Towards Better Active Deep Learning with Automatic Calibration Diagnosis

Ondřej Podsztavek

Czech Technical University in Prague
Faculty of Information Technology
Department of Computer Systems

A dissertation submitted to the Faculty of Information Technology of Czech Technical University in Prague in partial fulfillment of the requirements for the degree of *Doctor*.

Doctoral study program: Informatics
Supervisor: prof. Ing. Pavel Tvrdík, CSc.
Co-supervisor: RNDr. Petr Škoda, CSc.

Prague 2025

Abstract

Abstrakt

Acknowledgements

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

There is a lot of data everywhere, on the internet, in science etc. Take astronomy as an exemplary scientific domain with a lot of observational data. For example, the archives of the Large Sky Area Multi-Object Fiber Spectroscopic Telescope (LAMOST) (Gang et al. 2012) and Sloan Digital Sky Survey (SDSS) (Blanton et al. 2017) contain millions of astronomical spectra, the Gaia mission (Gaia Collaboration et al. 2016) will survey more than billion stars, and Vera C. Rubin Observatory (Ivezić et al. 2019) will observe about 40 billion galaxies and stars. We see that there are and will be so many astronomical observations that most of them will never be seen by astronomers because it would take ages, so it is simply infeasible.

However, large data sets themselves are of little value if they are only stored on some storage devices. We have to process them to gain value. It is usually valuable to annotate all samples from a data set with some annotations, e.g. measure a particular physical property of astronomical objects. Then, we can filter the data set and select a subset of interest for further detailed analysis, e.g. only astronomical objects that belong to a particular category based on the measured physical property. How do we annotate samples from a data set? We imagine that humans can annotate all samples only if the given data set is small enough. However, this is infeasible anymore in the case of large data sets that, for example, comprise millions of astronomical observations, as the examples above illustrate. Therefore, we must rely on automatic methods to process large data sets without much human intervention.

The general situation is that we have an unannotated large data set of samples we want to annotate. We see that we cannot rely on human anno-

tators anymore. We, as humans, usually want to focus on more important tasks and leave the process of annotation to methods that we develop to save time. Human interventions can potentially only support these methods. Now, suppose that we have a suitable method for our task at hand. The method outputs annotations that we call *predictions* in the context of this dissertation. On the other hand, for each sample, there is the *true annotation*. The closer the predictions of a method are to corresponding true annotations, the better the method is. Unfortunately, any method, as well as humans or their combination, can make errors, i.e. annotate samples incorrectly. How can we identify such incorrectly annotated samples in a large data set?

Firstly, some methods can associate their prediction with uncertainty, i.e. their confidence in a prediction, which is called *predictive uncertainty*. Then, someone can check only the predictions associated with high predictive uncertainty. However, we have to ensure that these predictive uncertainties are reliable and that there are ways to verify that. Secondly, we can perform a *consistency check*, i.e. we can employ more than one method to check if their predictions are consistent for the same samples. Then, we are more confident that the predictions are correct. Therefore, we have to develop a diverse set of methods because a larger and more diverse set means more confidence in predictions (assigned annotations).

We see *active deep learning* as a set of such suitable methods that effectively combine *deep learning* (a set of data-driven methods) with *active learning* (the assistance of humans) to save humans as much time as possible while outputting reliable predictions. What are deep learning, active learning, and active deep learning? Why do we see active deep learning as a set of such suitable methods?

First, both active and deep learning are subfields of machine learning. Therefore, active deep learning, as the synergy of those two, is also its subfield. Machine learning methods usually train models using an annotated data set (called traininǵ set) consisting of inputs with corresponding true annotations so that the model can solve a given task.

Deep learning methods train complex (i.e. *deep*) models by composing them from several processing layers. These methods are based on *representational learning*, which desires to extract a suitable representation that makes the task trivial to solve. A deep model is trained to create a hierarchical representation of an input fed into the first layer. Each layer extracts a representation for the next layer. The final layer outputs a prediction. Therefore, we think of deep models as internally producing a representation in the next-to-last layer processed by the last layer. The most straightforward architecture of a deep model is a feedforward *fully connected neural*

network (FCNN), a composition of matrix multiplications, bias additions, and non-linear activation functions. Such deep architectures are trained with backpropagation and variants of stochastic gradient descent algorithms. Due to representational learning and these training algorithms, deep learning is very powerful and solves tasks that have resisted human attempts for years. However, deep learning models will work adequately only if two conditions are met: 1. there is a sufficiently large human-annotated training set, and 2. the training set is representative of the target data set (i.e. the data set we want to process).

It could be problematic to satisfy these conditions. For example, astronomical data sets mainly do not satisfy both conditions. First, large human-annotated data sets that we could use as training sets are scarce in astronomy. Of course, some astronomical observations are manually annotated. However, this applies only to smaller data sets or subsets preselected based on predictions of some methods. However, the performance of machine learning methods depends on human-annotated training sets. Otherwise, if we trained them with predictions of some methods, they would only replicate the methods with all their errors. Overall, in astronomy, we are in a situation where we commonly have only small human-annotated training sets. Moreover, the scarce and small human-annotated training sets are diverse due to different instruments, scientific goals, or observations from different sky parts and depths. Therefore, they are usually not representative of the target data set.

Active learning can solve the problem of the lack of a large and representative human-annotated training set. Active learning is based on the idea that a model will achieve better performance with a smaller training set if samples in its training set are chosen based on the needs of the model itself. Active learning approaches this situation by querying a batch of unannotated samples. Humans manually annotate this batch. Selecting an informative batch for annotation is crucial as it can speed up obtaining a satisfactory model and save time and expenses spent on manual annotation. We want a batch that will improve the performance of the model as much as possible. That means we need a batch that 1. contains samples problematic for the model (i.e. samples with high predictive uncertainty), 2. is diverse, 3. but not redundant. Having an annotated batch, the training set of the model is extended with it, and the model is retrained. This whole process repeats until we are satisfied with the performance of the model.

