction techniques. For this purpose, we utilise the k-Means clustering algorithm [9] from `scikit-learn` . K-Means is a centroid-based clustering algorithm that partitions a dataset into k clusters, where k is a user-specified parameter. The algorithm operates by initialising k centroids, assigning each data point to the nearest centroid, and iteratively updating the centroids to minimise within-cluster variance until convergence. We evaluate the clustering quality using the silhouette score [26] to choose an appropriate k that balances the number of clusters and clustering quality. Subsequently, we project the embeddings to a 2D space using UMAP and examine the spectra and images of the clusters formed.

4 Results

In this section, we present the results of our AstroCLIP reproduction and extension. Where appropriate, we compare our results to the original work (v1) [7] and discuss possible reasons for any discrepancies. A discussion on the implications of our results and potential future work is also provided.

4.1 Loss Curves

Fig.5 shows the average InfoNCE loss per epoch for the training and validation sets during the cross-modal contrastive training. It is important to note that validation loss in the context of the contrastive framework is not a direct measure of performance but rather a proxy for the model's generalisation ability, as the training does not involve any labelled data.

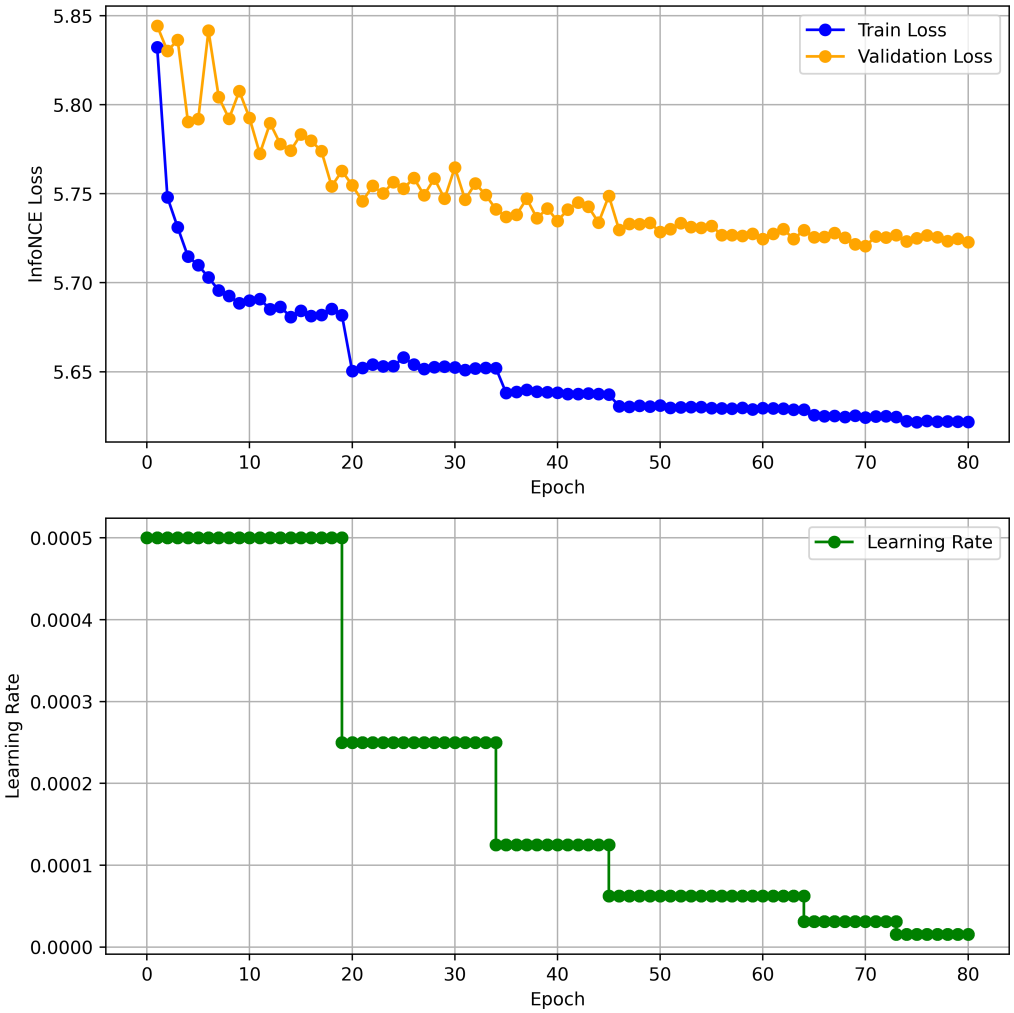


Figure 5: Average InfoNCE loss per epoch for training and validation sets during constrastive training (top), and the learning rate schedule (bottom). The learning rate is reduced by a factor of 2 if the validation loss does not improve for 5 epochs.

Both the training and validation losses decrease steadily over the 80 epochs, indicating that the model is effectively learning to align the image and spectrum embeddings in the shared latent space. The validation loss is consistently higher than the training loss, as expected, but it does not diverge significantly nor begin to increase, suggesting that the model is not overfitting. Whenever the training loss plateaus for five epochs, the learning rate is reduced by a factor of two, resulting in a further decrease in the loss. This suggests that the model is

prone to overshooting the optimal parameters and that the learning rate schedule is effective in mitigating this. The loss continues to decrease, albeit at a slower rate, until the end of the training. This could imply that the model has not fully converged and might benefit from additional training. The original work does not provide loss curves, learning rates, nor the number of epochs, and thus a direct comparison is not possible.

4.2 Query Similarity Search

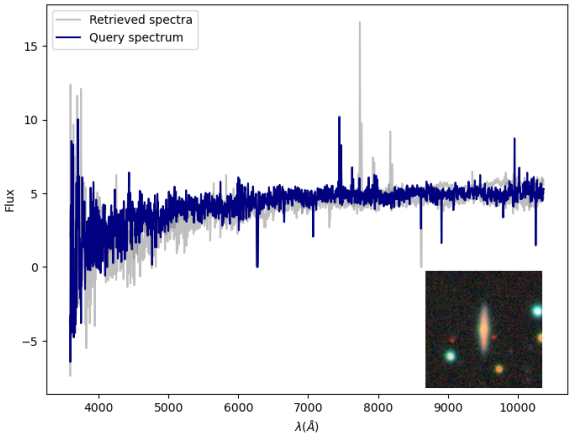
Fig.6 showcases the three most similar galaxies (based on the cosine similarity of their embeddings) to four random query galaxies for all possible combinations of modalities. By construction, for in-modal searches the best match for the query galaxy is the galaxy itself, as the similarity score is 1.0. The model is able to retrieve galaxies that are visually similar to the queries. The colour is well preserved in both in-modal and cross-modal searches, indicating that the model has learned to align the image and spectrum embeddings in the shared latent space. If the model was trained as a single-modal manner, we would expect in-modal image searches to perform better than cross-modal searches. However, the model is able to retrieve visually similar galaxies in both cases, indicating that the structuring is informed by both modalities. This is the reason why spectrum query - spectrum retrievals $S_C(\mathbf{z}_q^{\mathrm{sp}}, \mathbf{z}_j^{\mathrm{sp}})$ are also succesful.

