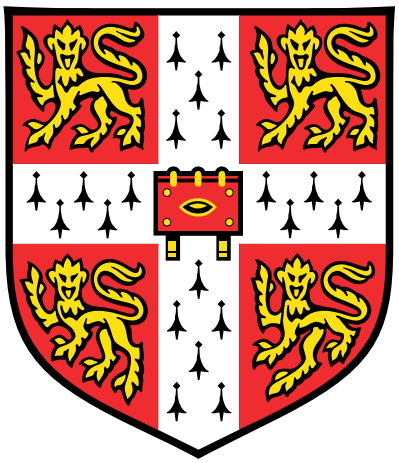
MPhil Data Intensive Science
University of Cambridge

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

Data Analysis Project



Andreas Vrikkis

December 14, 2023

LaTeX Word count: 2825

Contents

1 Introduction

2 Method/Background/Theory

2.1 Foundational Models

2.2 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 (potentially labelled) data [1]. The resulting network 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 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. In many cases, however, labelled data are scarce or expensive to obtain. In self-supervised learning (SSL), a model is trained on 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 [2], masked language modelling [3] and contrastive learning [4]. These techniques have shown success in generating versatile and informative representations across NLP and computer vision.

2.3 Cross-modal Constrastive Learning

Constastive learning is a self-supervised learning technique. The key concept behind our training objective is that different observational modalities represent correlated transformations of the same underlying physical object, forming a positive pair. By modality we refer to the type of data input such as iamges, text, etc. each requring a different processing technique. This is an extension of the single-modal contrastive learning framework such as SimCLR [cite], which employs stochastic data augmentation to create two views from the underlying image. Instead, we begin with two different modalities in different spaces and map them to a common embedding space, similar to the cross-modal framework connecting language and images in Ref.[5]. Under the constrastive loss function, each positive pair is projected together in the embedding space, while negative pairs are pushed apart.

The cross-modal contrastive learning framework is illustrated in Fig.1. Here, $\mathbf{x}_i \in \mathbb{R}^N$ and $\mathbf{y}_i \in \mathbb{R}^M$ are two modalities of the same underlying object. These are passed through a pair of encoder networks $f_\theta : \mathbb{R}^{\mathbb{N}} \to \mathbb{R}^{\mathbb{D}}$ and $g_\phi : \mathbb{R}^{\mathbb{M}} \to \mathbb{R}^{\mathbb{D}}$, with θ, ϕ trainable parameters, which extract representation vectors in the shared embedding space where contrastive loss is applied. These are denoted as $\mathbf{z}_i^x \in \mathbb{R}^D$ and $\mathbf{z}_i^y \in \mathbb{R}^D$, where D is the dimensionality of the embedding space and the superscripts x and y denote the originator modality.