Now, we know what active deep learning is and why it is a suitable set of methods to approach the general situation described above. It combines deep learning that solves tasks that have resisted human attempts for years and active learning that can bring in human knowledge in an efficient way.

1.2 Goals

The *general goal* of this dissertation was to improve active deep learning to effectively and reliably annotate large data sets with a particular emphasis on large data sets of astronomical spectra. Active deep learning was not applied to astronomical spectra prior to the first publication included in this dissertation, i.e. Škoda, **Podsztavek**, and Tvrdík (2020a). Therefore, the *first goal* was to verify that active deep learning is a suitable set of methods for large data sets of astronomical spectra.

In Section 1.1, we saw that the performance of active deep learning methods depends on batches selected for annotation. First, among others, the batches have to contain samples with high predictive uncertainty. Next, we focused on this point. Therefore, the *second goal* was to develop methods that allow us to select samples with high predictive uncertainty. This included research in methods for predictive uncertainty quantification on tasks related to astronomical spectra.

Nevertheless, how do we know that models produced by the methods quantify predictive uncertainties reliably? The machine learning community has developed many scalar scores to evaluate the reliability of predictive uncertainties. However, scalar scores express only the degree of reliability of predictive uncertainties and give us no clue what is wrong if there are some problems. Therefore, the *third goal* was to develop a method that can help us identify problems with the reliability of predictive uncertainties, if there are some.

1.3 Contributions

The *first contribution* is the first application of an active deep learning method in astronomical spectroscopy (Škoda, **Podsztavek**, and Tvrdík 2020a). To clarify contributions to the publication of the method, its first author, my co-supervisor Dr Petr Škoda, worked mainly on the astronomical part of the publication while I (with the help of my supervisor Prof. Pavel Tvrdík) worked mainly on the active deep learning part.

The *second contribution* is a method for probabilistic prediction of spectroscopic redshift (**Podsztavek**, Škoda, and Tvrdík 2022) based on Monte Carlo (MC) dropout (Gal and Ghahramani 2016a). Here, probabilistic prediction means that we do not only produce point predictions (e.g. a single real number), but we also produce predictive uncertainties. Therefore, the method contributes to predictive uncertainty quantification. It contributes to astronomical spectroscopy as the spectroscopic redshift is predicted from

astronomical spectra.

The *third contribution* is a method for probabilistic prediction of the atmospheric properties of exoplanets (Yip et al. 2022a) based on deep ensembles (Lakshminarayanan et al. 2017). This method was a winning solution to the Ariel Data Challenge 2022 competition. The method is published as part of the publication by Yip et al. (2022a), which describes the results and outcomes of the competition. I, as a co-author of the publication, also describe the method in the publication. Again, this is a contribution specific to astronomical spectroscopy because the properties are predicted mainly from astronomical spectra plus some other auxiliary data.

The *fourth contribution* is a method for automatic miscalibration diagnosis of probabilistic prediction (**Podsztavek** et al. 2024). This is a contribution to predictive uncertainty evaluation and is general to probabilistic models that produce predictive probabilistic distributions.

1.4 Structure

First, to contextualise this dissertation, we will review the literature in Chapter 2. Second, we will develop the active deep learning method for the discovery of astronomical objects of interest in Chapter 3. Third, we research two promising deep learning methods for predictive uncertainty quantification and their applications in astronomy in Chapter 4 and Chapter 5. Then, we develop a method for automatic evaluation of predictive uncertainties that automatically identifies a problem of a model if it has one in Chapter 6. Finally, we summarise the results and make conclusions for future research and implementation in practice in Chapter 7.

Chapter 2

Literature Review

2.1 Active deep learning

The first goal (see Section 1.2) was to verify that active deep learning is a suitable set of methods for large data sets of astronomical spectra. We set the goal because active deep learning was not applied to astronomical spectra prior to our publication, i.e. Škoda, **Podsztavek**, and Tvrdík (2020a). In astronomy, on active deep learning, only the publication by Walmsley et al. (2020) was published during the peer-review process of our publication. Walmsley et al. (2020) used active deep learning to classify the morphology of galaxies from their images.

We have briefly introduced active deep learning through a description of deep learning and active learning in Section 1.1. More information on active deep learning and its application can be found in surveys by Ren et al. (2021), Liu et al. (2022), M. Wu et al. (2022), and Wan et al. (2023) that were all published after our publication. Next, we review relevant literature concerning deep learning to select an appropriate deep model and active learning applications in astronomy.

2.1.1 Deep learning

Throughout this dissertation, we experiment with two kinds of deep models, specifically feedforward neural networks: 1. *fully connected neural networks* (FCNNs) and 2. *convolutional neural networks* (CNNs). FCNNs were described in Section 1.1. CNNs are introduced next after two general paragraphs on machine learning.

In machine learning (see Bishop 2006), we usually want to train *model parameters* θ using an annotated data set $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^N$ consisting of N pairs of *input vectors* $\{\boldsymbol{x}_i\}_{i=1}^N$ with corresponding *target values* $\{y_i\}_{i=1}^N$ so that the

model can solve a given task. There are two fundamental types of tasks: 1. *classification* tasks where a target value has a value from a finite set of C discrete classes, i.e. $y_i \in \{1, \ldots, C\}$; and 2. *regression* tasks where a target value is a real number, i.e. $y_i \in \mathbb{R}$.