(a) $\mathbf{z}_q$ (b) $S_C(\mathbf{z}_q^{\mathrm{im}}, \mathbf{z}_j^{\mathrm{im}})$ (c) $S_C(\mathbf{z}_q^{\mathrm{sp}}, \mathbf{z}_j^{\mathrm{sp}})$ (d) $S_C(\mathbf{z}_q^{\mathrm{sp}}, \mathbf{z}_j^{\mathrm{im}})$ (e) $S_C(\mathbf{z}_q^{\mathrm{im}}, \mathbf{z}_j^{\mathrm{sp}})$

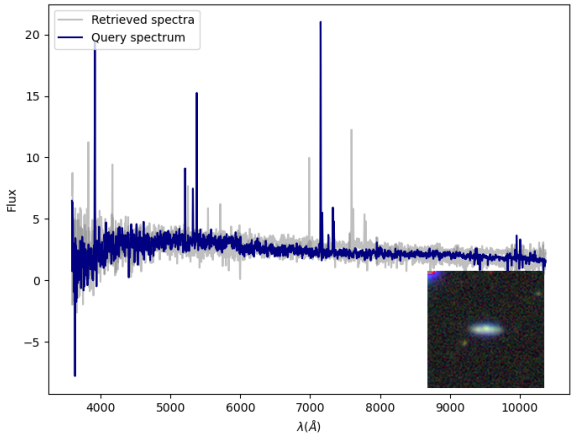
Figure 6: Query retrieval for four galaxies in the validation set. The images in block (a) are the queries. Block (b) shows the top 3 most similar galaxies to the queries based on image-image embedding similarity. Block (c) shows the top 3 most similar galaxies to the queries based on spectrum-spectrum embedding similarity. Blocks (d) and (e) show the top 3 most similar galaxies to the queries based on spectrum-image and image-spectrum embedding similarity, respectively.

We further illustrate the search capabilities by presenting the retrieved spectra for some of the queries for all combinations of modalities in Fig.7. The 5 most similar spectra are shown. The model is largely able to retrieve spectra similar to the queries, in all modality pair combinations. This is both in terms of the overall shape of the spectrum and the presence of specific spectral features, such as particular spikes.

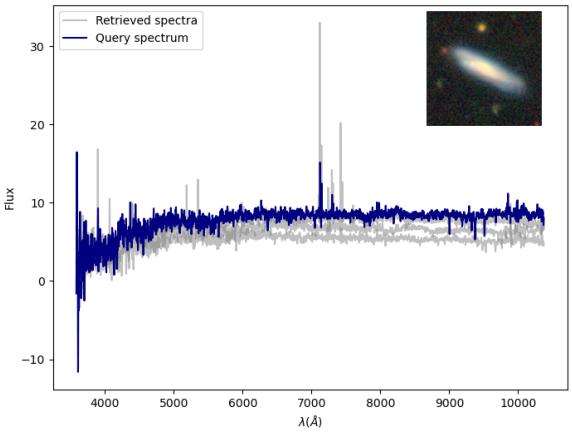
This search retrieval ability can be useful in a variety of applications, such as seach for rare or unusual objects (see some examples of this in Ref.[6]). The results are largely consistent with the original work, although a direct comparison is difficult as the images and spectra chosen for queries are random. It is worth noting that our results are achieved using a much smaller model than the original work, yet still yields (visually) similar results.



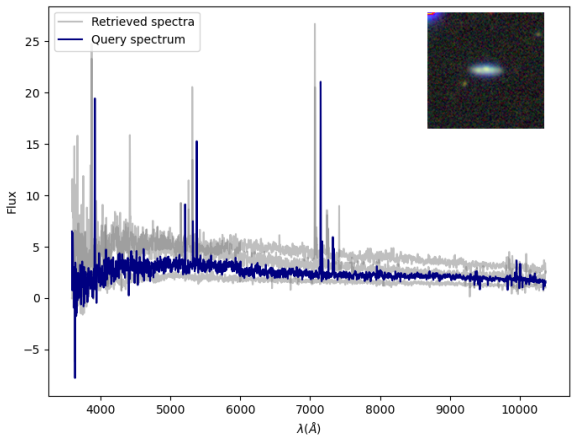
(a) $S_C(\mathbf{z}_q^{\text{im}}, \mathbf{z}_j^{\text{im}})$



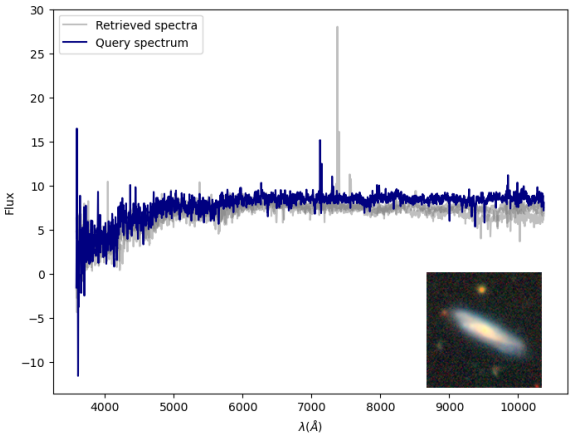
(b) $S_C(\mathbf{z}_q^{\text{im}}, \mathbf{z}_j^{\text{im}})$



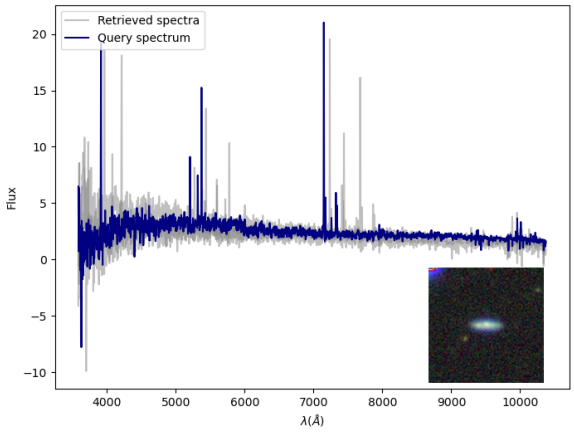
(c) $S_C(\mathbf{z}_q^{\text{sp}}, \mathbf{z}_j^{\text{im}})$



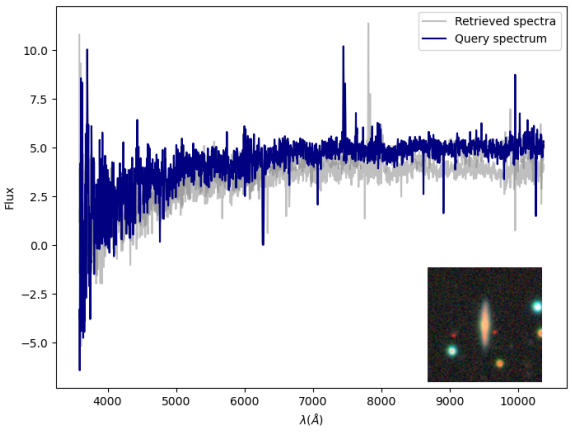
(d) $S_C(\mathbf{z}_q^{\text{sp}}, \mathbf{z}_j^{\text{im}})$