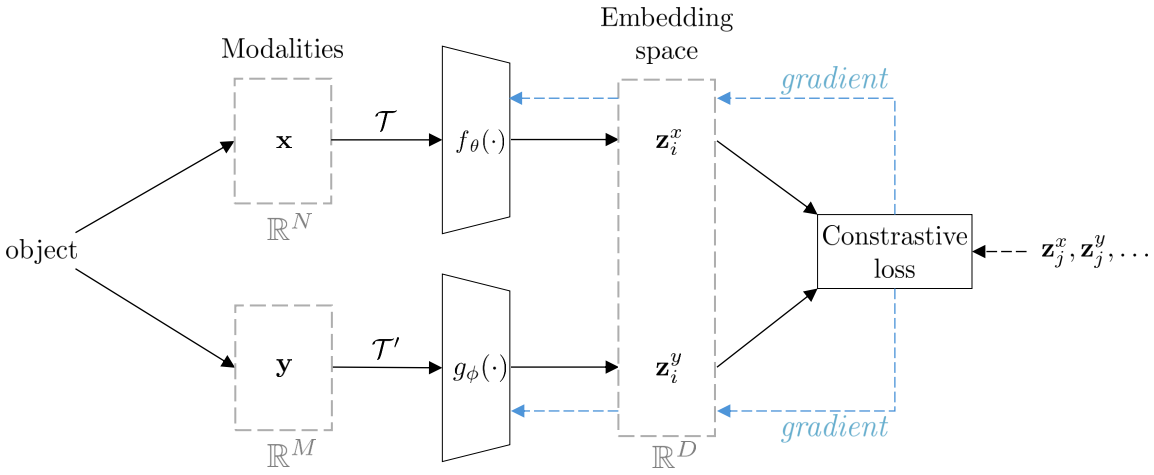


Figure 1: Cross-modal contrastive learning framework. An underlying physical object is observed in two different modalities, $\mathbf{x}_i \in \mathbb{R}^{\mathbb{N}}$ and $\mathbf{y}_i \in \mathbb{R}^{\mathbb{M}}$. An augmented version of each modality undergoes a transformation $\mathcal{T}, \mathcal{T}'$ 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 constrastive loss (InfoNCE) is applied to these representations along with the negative pair denoted by j, with the gradients backpropagated to update the encoder networks (appended with projection heads).

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 [6]. Instead, we approximate each modality as a noisy transformation of the same underlying object and use an Information Noise-Contrastive Estimation (InfoNCE) loss function [7] 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 between the two representations in the embedding space 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 constrastive methods [5]. Under InfoNCE, points in the embedding space that correspond to the same object are pulled together, while points that correspod to different objects are pushed apart.

Training cross-modal models from scratch under CLIP loss on cross-modal problems has been shown to often underperform compared to single-modal problems. However, the addition of pre-trained single modal models as initialisation can significantly improve performance. This

is the approach taken in this project, where only an additional projection head is trained on top of the pre-trained models.

2.4 Embedding Space

The low-dimensional, rich representations of the data in the shared embedding space can then be used for a variety of downstream tasks. It has been shown in various contexts that despite the contrived training objective, the embedding space can capture a significant amount of semantic information that often outperform supervised training on zero-shot and few-shot learning tasks. Recall that during constrastive training, the embedding space structures itself such that semantically similar objects are close together, while semantically dissimilar objects are far apart. This means that the positions of the representations in the embedding space can be used to infer semantic relationships between the objects and can be used for downstream tasks. A typical way of visualising the embedding space is to project the representations to a 2D space using dimensionality reduction techniques.

UMAP (Uniform Manifold Approximation and Projection) is a dimensionality reduction technique that aims to preserve the global structure and local relationships of data when mapping high-dimensional data to a lower-dimensional space [8]. UMAP constructs 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.

As the embedding space structure is informed by the contrastive loss, the arrangment of the representations holds semantic information. Island structures in the embedding space can be interpreted as clusters of semantically similar objects, while the distance between islands can be used to infer semantic relationships between the clusters.

3 Implementation

In AstroCLIP, we consider two modalities of galaxies: images and spectra. In essence, we approach the two modalities as filtered, noisy views of the same underlying galaxy. This implies that they should possess a shared latent space in which the embeddings of these cross-modal representations are aligned. To obtain these embeddings, we deploy a pair of models to encode the images and spectra. This is done in a two-step process:

1. We utilise two pre-trained single-modal models, one for images and one for spectra, which were pre-trained using SSL. For the pretrained image embedder, we use a galaxy image encoder from Stein et. al (see Ref.[9]), which is based on MoCo v2 [10]. For galaxy spectra, we utilise the encoder part from the Spender model [11].

2. We append a simple Multilayer Perceptron (MLP) projection head to each of the pre-trained models to compress the representations into a shared $d = 128$-dimensional embedding space. We then train (or fine-tune) these projection heads under constrastive learning to align the embedded representations of the two modalities under shared semantics.

We keep the pre-train single-modal models frozen during training, only updating the projection heads, instead of training the entire model from scratch. This is to align with the previous studies that show that pre-training the single-modal models significantly improves performance in cross-modal tasks [5].

We note here the deviation from the original AstroCLIP paper (v1): in the original papers, the authors pre-train a transformer model, structured similar to GPT-2, to embed the spectra. This is a much larger model than Spender, totalling around 43.2M parameters. However, the

transformer model is not publicly available, and the authors do not provide details on the pre-training process. Instead, we opt to use the Spender model based on the suggestion 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

DESI Legacy Survey Images

We use the same data as the original AstroCLIP paper (v1). For galaxy images, we sue the DESI Legacy Survey Data Release 9 imaging data from January 2021 [12] as prepared by Stein et. al [9]. 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 came from the Mayall Legacy Survey. For the southern galactic cap (SGC), the data were collected by the Dark Energy Camera Legacy Survey (DECaLS). We filter out the images that were identified as stars by the Legacy Survey team and impose a mag_z cutoff for z-bands above 20. This corresponds to an initial dataset of 41M (g,r,z) images of size 152×152, which we centre-crop to 96×96 for training. The cropping is done as the great majority of galaxies cover less area than the total size of the image and thus often include overlaping regions of the sky.