To properly train, validate, and test a model, we split a given data set into training, validation, and test sets. Usually, we iteratively show the training set to the model during training to train its model parameters. The model (that depends on its parameters) y_θ takes the input vector $\boldsymbol{x}_i$ and outputs a *prediction* $\hat{y}_i = y_\theta(\boldsymbol{x}_i)$. We want the prediction $\hat{y}_i$ to be equal to its target value y_i. The validation set is used to optimise the model complexity controlled by *hyperparameters*. Finally, the performance of the trained model is evaluated on the test set. At this phase, model parameters and hyperparameters are fixed. The test set represents new data to which the model will be applied in the future, so it tests the ability of the model to generalise to unseen input vectors.

CNNs (LeCun et al. 1989) are state-of-the-art deep models for many (especially computer vision) tasks. They started to be recognised when a CNN named AlexNet (Krizhevsky et al. 2012) won the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) 2012 (Russakovsky et al. 2015). They usually consist of convolutional, pooling, and fully connected layers and are mainly designed to process images. However, CNNs can process any data with a grid structure. Therefore, we can take advantage of CNNs because spectra have a grid structure. An astronomical spectrum can be viewed as a 1-dimensional array of flux values, whereas a typical image is a 3-dimensional array of RGB channels. A typical CNN consists of two parts: convolutional layers with pooling layers followed by fully connected layers.

Convolutional layers perform mathematical convolution, i.e. they convolve their inputs with trained *kernels*. They leverage three essential properties of CNNs: 1. *sparse interactions* (a kernel, i.e. a convolutional layer, has fewer parameters than a fully connected layer); 2. *parameter sharing* (rather than having a separate set of parameters for each possible location of a given object, a convolutional layer has one set for all locations); and 3. *equivariance to translation* (if an object shifts in the input, its corresponding output shifts by the same distance vector). The output of each convolutional layer is passed through a non-linear activation function. Nowadays, the most used non-linear activation function is the *rectified linear function* (ReLU):

$$\mathrm{ReLU}(a) = \max(a, 0).$$

A series of convolutional layers might alternate with pooling layers. The pooling layers make the representation invariant to small translations and

rotations in the input. Moreover, they reduce the size of the representation, and therefore, they reduce the number of model parameters of fully connected layers. In this dissertation, we use *max pooling layers* that reduce the representation by applying a filter to it that selects only the maximal value from a predefined set of adjacent pixels. The representation from the convolutional and pooling part is passed through the fully connected part. In regression, the last layer usually does not perform any non-linear activation function and only produces predictions. In classification, the last layer usually performs the softmax function:

$$\text{softmax}(\boldsymbol{z}_i)_j = \frac{\exp(z_j)}{\sum_{k=1}^{C} \exp(z_k)},$$

where $\boldsymbol{z}_i = (z_1, \ldots, z_C)$ is an input to the last layer. Then, the output vector $\text{softmax}(\boldsymbol{z}_i)$ of the layer can be interpreted as a probability vector, i.e. a vector with non-negative elements that sum up to one. The matching prediction $\hat{y}_i$ equals to the index of maximal value of the output vector $\text{softmax}(\boldsymbol{z}_i)$:

$$\hat{y}_i = \arg\max_{j \in \{1,\ldots,C\}} \text{softmax}(\boldsymbol{z_i})_j.$$

CNNs were successfully applied to many astronomical tasks. For example, Aniyan and Thorat (2017), Domínguez Sánchez et al. (2018), and Alhassan et al. (2018) used CNNs to automate the morphological classification of radio sources. Alger et al. (2018) localised host galaxies for a given radio component with a CNN using data from experts and crowdsourced training data. Furthermore, George and Huerta (2018) applied two CNN time-series data to the detection and parameter estimation of gravitational waves from binary black hole mergers. The two CNNs achieved a similar performance as previous advanced methods but were much faster, thus allowing real-time processing.

For all these reasons, we decided to employ CNNs as deep models for the astronomical tasks in this dissertation.

2.1.2 Active learning

Active learning with shallow models (i.e. not deep models) has also been successful in astronomy. It was applied to estimate the parameters of stellar population synthesis models (Solorio et al. 2005) or the classification of light curves of variable stars (Richards et al. 2012). Astronomers used active learning to learn a model for photometric data classification from spectroscopic data (Gupta et al. 2016; Vilalta et al. 2019) and to minimise the number of required spectroscopically confirmed annotations in preparing training sets for

the photometric classification of supernova light curves (Ishida et al. 2019a). Furthermore, active learning was shown to perform well in anomaly detection in light curves of supernovae (Ishida et al. 2019b). These applications prove the potential of active learning for astronomy. We take a step forward and use a deep model instead of a shallow one, which brings new challenges.

2.2 Predictive uncertainty quantification

The second goal (see Section 1.2) was to develop methods that allow us to select samples with high predictive uncertainty.

Deep models (i.e. FCNNs and CNNs) in their simple forms do not provide predictive uncertainties. Therefore, there is no way to distinguish whether a model produces reasonable predictions or random guesses. For example, suppose there is an input vector insufficiently represented in the training set. Therefore, there will be uncertainty if model parameters were trained correctly. This type of uncertainty is *model* or *epistemic uncertainty*. It can be reduced using more training data. We want the model to try to generalise to the insufficiently represented input vector, but, at the same time, we expect that the model will indicate this by producing a higher predictive uncertainty. Furthermore, there is another source of uncertainty in machine learning. Noisy input vectors or target values cause *data* or *aleatoric uncertainty*. Data uncertainty cannot be reduced even if we enlarge the training set with more data. Ideally, we should incorporate both model uncertainty and data uncertainty into the predictive uncertainties of the model.