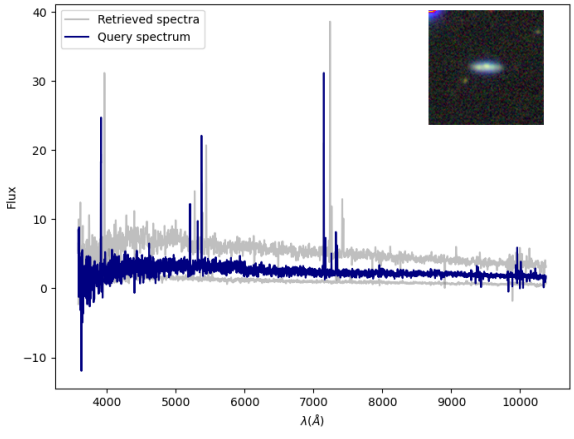
(e) $S_C(\mathbf{z}_q^{\text{im}}, \mathbf{z}_j^{\text{sp}})$



(f) $S_C(\mathbf{z}_q^{\text{im}}, \mathbf{z}_j^{\text{sp}})$



(g) $S_C(\mathbf{z}_q^{\text{sp}}, \mathbf{z}_j^{\text{sp}})$



(h) $S_C(\mathbf{z}_q^{\text{sp}}, \mathbf{z}_j^{\text{sp}})$

Figure 7: Spectral retrieval for query galaxies in the validation set. The top 5 most similar spectra are shown for each query in gray, while the query spectrum in shown in blue. The image of the query galaxy is shown in the corner of each plot.

4.3 Zero Shot Prediction of Physical Properties

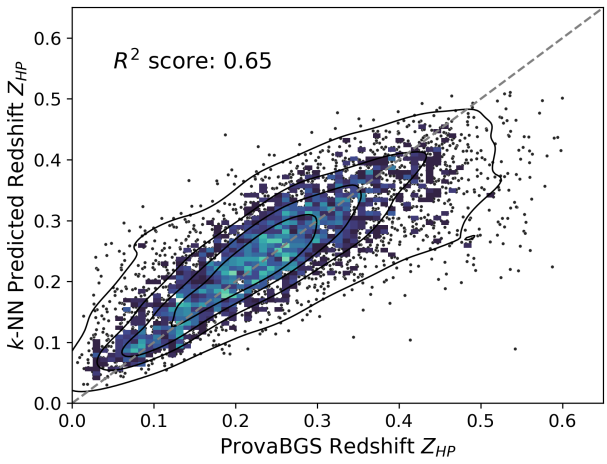
We present the results of the zero-shot prediction of redshift and stellar mass in Fig.8. These are scatterplot of the k-NN predictions against the PROVABGS catalogue values for the held-out set (gray points). A 2-D histogram is also plotted with a heatmap to visualise the joint distribution of the two variables. Additionally, a Kernel Density Estimate (KDE) is overlaid with contour lines (in black) that represent levels of density over the scatter plot. The coefficient of determination R^2 score is also calculated for each prediction, which is a measure on the proportion of variance that is explained by the model. The k-NN regression is performed for both in-modality and cross-modality regression. For in-modality, we refer to the case of using the same modality embeddings for both training and predicting the physical properties, while for cross-modality, we refer to the case of using spectrum embeddings for training and image embeddings for predicting.

We can conclude that, similar to the original work, our model has a strong zero-shot performance, indicating that the embedding space is able to organise itself around physically meaningful properties. The notably good performance even in the cross-modal case indicates that neighbours indeed share physically meaningful features regardless of the originating modality. Similar to the original work, the in-modality regression outperforms the cross-modality regression. This showcases that, although the contrastive loss aims to connect embeddings between modalities, this training objective causes the emergent property of aiding in structuring the embeddings within respective modalities. Even though redshift was not part of the training, the spectrum-spectrum regression for redshift is able to capture a very high $R^2 = 0.87$ of the variance in the data. This means that redshift has emerged as a natural property to facilitate the structuring of the embedding space.

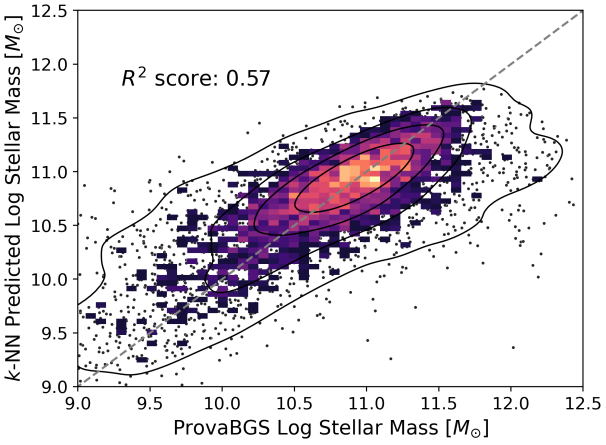
Our reproduction results are consistent with the original work, but we note that we slightly underperform in all cases. This could be due to the smaller spectrum model used in our work. It could also be due to the model architecture itself, as the original work uses a transformer-based model which could capture a more nuanced representation of the spectra. Table 1 tabulates our discrepancies.

Physical Property		Original Work R^2	Our Reproduction R^2
Redshift	Image Embeddings	0.71	0.65
	Spectrum Embeddings	0.97	0.87
	Cross-Modal	0.64	0.59
Stellar Mass	Image Embeddings	0.66	0.57
	Spectrum Embeddings	0.86	0.74
	Cross-Modal	0.58	0.50

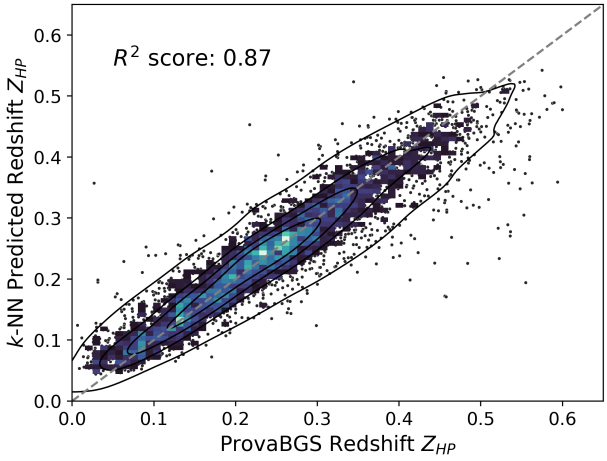
Table 1: Comparison of R^2 scores for zero-shot prediction of redshift and stellar mass between the original AstroCLIP paper (v1) and our reproduction.