DESI Spectra

To pair the images with spectra, we cross-match the galaxy spectra from the DESI Early Data Release [13], which contains spectra observed by the Survery Validation compaign. This cross-match is done using the target IDs associated with each galaxy. This results in 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.

Data Catalogue for Downstream Tasks

For experiments involving the prediction of physical properties from the embeddings, we further cross-match the image-spectrum pairs with the PRObabilistic Value-Added Bright Galaxy Survery (PROVABGS) catalogue [14]. We then specifically extract the estimates of the redshift ($\mathcal{Z}$) and stellar mass ($M_\star$) for each galaxy ID that is present in the catalogue. We then perform the same filtering process as the original AstroCLIP paper: we only select enetries for which $M_\star > 0$ and mag_g,mag_r,$\text{mag}_z > 0$, removing spurious entries. This yields a total of 105,159 entries, which we split into training and validation using a 90/10 ratio.

3.2 Pre-trained Image Embedder

The pre-trained image embedder we used in this reconstruction was developed by Stein et. al. (2021a) [9]. It is based on the MoCo v2 framework [10], and it uses a ResNet-50 backbone as the encoder network. The model was pre-trained on a curated subset of 3.5M galaxies sampled uniformly by z-bands magnitude from the DESI Legacy Survey. They authors use a single-modal contrastive learning SSL framework, where each image undergoes multiple stochastic augmentations to create two views of the same image. The augmentations include galactic extinction, random cropping, random rotation, size scaling, point-spread function blur, jittering and Gaussian noise addition. The model was then trained on a contrastive loss function to align the representations of the two views in the embedding space.

The model is publicly available and can be downloaded from the authors' GitHub repository. This model has 50 ResNet blocks that contain a batch normalisation opeartoin, followed by

an activation function and a convolutional layer. It has a total of 28M parameters. We keep the convolutional layers frozen during constrastive training, and use the final fully connected layer as the trainable projection head. This amounts to 4.5M trainable parameters that are finetuned under InfoNCE loss.

3.3 Pre-trained Spectrum Embedder

We choose to use the Spender model [11], an autoencoder network, to extract latent parameters from the observed spectra. The encoder part of the network consists of three convolutional layers with progressively wider kernel sizes (5, 11, 21), each followed by a trainable PReLU activation function and max-pooling. This translates the $M = 3921$ spectral components into 512 channels. The model then applies attention by splitting the channels into two: $\mathbf{h}$ and $\mathbf{k}$, each of dimension 256×72. It then combines these channels as follows:

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

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. These are then fed into a series of fully connected layers to compress the representation into a $s = 6$ dimensional latent space. These are then fed into 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.

For AstroCLIP, we throw away the decoder part of the network, and use the convolutional layers and attention mechanism of the encoder. We replace the MLP which was used to compress the representation into a 6-dimensional latent space with a new fully connected MLP with 3 hidden layers of size 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. The frozen layers contain semantic information about the spectra that is useful to incorporate through transfer learning. Overall, this amounts to [fix] trainable parameters, two orders of magnitude less than the original AstroCLIP encoder (4.5M parameters) [15].

3.4 Constrastive Training

The pre-trained models form parts of our unified AstroCLIP model. We then train both models under InfoNCE loss defined in eq.(1). The embedded representations that originate from the same galaxy are considered positive pairs, with all other considered negative pairs. Before an image is passed through the image embedder, we apply basic augementations in the following manner: a fixed crop to the central 96×96 pixels, random roation, random horizontal flip and gaussian blur. We set the batch size to $K = 512$ image-spectrum pairs, as it was shown in Ref.[4] that larger batch sizes often correlate with better performance. We train the models using the Adam optimiser [16] for 80 epochs on a single NVIDIA A100-SXM-80GB GPU on the Camridge 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` [17]. Similar to the results observed in other studies such as Ref.[18], our performance improves when we keep the temperature parameter τ constant rather than allowing it to vary. The training process takes roughly 2 hours to complete.