In the machine learning community, there is a significant amount of work to develop methods to train probabilistic deep models, i.e. models that produce predictive uncertainty. The fundamental probabilistic deep models are density networks (DNs) (Nix and Weigend 1994) and mixture density networks (MDNs) (Bishop 1994). A DN outputs a Gaussian distribution, i.e. its mean and variance. An MDN usually outputs a mixture of Gaussian distributions, i.e. their means, their variances and weights of the mixture. Advanced methods are well summarised in a survey by Gawlikowski et al. (2023).

We chose to base our methods (for predictive uncertainty quantification on astronomical spectra) on two methods that are promising in terms of calibration improvement (Gawlikowski et al. 2023, p. S1557)[1] while both are relatively simple to implement, which is an important for further practical

[1] In 2021, the survey by Gawlikowski et al. (2023) was initially released on arXiv. In 2023, it was published again in Artificial Intelligence Review. Therefore, we had access to the survey before our publications on predictive uncertainty quantification (i.e. **Podsztavek**, Škoda, and Tvrdík 2022; Yip et al. 2022b).

implementations in astronomy:

- a Bayesian method based on Monte Carlo dropout (Gal and Ghahramani 2016a), and
- an ensemble method based on deep ensembles (Lakshminarayanan et al. 2017).

2.2.1 Monte Carlo dropout

The Monte Carlo (MC) dropout method (Gal and Ghahramani 2016a) uses dropout (Srivastava et al. 2014) as a Bayesian approximation that brings theoretically justified predictive uncertainties into deep learning. Gal and Ghahramani (2016a) showed that training neural networks with dropout is equivalent to *Bayesian variational inference*. The original idea behind dropout is to randomly set outputs of neurons to 0 with a given probability p during training. Dropout prevents overfitting because neurons are not allowed to co-adapt too much.

In Bayesian variational inference, we model the true *posterior distribution* (used to get the true predictive distribution) with a *variational distribution* (i.e. an approximate posterior distribution) that is computationally feasible. Kullback–Leibler divergence is used to get the variational distribution as close to the true posterior distribution as possible. Because of the equivalence, we still minimise a loss function with respect to model parameters θ. If we have a regression task, we use the *mean squared error loss* function (hereafter MSE loss) that is defined as:

$$\mathcal{L}_{\text{MSE}}(\theta) = \frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2 + \lambda\|\theta\|^2, \tag{2.1}$$

where $\|\theta\|^2$ is the L_2 regularisation with the weight decay λ hyperparameter that controls overfitting by discouraging the model parameters θ from taking large values. Then, we also use dropout while making predictions (i.e. we execute *stochastic forward passes*) to sample T predictions $\{y_\theta(\boldsymbol{x}_i)^{(1)}, \ldots, y_\theta(\boldsymbol{x}_i)^{(T)}\}$. These are empirical samples from approximate predictive distributions. We can estimate the predictive mean of the approximate predictive distribution for input vector $\boldsymbol{x}_i$ by the *sample mean*:

$$\hat{y}_i = \frac{1}{T}\sum_{j=1}^{T} y_\theta(\boldsymbol{x}_i)^{(j)}.$$

This estimate is referred to as *MC dropout* in literature. Furthermore, we can represent the predictive uncertainty of the approximate predictive distribution by its predictive variance and estimate it with the *sample variance*:

$$s_i^2 = \frac{1}{T-1} \sum_{j=1}^{T} (y_\theta(\boldsymbol{x}_i)^{(j)} - \hat{y}_i)^2.$$

To get Bayesian CNNs, dropout has to be performed at each layer with model parameters (Gal and Ghahramani 2016b). Therefore, dropout is applied to both convolutional and fully connected layers of the model while training and making predictions.

There were a few applications of MC dropout in astronomy. Levasseur et al. (2017) employed a Bayesian CNN to estimate the predictive uncertainties of lensing parameters. Intelligent exoplaNet Atmospheric RetrievAl (Soboczenski et al. 2018) is a preliminary experiment with a Bayesian CNN applied to synthetic spectra of exoplanets to predict the temperature, structure, and composition of their atmospheres. H. W. Leung and Bovy (2018) used a Bayesian CNN to predict stellar parameters from the Apache Point Observatory Galactic Evolution Experiment spectra with predictive uncertainties to cope with noisy and missing flux values. They introduced a custom loss function and a complex system composed of a large CNN and small neural networks, both using MC dropout. Möller and Boissière (2019) experimented with MC dropout applied to a recurrent neural network. They evaluated their method on the classification of supernovae using simulated light curves. In the Galaxy Zoo project, Walmsley et al. (2020) combined Bayesian CNN with active learning to reduce the amount of needed annotated data. Killestein et al. (2021) employ a Bayesian CNN to classify transients in images.

2.2.2 Deep ensembles

Deep ensembles (Lakshminarayanan et al. 2017) combine outputs of M probabilistic models, specifically probabilistic neural networks. Those neural networks can be simple DNs or MDNs. The weights of the neural networks are randomly initialised to diversify the ensemble. Further diversifications can be achieved by training them on random subsets of a given training set. We see that the neural networks are independent, so they can be trained in parallel. The output of the ensemble is a uniformly weighted mixture model. For example, if the neural networks produce Gaussian distributions, the prediction is a uniformly weighted mixture of Gaussian distributions.

2.3 Predictive uncertainty evaluation

The third goal (see Section 1.2) was to develop a method that can help us identify problems with the reliability of predictive uncertainties. We limit ourselves to regression tasks and leave classification tasks for future research.