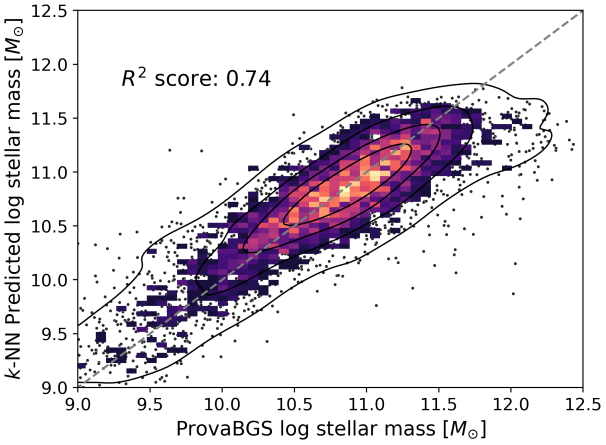
(a) Image Embeddings for Redshift



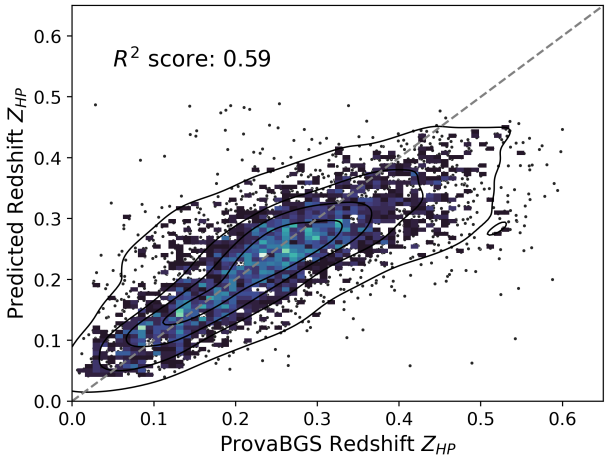
(b) Image Embeddings for Stellar Mass



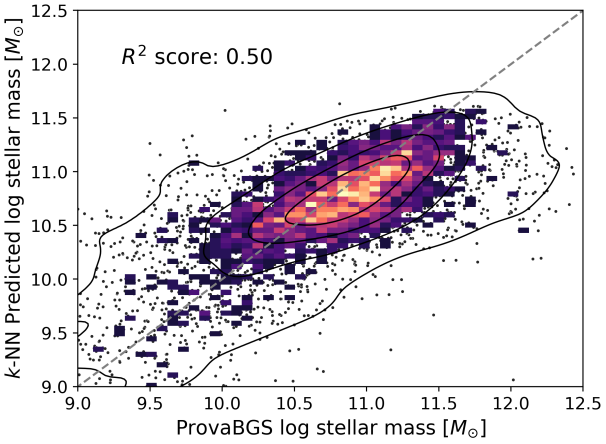
(c) Spectrum Embeddings for Redshift



(d) Spectrum Embeddings for Stellar Mass



(e) Cross-Modal Embeddings for Redshift



(f) Cross-Modal Embeddings for Stellar Mass

Figure 8: Zero-shot prediction of redshift (left column) and stellar mass (right column) for the held-out set using k-NN regression purely on the embeddings. The results are shown for in-modal prediction using image embeddings (top row), spectrum embeddings (middle row), and cross-modal prediction using spectrum embeddings for training and image embeddings for prediction (bottom row). On the x-axis we plot the PROVABGS catalogue values as ground truth, and on the y-axis we plot the k-NN predictions. A perfect prediction would lie on the diagonal line $y = x$. The R^2 score is calculated for each prediction and is annotated in the top left corner of each plot.

4.4 Embedding Space Clustering

In this section, we present the results of our extension. Fig.9 displays the UMAP projection of the spectrum embeddings, coloured by their catalogued redshift values (Z_{HP}) on the left and stellar mass values ($M_\star$) on the right. The projection reveals a clear structure: low redshift galaxies are clustered in the lower left corner, with higher redshifts appearing towards the upper right corner. The same pattern is evident in the stellar mass plot. This again demonstrates the emergent behaviour of the model in structuring the latent space around physically meaningful properties, despite not being explicitly trained on them.

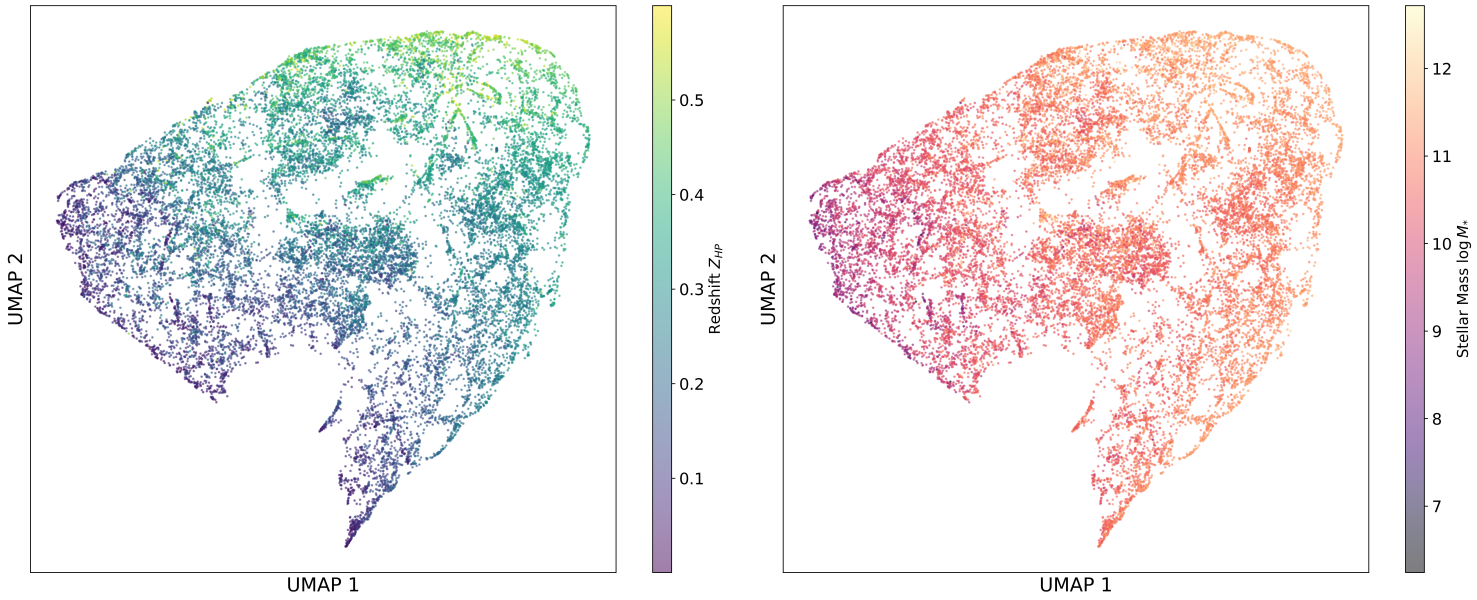


Figure 9: **Spectrum embeddings** UMAP projection coloured by redshift Z_{HP} (left) and stellar mass $M_\star$ (right). The projection reveals a clear structure, where low redshift and stellar mass galaxies are clustered in the lower left corner, rising to higher values as we move to the upper right corner.