3.5 Downstream Tasks

To demonstrate the capabilities of AstroCLIP's embedding space, we demonstrate its performance across a range of tasks for which it was not explicitly trained for, nor fine-tuned. We firstly embed all galaxy image and spectra in the validation set (obtained as outlined in section 3.1) using the trained AstroCLIP model as follows:

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

We then normalise both image and spectrum embeddings as:

$$\bar{\mathbf{z}}^{im} = \frac{\mathbf{z}^{im}}{||\mathbf{z}^{im}||_2}, \quad \bar{\mathbf{z}}^{sp} = \frac{\mathbf{z}^{sp}}{||\mathbf{z}^{sp}||_2}. \quad (5)$$

and use the normalised galaxy embeddings in a shared, cross-modal latent space to perform the tasks outlined below.

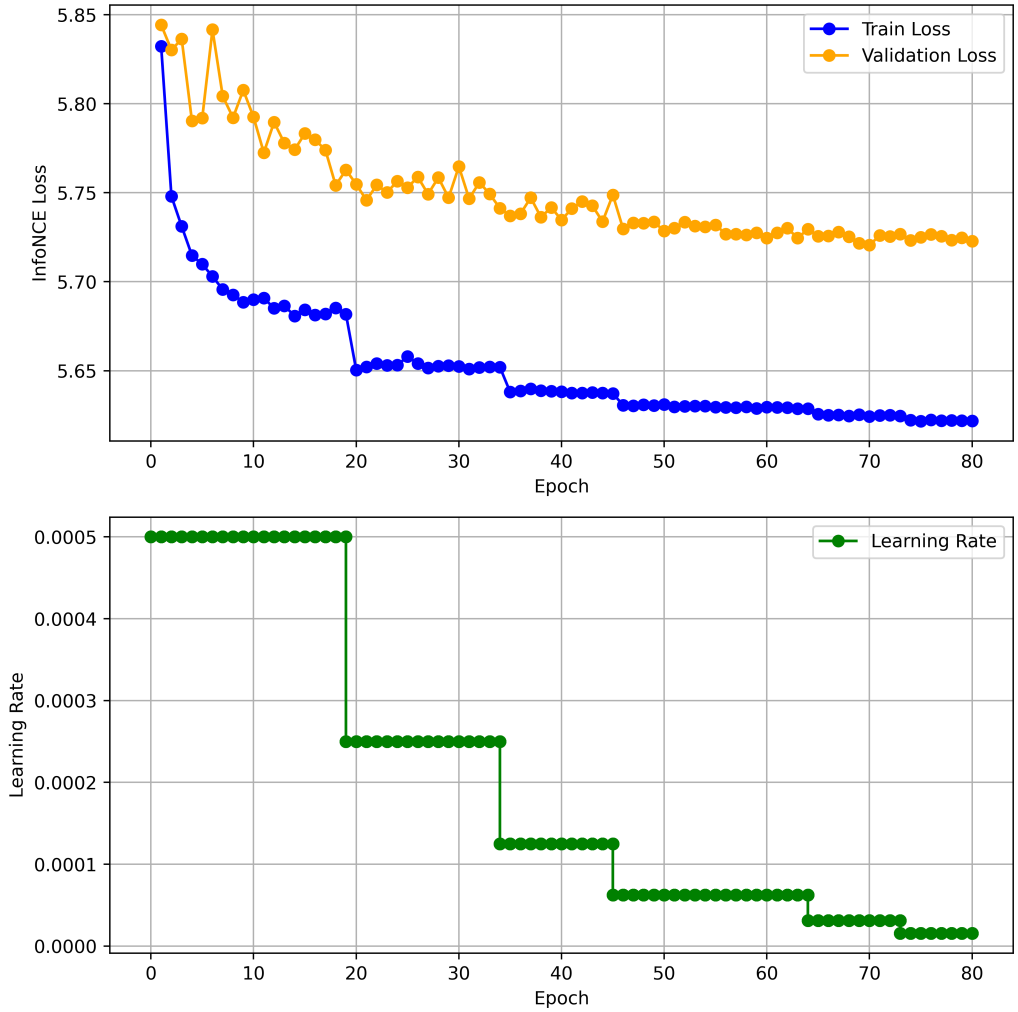
Query Retrieval

We showcase the architecture's ability to align embeddings of galaxies, we choose a random galaxy from the validation set and retrieve similar galaxies purely based on the embedding space structure. Specifically, we perform the galaxy search using 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 that correspond to a specific query spectrum $\mathbf{y}_i$, we calculate the cosine similarity between the query spectrum embedding $\{\mathbf{x}_i^{sp}\}_{val}$ and the image embeddings $\{\mathbf{x}_i^{im}\}_{val}$ in the validation set. The target images with the highest similarity values are then returned. This process requires no additional transformations or alterations.