A key to predictive uncertainty evaluation is the paradigm of maximising the *sharpness* of predictive distributions (that commonly represent predictive uncertainties) subject to their *calibration* (Gneiting et al. 2007). Sharpness means the concentration of probability distributions, while calibration means statistical consistency with corresponding target values.

Proper scoring rules are commonly employed to evaluate predictive uncertainties. A scoring rule is a loss function for predictive distributions, as opposed to point predictions. It is *proper* if it has the property that a predictive distribution that matches the true data-generating distribution minimises the expected score. Implicitly, that property means that a proper scoring rule measures calibration and sharpness jointly. The two most used proper scoring rules are the *negative log-likelihood* (NLL):

$$\mathrm{NLL}(f_i, y_i) = -\log f_i(y_i),$$

where f_i denotes the predictive probability density function (PDF) corresponding to the target value y_i, and the *continuous ranked probability score* (CRPS):

$$\mathrm{CRPS}(F_i, y_i) = \int_{-\infty}^{\infty} (F_i(a) - \mathbf{1}_{a \geq y_i})^2 da.$$

where F_i denotes the predictive cumulative distribution function (CDF) corresponding to the target value y_i.

We need to use a proper scoring rule to measure and confirm the improvement in predictive performance to compare two probabilistic models. However, scalar scores such as proper scoring rules or the *calibration error* (Kuleshov et al. 2018) are insufficient to fully evaluate predictive uncertainties because there can be problems that scalar scores would not reveal. These problems might cause *miscalibration*, and scalar scores express only the degree of miscalibration, not its cause. Therefore, we have to use other tools to evaluate predictive uncertainties.

First, we must specify what we mean by calibration, specifically probabilistic calibration. We define *probabilistic calibration* following Gneiting et al. (2007). At an instance $i \in \{1, \ldots, N\}$, nature chooses a true data-generating distribution G_i, and a model picks a predictive CDF F_i. Both

G_i and F_i might depend on stochastic parameters. The predictive distributions are probabilistically calibrated relative to the true data-generating distributions if

$$\frac{1}{N}\sum_{i=1}^{N} G_i \circ F_i^{-1}(p) \to p$$

for all $p \in (0, 1)$, where the arrow denotes the almost sure convergence as $N \to \infty$. This definition is equivalent to the uniformity of distribution of *probability intergral transform* (PIT) values:

$$\{F_i(y_i) \mid i \in \{1, \ldots, N\}\},$$

where the target value y_i is an observed value of a random variable with the distribution G_i. The PIT is translation- and scale-invariant. We diagnose miscalibration by visualising the histogram of PIT values (hereafter the *PIT histogram*) and inspecting its shape.

In the machine learning literature, the PIT histogram or *calibration plot* (also known as the *reliability diagram*) are standard tools to diagnose miscalibration. These two tools are equivalent because both display an estimate of the PIT distribution: the PIT histogram shows a density estimate, whereas the calibration plot displays an estimate of the CDF.

One should be able to diagnose miscalibration by visually inspecting a PIT histogram or calibration plot. However, understanding the cause of miscalibration requires much experience. Simple causes of miscalibration (*bias*, *underdispersion* and *overdispersion*) can be identified easily. They express themselves respectively as a PIT histogram with a single peak at an edge, a U-shaped and a bell-shaped PIT histogram (see visual guide in Figure 2.1). However, suppose the cause of miscalibration is not a simple one or multiple causes co-occur. In that case, the potential shapes of PIT histograms cannot be easily enumerated, which makes their interpretation difficult or even impossible for inexperienced users.

Therefore, we provide a user-friendly interpretation of PIT histograms, from which users can recognise causes of miscalibration. Subsequently, users can deal with those causes and get more reliable predictive distributions.