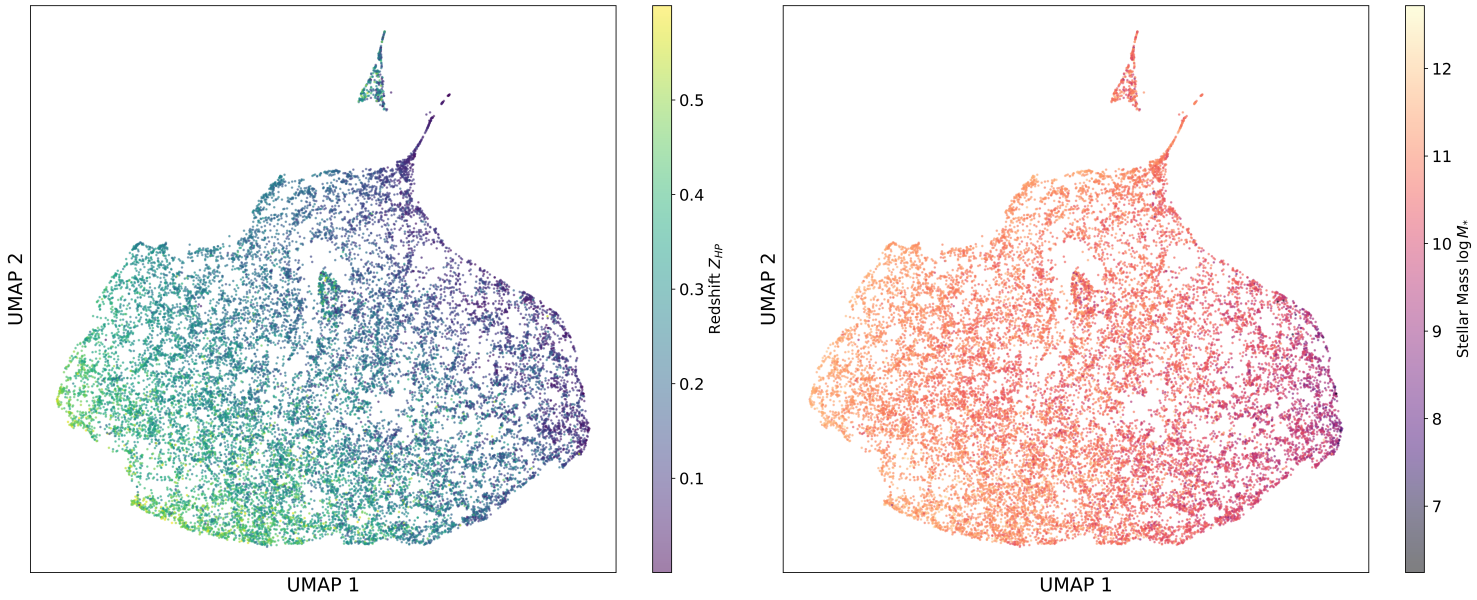


Figure 10: **Image embeddings** UMAP projection coloured by redshift Z_{HP} (left) and stellar mass $M_\star$ (right). The projection reveals a clear structure, where low redshift and stellar mass galaxies are in the left, rising to higher values as we move to the right.

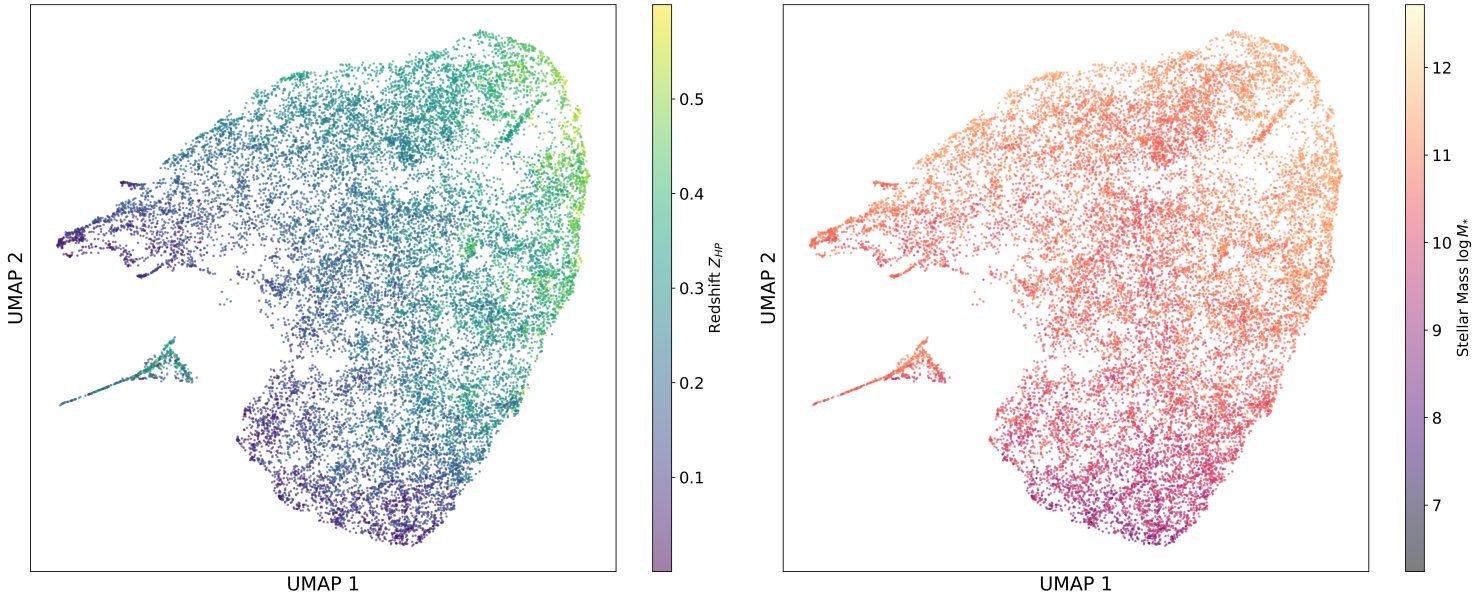


Figure 11: **Combined image-spectrum embeddings** UMAP projection coloured by redshift Z_{HP} (left) and stellar mass $M_\star$ (right). High redshift and stellar mass galaxies are in the lower left corner, falling to lower values as we move to the upper right corner.

Fig.10 and Fig.11 show the UMAP projection of the image and the combined image-spectrum embeddings, respectively. Similar to the spectrum embeddings, these projections also exhibit a clear structure. However, there is a notable difference: these mappings include a significant island of galaxies that do not correspond to particular property values. Instead, this island follows a similar pattern to the main cluster, suggesting that this separation is caused by some other feature.