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

Zero Shot Prediction of Physical Properties

To evaluate AstroCLIP's ability to predict physical properties of galaxies, we consider the task of predicting the catalogue-based redshift $\mathcal{Z}$ and stellar mass $M_\star$ of galaxies from the embedded galaxy representations. We use simple k-Nearest Neighbour (k-NN) our embedded images and spectra to predict the physical properties without any additional training or finetuning. We do this by further splitting the 30K pairs in the validation set into a training and held-out test set. We then train a k-NN regressor on the new training set and evaluate its performance on the test set. We then instantiate the $KNeighborsRegressor$ with parameters set to `weights="distance"` and `n_neighbors=16`. By setting `weights` to `"distance"`, the algorithm ensures that closer neighbors have a greater influence on the prediction, as their contributions are inversely proportional to their distances from the query point. This method allows us to calculate the predicted values for redshift and stellar mass as weighted averages of the target values of these nearest neighbors.



H

Figure 2

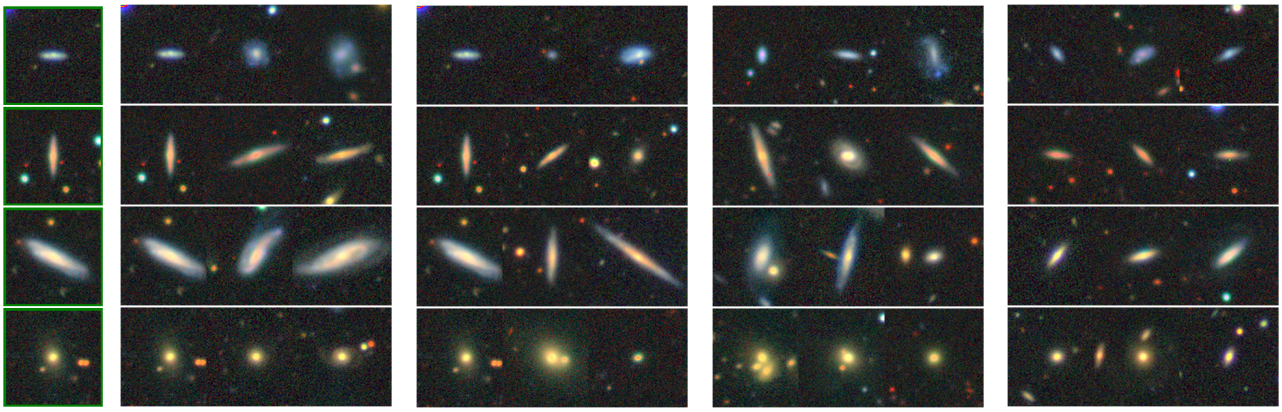
Visualisation and Clustering of the Embedding Space

4 Results

In this section, we present our results from the AstroCLIP reproduction. Where appropriate, we compare our results to the original AstroCLIP paper (v1) [15] and discuss possible reasons for any discrepancies. A discussion on the implications of our results and potential future work is also provided.