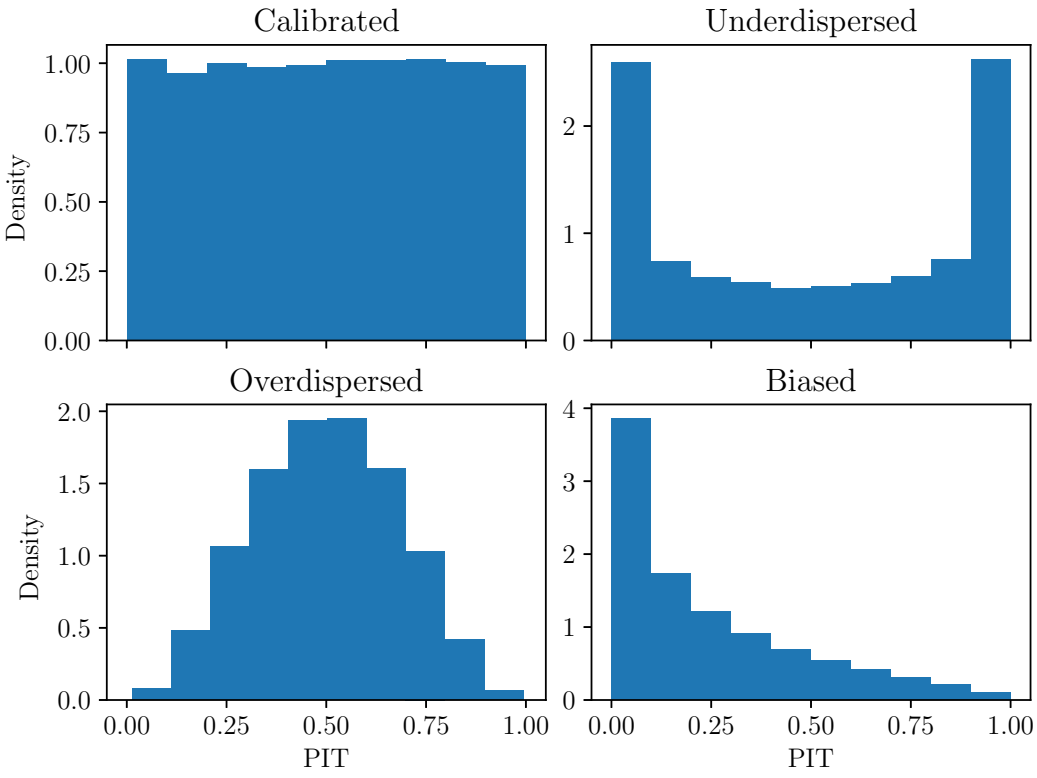

Figure 2.1: Visual guide to interpretation of PIT histograms

Chapter 3

Method for Discovery of Objects of Interest[1]

The first goal (see Section 1.2) was to verify that active deep learning is a suitable set of methods for large data sets of astronomical spectra. We prove that by presenting an active deep learning method that allowed us to annotate (i.e. discover) rare objects of interest in a large data set of spectra although only a small training set was available from a different data set of spectra.

3.1 Astronomical motivation

The stellar spectral classification, as explained in Gray and Corbally (2009), is an important astrophysical task of assigning a particular annotation (mixture of letters and Arabic and Roman numbers), called the spectral class, to each spectrum based on the visual similarities (e.g. presence, strength, and width of the spectral lines of a given element, or a combination of multiple lines). A common automatic procedure (see e.g. Gray and Corbally 2009, Chap 13.5) uses statistical matching (mainly using χ^2 fitting) of a given spectrum with an extensive set of template spectra that may be either synthetic or come from a library of carefully selected stars (called spectral standards). This method is also used in various modifications for the automatic spectral classification of large surveys, such as the Sloan Digital Sky Survey (SDSS) (Y. S. Lee et al. 2008) or Large Sky Area Multi-Object Fiber Spectroscopic Telescope (LAMOST) (Y. Wu et al. 2011; Y. S. Lee et al. 2015).

[1]This chapter is based on P. Škoda, **O. Podsztavek**, and P. Tvrdík (2020a). "Active deep learning method for the discovery of objects of interest in large spectroscopic surveys". In: *Astronomy & Astrophysics*. DOI: 10.1051/0004-6361/201936090.

A problem arises in many cases when appropriate model of a spectrum is not known and the library used for matching is not rich enough to contain unusual or new types. In addition to this, many types of celestial objects may show complex shapes of only several prominent spectral lines (mainly Hα or other Balmer and Paschen lines) that cover only small parts of the whole spectrum. The integral statistics then fails, and target-tailored methods must be applied to discover such usually rare objects. This is the case of various objects with emission lines that allow us to study a wide range of interesting physical processes.

Pre-main-sequence stars such as young stellar objects and T Tau stars (Reipurth, Pedrosa, and Lago 1996; Kurosawa, Harries, and Symington 2006), or hot stars with expanding envelopes or strong winds show prominent emission lines, as do cataclysmic variables, novae, and even late-type stars with chromospheric activity. See Kogure and K.-C. Leung (2007) or Traven et al. (2015) for a comprehensive overview of these cases.

We showcase an active deep learning method on the discovery of classical Be stars (Porter and Rivinius 2003) and rare B[e] stars (Zickgraf 2003). Be and B[e] stars have complicated emission-line profiles that often look like symmetric or slightly asymmetric double-peaks, sometimes superimposed on absorption lines, depending on their disk geometry (Silaj et al. 2010). The manual annotation of their profiles (Hanuschik, Kozok, and Kaiser 1988) is a challenging task even on small samples, but it becomes impossible in large data sets of spectra. The classical method to finding emission lines is to compute integral statistics around their expected positions. It is similar to the standard method of measuring the line equivalent width (Kang and S.-G. Lee 2012; Waters and Hollek 2013). Such an integral measure based on three-pixel statistics was taken by Lin et al. (2015) on the LAMOST data release (DR) 1 to find strong uprising peaks. This resulted in a catalogue of 203 emission-line stars, 23 of which were identified as classical Be stars and 180 are claimed to be discovered candidates. To find double-peak profiles hidden in deep absorption, Hou et al. (2016) used a more advanced method based on the difference of several statistics with different kernel width. They made an extensive analysis of the LAMOST DR2 and published a catalogue of 11 204 spectra of emission-line stars.

We propose an alternative method for the discovery of emission-line spectra here based on active deep learning. For the sake of simplicity, we limit ourselves to the vicinity of the Hα line. Next, we describe the first systematic investigation of the LAMOST DR2 using a convolutional neural network (CNN) in combination with active learning, i.e. an active deep learning method.

3.2 Active deep learning method for discovery of objects of interest

The discovery of objects of interest in large data sets of spectra would be a standard machine learning task if a large and representative training set of a given large data set was available. With such a training set, it would be straightforward to train a machine learning model and classify the data set with high accuracy. However, our experiments showed that if there is not such a training set, standard machine learning methods provide poor results with a high rate of both false and missed candidates.

This means that if the training set is not a sufficient representation of a target data set, for example, when the training set is biased or comes from another, but similar instrument, other machine learning methods need to be developed to obtain reasonable discovery results. We propose and evaluate here a classification method based on extension of a CNN with class balancing and active learning. Next, we explain in detail why and how we combined a CNN with a class balancing algorithm and active learning. This unified active deep learning method allowed us to discover objects of interest (objects with emission-line spectra) in the LAMOST DR2 altough only a small training set was available from a different data set of spectra.