To further investigate this and other potential clusters, we apply DBSCAN to directly cluster the UMAP projection of the image embeddings. We set $\epsilon = 0.20$ and the minimum number of samples to 5. These parameters were chosen through trial and error to produce a reasonable number of clusters in both the image and spectrum projections, each containing a sufficient number of galaxies. We then select sample galaxies from each cluster for visual inspection. The clusters for the image embeddings are illustrated in Fig.12.

The clustering reveals that the image embedder has effectively isolated galaxy images containing artifacts caused by the image capture process into a distinct cluster (Cluster 2). The galaxies in these images exhibit variable redshifts and stellar masses, following the same distribution as the main cluster. This capability of the model to quickly identify artifacts in galaxy catalogues is potentially very useful. Additionally, we highlight three more clusters: Cluster 3 and Cluster 4 primarily consist of galaxies that largely fill the image, are predominantly spiral, and have high brightness. This indicates that the model can distinguish galaxies based on their morphology, a key feature in galaxy classification.

It is important to note that although we visualise image embeddings in isolation, their structure is informed by the spectra during training, meaning they also contain information about the galaxy's spectrum. This reciprocal relationship holds true for the spectrum embeddings, which are informed by the images. We make this statement more concrete by showcasing the results of k-Means clustering on the full dimensional space of just the spectrum embeddings. Here, we first perform the clustering and only subsequently project the clusters onto the UMAP projection. We opt for 10 clusters, as this configuration provides a reasonable silhouette score and a sufficient number of galaxies in each cluster.

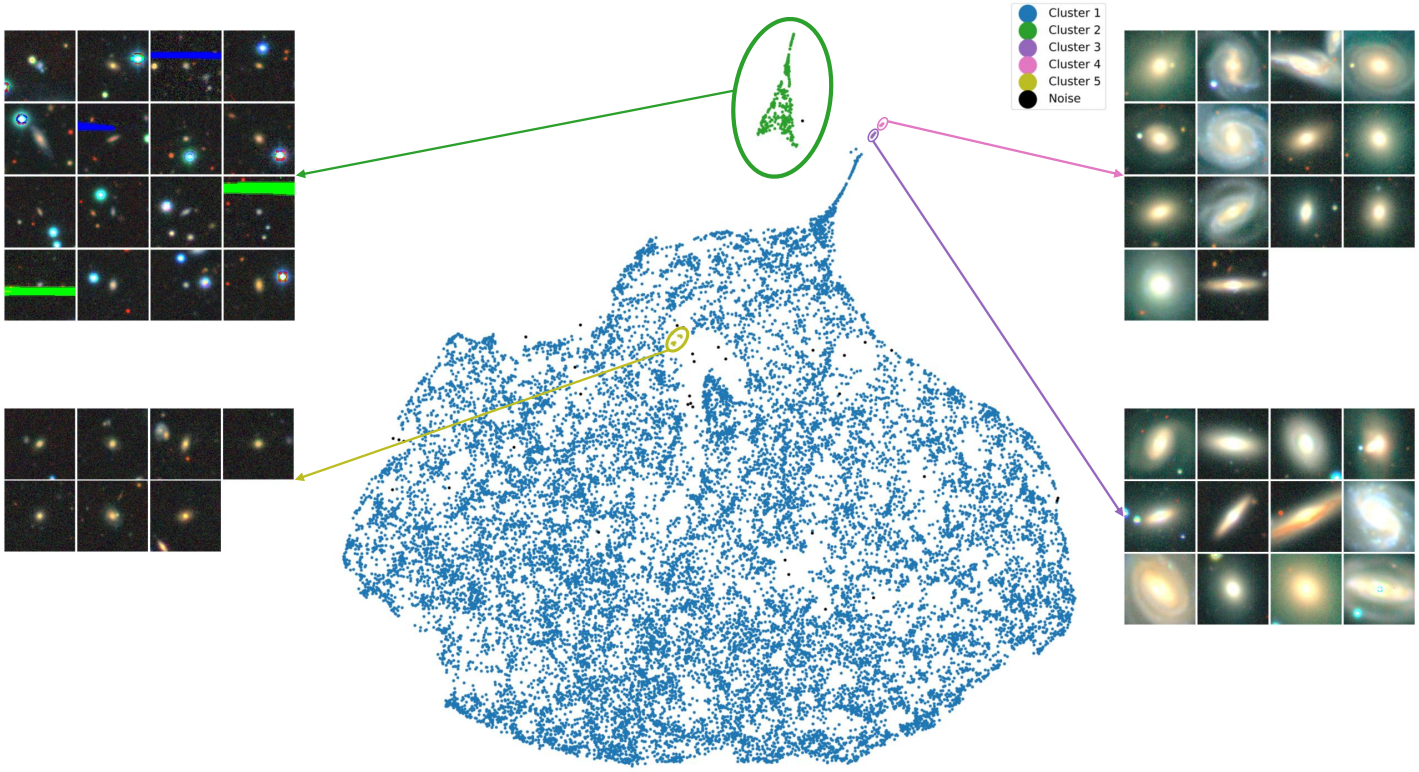


Figure 12: UMAP projection of the image embeddings coloured by the DBScan clustering. We show the images of samples of the largest island cluster (top left), and all the galaxy images contained in clusters 3, 4 and 5.

We then visualise a random sample of the spectra in each cluster, as shown in Fig.13. We observe that galaxies within the same cluster exhibit similar trends in their spectra, including overall shape, the presence of specific spectral features, and flux range and scale. Additionally, not only are the spectra within each cluster similar, but the images of the galaxies are also visually comparable, as shown in Fig.14. This is significant because we are only considering the spectrum embeddings, indicating that the model has learned to structure them in a way that is informed by the images. This capability of the AstroCLIP model allows for the alignment of the two modalities in a shared latent space, enabling clustering based on morphology, which is crucial for galaxy classification.

A notable feature of this latest UMAP projection is a distinct ring of galaxies within the main cluster, permeating different clusters. This could be an intriguing pattern to investigate further, as it may indicate a specific physical property not captured by redshift or stellar mass.