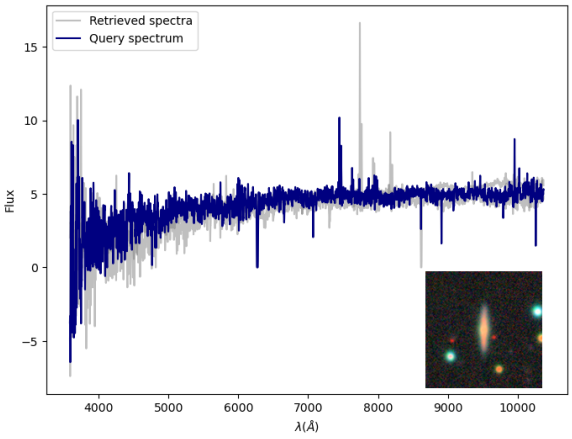
4.1 Loss Curves

4.2 Query Retrieval

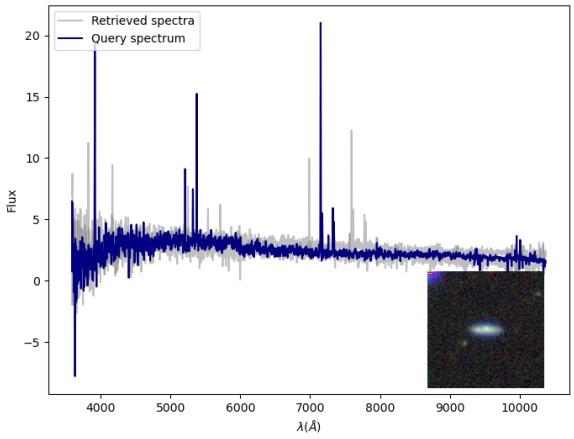


(a) $\mathbf{z}_q$ (b) $S_C(\mathbf{z}_q^{im}, \mathbf{z}^{im})$ (c) $S_C(\mathbf{z}_q^{sp}, \mathbf{z}^{sp})$ (d) $S_C(\mathbf{z}_q^{sp}, \mathbf{z}^{im})$ (e) $S_C(\mathbf{z}_q^{im}, \mathbf{z}^{sp})$

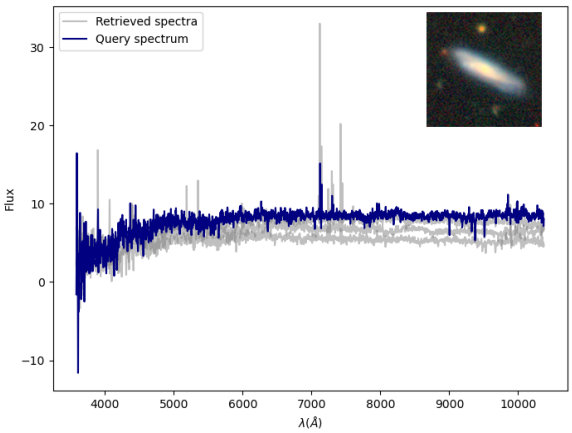
Figure 3: Your overall figure caption here.