In Subsection 2.1.1, we showed that CNNs are state-of-the-art deep models for data with grid structure including spectra and that they were successfully applied to astronomical tasks. Therefore, we use a CNN as the deep model in the active deep learning method.

When discovering rare objects of interests in large data set of spectra, we face the class imbalance problem. Annotated spectra of rare objects of interest (hereafter target spectra) will usually be in the minority, in contrast to annotated spectra of abundant objects (hereafter non-target spectra). Therefore, the training set will tend to be imbalanced. Moreover, target spectra will be in a significant minority in general large data sets of spectra. Our application of the active deep learning method revealed exactly the class imbalance problem. The archive of the Ondřejov 2 m Perek telescope is focused on the observation of emission-line stars. Although there is almost the same percentage of single-peaks as absorptions, double-peaks are still in the minority. Moreover, there are (at least by order of magnitude) fewer emission-line spectra than standard ones in the LAMOST DR2 because emission-line objects are rare in the universe. In this case, class balancing is an essential part of workflows and leads to successful performance. For example, we refer to Calleja et al. (2011) or Lyon et al. (2016) for the necessity of class balancing in astronomy and Rastgoo et al. (2016) in medicine. To overcome the fact

that CNNs will tend to discriminate the minority classes, we incorporate the *synthetic minority over-sampling technique* (SMOTE) proposed by Chawla et al. (2002). This technique allows enlarging the number of annotated target spectra to the same size as the more abundant non-target spectra.

Our experiments showed that the combination of a CNN and class balancing is still not sufficient for the discovery of objects of interest because the first prediction of candidates delivered a considerable amount of false candidates and featureless noisy spectra. The reason for this failure was an imperfect training set. Therefore, we decided to explore active learning to circumvent the requirement of good representativeness of the trainig set to exploit the full potential of deep models to discover objects of interest.

In the case of large data sets of spectra, there are huge pools of unannotated samples that can be processed and gathered at once (a *pool-based setting*). Spectra are queried selectively from the pool according to an informativeness measure that evaluates all spectra in the pool. Concerning CNNs for classification, the most straightforward *query strategy* is the *uncertainty sampling*. This strategy selects spectra for which the CNN provided the least certain annotations because the last layer of a CNN usually performs a softmax function (see Subsection 2.1.1). Therefore, to query spectra for annotation, for all the spectra in the pool, we compute the *information entropy*:

$$H(\boldsymbol{z}_i) = -\sum_{j=1}^{C} \text{softmax}(\boldsymbol{z}_i)_j \ln(\text{softmax}(\boldsymbol{z}_i)_j),$$

where $\boldsymbol{z}_i$ is an input to the last layer of the CNN. Then, the query strategy selects spectra with the highest information entropy.

Because the training of a CNN can be time-consuming, our method uses *batch mode* active learning, which iterates in cycles. We annotate a batch of queried samples in each iteration to save time and computational resources (i.e. training of a CNN). More specifically, the method selects a batch of a previously specified size from all spectra in the pool, and we manually annotates them. Then, we add all the manually annotated spectra to the training set, so that it contains training samples from the previous iterations and newly annotated spectra.

Lastly, to decide when to stop the active learning iterative procedure, we need to track the performance of the CNN. The obvious possibility is to estimate a performance measure and stop learning when a plateau is reached (i.e. when adding newly annotated spectra to the training set would not increase the performance of the CNN).

When a large pool of unannotated samples contains a negligible amount of target spectra, it is reasonable to estimate *precision*, that is the ratio of

correctly predicted target spectra and all (both correctly and incorrectly) predicted target spectra. In the case of precision, we can expect that a random sample of spectra classified into target classes will contain the true target spectra. On the other hand, a random sample of all spectra or non-target spectra will probably contain only non-target spectra. Therefore, an estimation of any performance based on such random samples will not yield a useful result. For example, an estimate of accuracy, which has to be based on a random sample of all spectra, will almost certainly be 1 or very close to it. Moreover, when discovering rare objects, we are not interested in accuracy, but rather in precision and recall. Recall is the ratio of correctly predicted target spectra and all target spectra. However, the estimation of recall faces the same problem as the estimation of accuracy. For this reason, we cannot have any randomly sampled performance estimation set fixed for all iterations. We have to sample a new random sample in every iteration as the set of predicted target spectra is changing.

In summary, our active deep learning method takes an annotated data set as the initial training set and balances it. Having a balanced training set, we train the CNN and use the trained CNN to classify all spectra in the unannotated pool. Then, we use the uncertainty sampling query strategy to obtain a batch of samples. We annotate the batch. The annotated samples are taken out of the unannotated pool and placed into the training set. We repeat these steps until the performance of our CNN is satisfactory. When we are satisfied with the CNN performance, the unannotated samples that were lastly predicted as target ones become new candidates. Finally, we move the samples manually annotated as targets from the training set to the candidate set. The flowchart in Figure 3.1 illustrates all steps of our active deep learning method.

3.3 Experiments

To illustrate the application of our active deep learning method, we have performed experiments with the discovery of objects with signatures of Hα emission in the LAMOST DR2 using annotated data set from the Ondřejov 2 m Perek telescope.