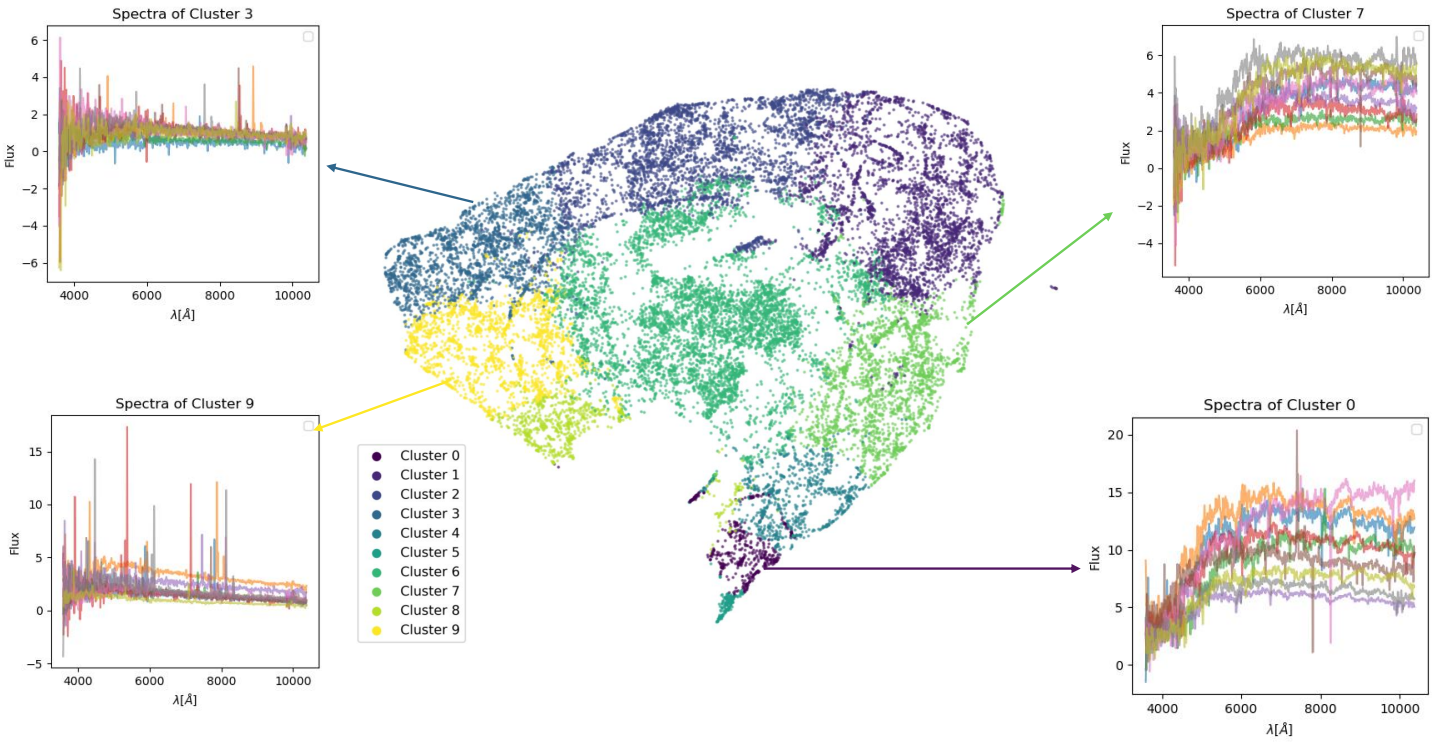


Figure 13: UMAP projection of the spectrum embeddings coloured by the k-Means clustering. The spectra of random samples from clusters [0,3,7,9] are shown in the four corners of this plot. The colours in the spectrum plots correspond to different galaxies within the specified spectrum (they are not related to the cluster colouring). All spectra follow similar trends within their respective clusters in terms of shape and range.

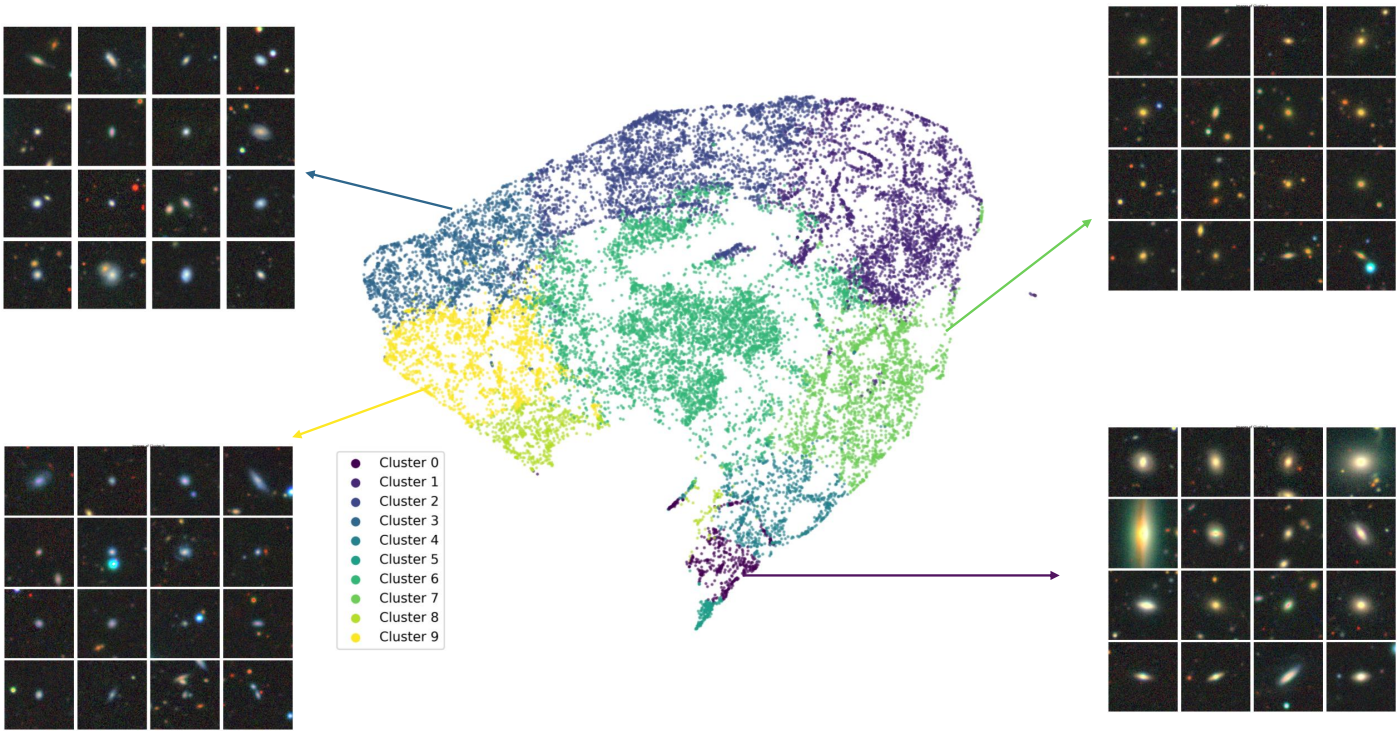


Figure 14: UMAP projection of the spectrum embeddings coloured by the k-Means clustering. The images of random samples from clusters [0, 3, 7, 9] are displayed in the four corners of this plot. The images are visually similar within their respective clusters.

5 Conclusion

In this study, we successfully reproduced and extended the results of AstroCLIP—a cross-modal foundation model for galaxies that aligns image and spectrum embeddings in a shared latent space. Initially, we deployed two separate pre-trained single-modal encoders for galaxy images and spectra, which we modified to align the embeddings in a shared latent space. We then trained the cross-modal model using an InfoNCE contrastive loss to align the embeddings of the two modalities. Utilising the extracted embeddings, we demonstrated the model's capability to retrieve similar galaxies based on a query galaxy and to predict physical properties such as redshift and stellar mass in a zero-shot regime. This ability is a key feature of the model, indicating that the latent space naturally structures itself around physical properties without being explicitely trained on them.