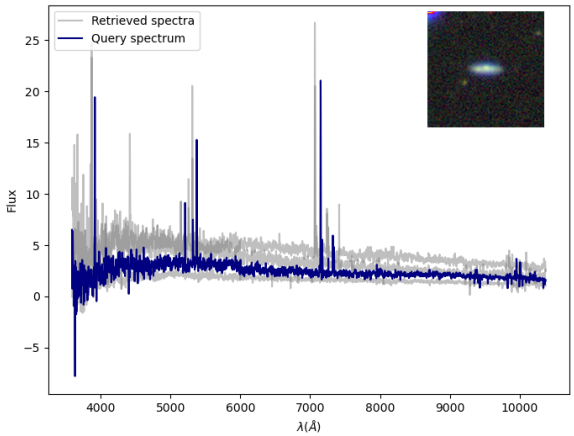
(a) Caption for im_im_1



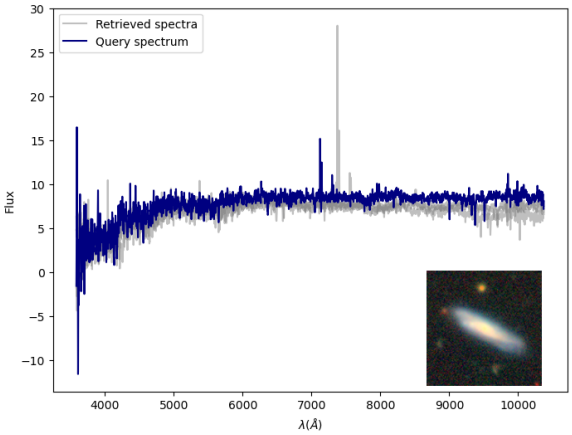
(b) Caption for im_im_2



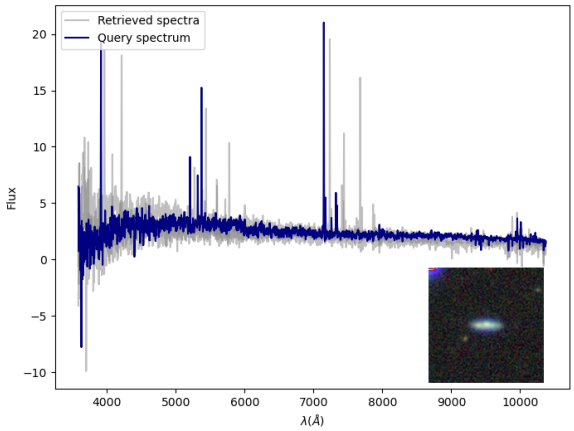
(c) Caption for im_sp_1



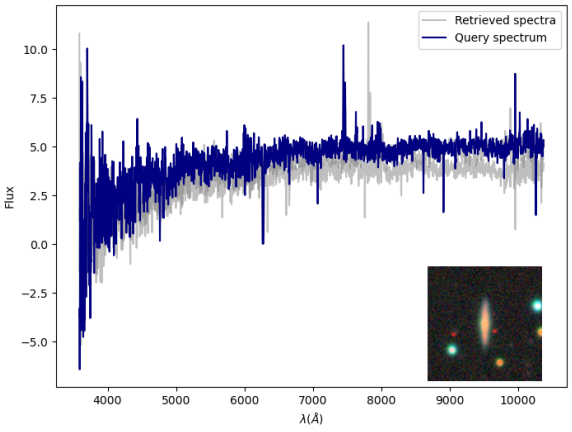
(d) Caption for im_sp_2



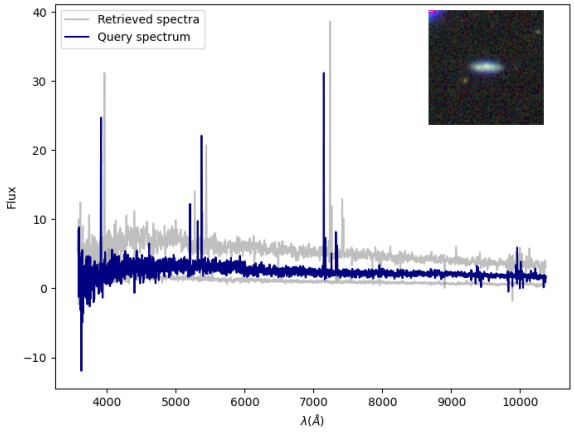
(e) Caption for sp_im_1



(f) Caption for sp_im_2



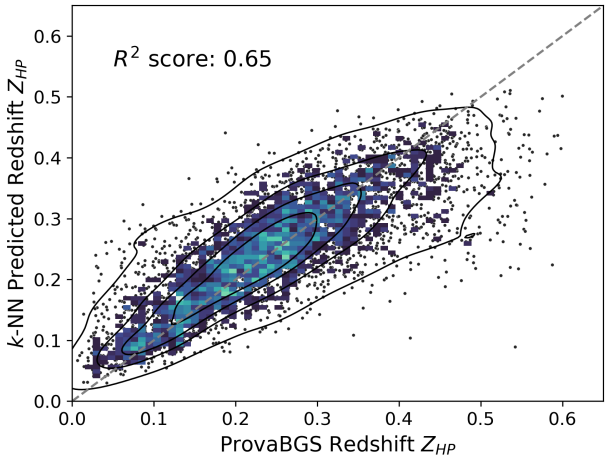
(g) Caption for sp_sp_1



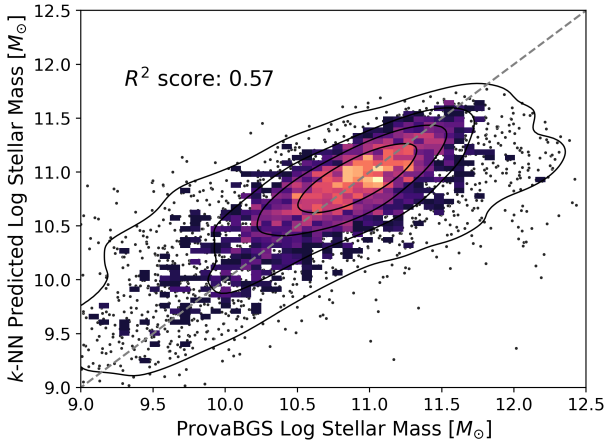
(h) Caption for sp_sp_2

Figure 4: Overall caption for all images.