Despite employing a much smaller model than the original work, we achieved comparable results in all tasks. Beyond the reproduction, we extended the analysis by examining the structure of the embedding space using clustering algorithms and dimensionality reduction techniques. We demonstrated that the embeddings form visually separable clusters in both modalities, exemplifying the model's ability to structure them based on shared semantics.

Our results confirm the potential of cross-modal contrastive pre-training to achieve high-quality foundation models for astronomical data. This work opens the avenue for integrating additional modalities into the shared space, such as photometry, multi-band images, or even text data. Ultimately, this constitutes one of the first applications of foundation models in astromony, marking a promising advancement in the broader field of machine learning in science.

References

[1] Arjun Dey, David J. Schlegel, Dustin Lang, Robert Blum, Kaylan Burleigh, Xiaohui Fan, Joseph R. Findlay, Doug Finkbeiner, David Herrera, Stéphanie Juneau, Martin Landriau, Michael Levi, Ian McGreer, Aaron Meisner, Adam D. Myers, and Moustakas et al. Overview of the desi legacy imaging surveys. *The Astronomical Journal*, 157(5):168, April 2019.

[2] Željko Ivezić, Andrew J Connolly, Jacob T VanderPlas, and Alexander Gray. *Statistics, data mining, and machine learning in astronomy: a practical Python guide for the analysis of survey data*, volume 8. Princeton University Press, 2020.

[3] Kyle W Willett, Chris J Lintott, Steven P Bamford, Karen L Masters, Brooke D Simmons, Kevin RV Casteels, Edward M Edmondson, Lucy F Fortson, Sugata Kaviraj, William C Keel, et al. Galaxy zoo 2: detailed morphological classifications for 304 122 galaxies from the sloan digital sky survey. *Monthly Notices of the Royal Astronomical Society*, 435(4):2835–2860, 2013.

[4] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners, 2021.

[5] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[6] George Stein, Jacqueline Blaum, Peter Z. Harrington, Tomislav Medan, and Zarija Lukic. Mining for strong gravitational lenses with self-supervised learning. *The Astrophysical Journal*, 932, 2021.

[7] Liam Parker, Francois Lanusse, Siavash Golkar, Leopoldo Sarra, Miles Cranmer, Alberto Bietti, Michael Eickenberg, Geraud Krawezik, Michael McCabe, Ruben Ohana, Mariel Pettee, Bruno Regaldo-Saint Blancard, Tiberiu Tesileanu, Kyunghyun Cho, and Shirley Ho. Astroclip: A cross-modal foundation model for galaxies, 2024.

[8] Sayash Kapoor and Arvind Narayanan. Leakage and the reproducibility crisis in machine-learning-based science. *Patterns*, 4(9), 2023.

[9] Simon J.D. Prince. *Understanding Deep Learning*. The MIT Press, 2023.

[10] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[11] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[12] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations, 2020.

[13] Ching-Yao Chuang, Joshua Robinson, Yen-Chen Lin, Antonio Torralba, and Stefanie Jegelka. Debiased contrastive learning. *Advances in neural information processing systems*, 33:8765–8775, 2020.

[14] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning Transferable Visual Models From Natural Language Supervision, February 2021. arXiv:2103.00020 [cs].

[15] David McAllester and Karl Stratos. Formal limitations on the measurement of mutual information. In Silvia Chiappa and Roberto Calandra, editors, *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pages 875–884. PMLR, 26–28 Aug 2020.

[16] Aäron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *CoRR*, abs/1807.03748, 2018.

[17] Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction, 2020.

[18] Erich Schubert, Jörg Sander, Martin Ester, Hans Peter Kriegel, and Xiaowei Xu. Dbscan revisited, revisited: Why and how you should (still) use dbscan. *ACM Trans. Database Syst.*, 42(3), jul 2017.

[19] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning, 2020.

[20] Peter Melchior, Yan Liang, ChangHoon Hahn, and Andy Goulding. Autoencoding galaxy spectra. i. architecture. *The Astronomical Journal*, 166(2):74, July 2023.

[21] DESI Collaboration Et Al. The early data release of the dark energy spectroscopic instrument, 2023.

[22] ChangHoon Hahn, Jessica Nicole Aguilar, Shadab Alam, Steven Ahlen, David Brooks, Shaun Cole, Axel de la Macorra, Peter Doel, Andreu A. Font-Ribera, Jaime E. Forero-Romero, Satya Gontcho A Gontcho, Klaus Honscheid, Song Huang, Theodore Kisner, Anthony Kremin, Martin Landriau, Marc Manera, Aaron Meisner, Ramon Miquel, John Moustakas, Jundan Nie, Claire Poppett, Graziano Rossi, Amélie Saintonge, Eusebio Sanchez, Christoph Saulder, Michael Schubnell, Hee-Jong Seo, Małgorzata Siudek, Federico Speranza, Gregory Tarlé, Benjamin A. Weaver, Risa H. Wechsler, Sihan Yuan, Zhimin Zhou, and Hu Zou. Provabgs: The probabilistic stellar mass function of the bgs one-percent survey, 2023.

[23] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017.

[24] ReduceLROnPlateau — PyTorch 2.3 documentation.

[25] Rohit Girdhar, Alaaeldin El-Nouby, Zhuang Liu, Mannat Singh, Kalyan Vasudev Alwala, Armand Joulin, and Ishan Misra. Imagebind: One embedding space to bind them all, 2023.

[26] Ketan Rajshekhar Shahapure and Charles Nicholas. Cluster quality analysis using silhouette score. In *2020 IEEE 7th International Conference on Data Science and Advanced Analytics (DSAA)*, pages 747–748, 2020.

Appendix

Use of Co-pilot

Co-pilot was utilised in the following ways:

- Autocompleting documentation for functions and for scripts.
- Understanding error messages with the 'Explain this error' button next to the error message.
- Responding to general queries, especially LaTeX.

For example:
Prompt1: How do i put two images together in a single figure, horizontally next to each other, with the same caption, in latex?
Answer1: You can use the 'subfigure' environment in LaTeX to put two images together in a single figure. Here is an example code snippet that puts two images next to each other with the same caption...
Prompt2: How can I change a function name and have it change throughout my workspace?
Answer2: In Visual Studio Code, you can use the "Rename Symbol" feature to change a function name throughout your workspace. Here are the steps:

1. Place your cursor on the function name you want to change.
2. Right-click the function name, and select "Rename Symbol" from the context menu. Alternatively, you can use the shortcut F2.
3. Enter the new function name and press Enter.

VS Code will automatically find and update all occurrences of the function name in your workspace. This includes not only the function definition but also all places where the function is called.