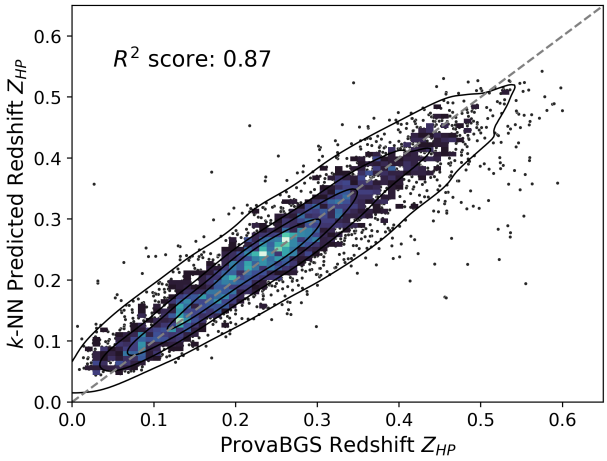
4.3 Zero Shot Prediction of Physical Properties



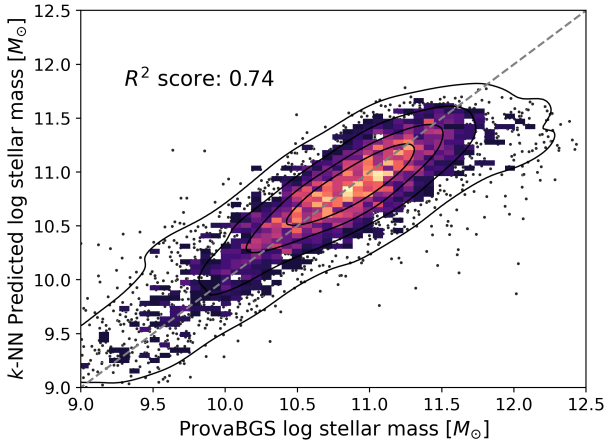
(a) Image Embeddings for Redshift



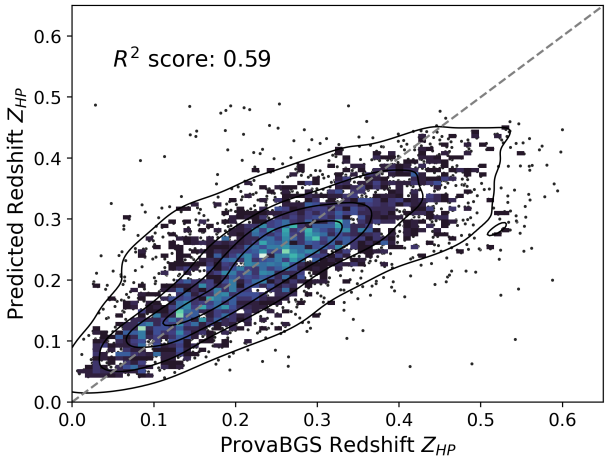
(b) Image Embeddings for Stellar Mass



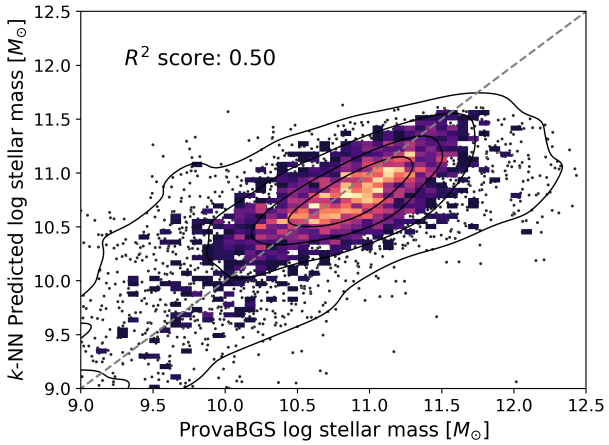
(c) Spectrum Embeddings for Redshift



(d) Spectrum Embeddings for Stellar Mass



(e) Cross-Modal Embeddings for Redshift



(f) Cross-Modal Embeddings for Stellar Mass

Figure 5: Comparative visualizations of zero-shot redshift and stellar mass estimation across different embedding types.

References

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

[2] 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.

[3] 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.

[4] 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.

[5] 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].

[6] 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.

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

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

[9] 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.

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

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

[12] 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.

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

[14] 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.

[15] 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.

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

[17] ReduceLROnPlateau — PyTorch 2.3 documentation.

[18] 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.