3.3.1 Data

The archive of spectra obtained with 700 mm camera in the Coudé spectrograph of the 2 m Perek telescope at the Ondřejov observatory of the Astronomical Institute of the Czech Academy of Sciences (hereafter *CCD700*

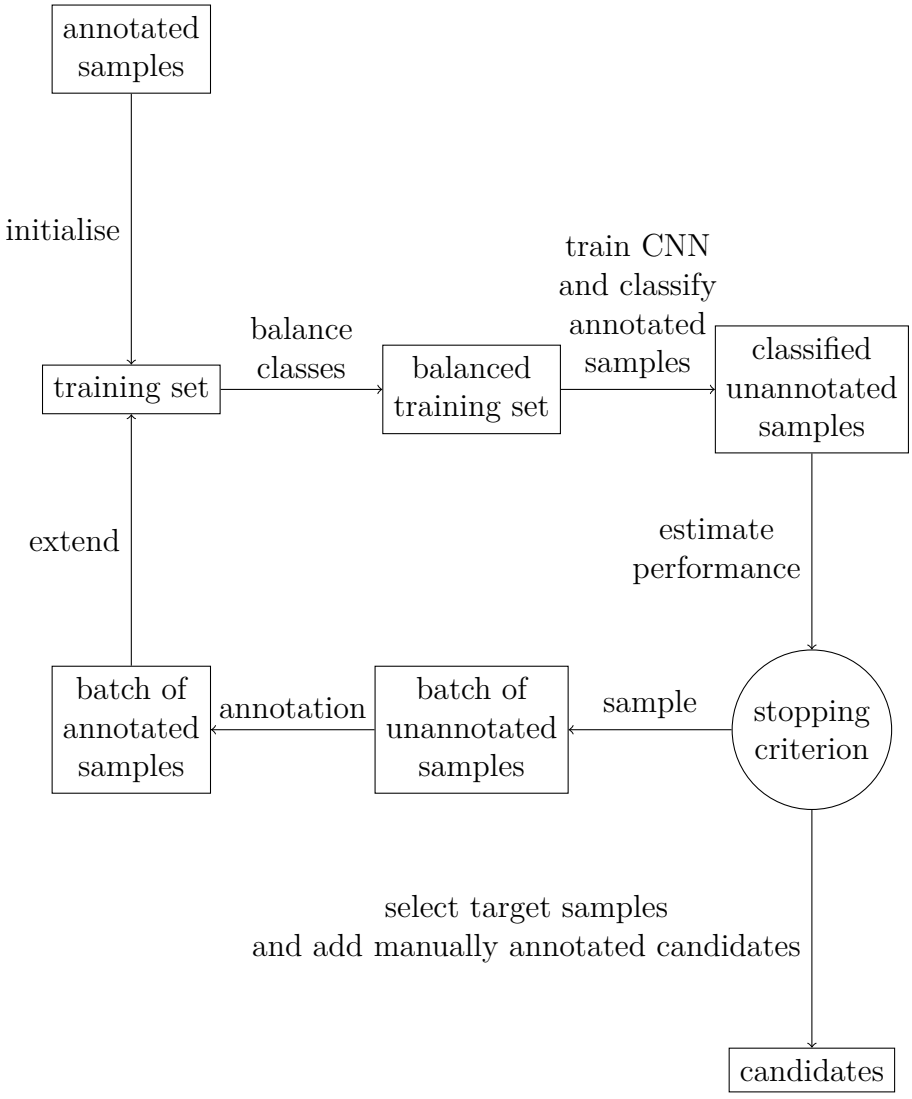

Figure 3.1: Flowchart of active deep learning method

set) is a unique source of spectra of emission-line stars (mostly Be and B[e] stars, stars with strong winds and several novae). This continuously growing archive, currently contains about 17 000 spectra, the majority of which (more than 13 000) are exposed in spectral range 6 250–6 700 Å with a spectral resolving power of about 13 000. The standard Image Reduction and Analysis Facility (IRAF) procedure reduces the spectra, including the calibration in air wavelengths and heliocentric correction.

LAMOST has delivered one of the currently largest collections of spectra. Four thousand fibres positioned by micro-motors feed 16 spectrographs. Its publicly available DR2 contains over four million spectra with a spectral resolving power of about 1 800, covering the range 3 690–9 100 Å. The LAMOST pipeline (Y. Wu et al. 2011) automatically assigns an estimated spectral class to spectra. However, the pipeline uses classification mostly based on the global shape and integral properties of a spectrum in given band-passes using a set of predefined templates. The local features (e.g. detailed line profiles) are ignored. Strong narrow emissions can even be rejected by the pipeline as possibly spoiled pixels. Therefore, we did not use the assigned spectral classes. Hereafter we call the set of all unannotated LAMOST DR2 spectra the *LAMOST pool*. The spectral axis of the Flexible Image Transport System (FITS) files in the LAMOST DR2 are expressed in the logarithm of the vacuum wavelength.

3.3.2 Data preparation

A common assumption in machine learning is that the training set (i.e. the CCD700 set) and target data set (i.e. the LAMOST pool) come from the same probability distribution (Pan and Yang 2010). However, we are interested in the classification of the LAMOST pool using a training set created from the CCD700 set, which contains mostly emission spectra. This means that the training set is highly biased. The distribution mismatch between the training set and the target data set is a well-known problem in machine learning and is called *domain adaptation* (Glorot et al. 2011).

Using the technology of the Virtual Observatory for cross-matching, we have identified only 22 spectra that were observed both by the Ondřejov 2 m Perek Telescope and LAMOST. Only a few (e.g. BT CMi, HD 53 416, or V395 Aur) of them show emission lines. The lack of annotated samples in the LAMOST pool prevents a straightforward usage of machine learning. To use the CCD700 set as our training set, we therefore applied a domain transfer to the spectra from the CCD700 set (based on optical engineering procedures), so that they will look as if they were exposed with the LAMOST spectrograph. Taigman et al. (2017) claims that domain transfer is useful

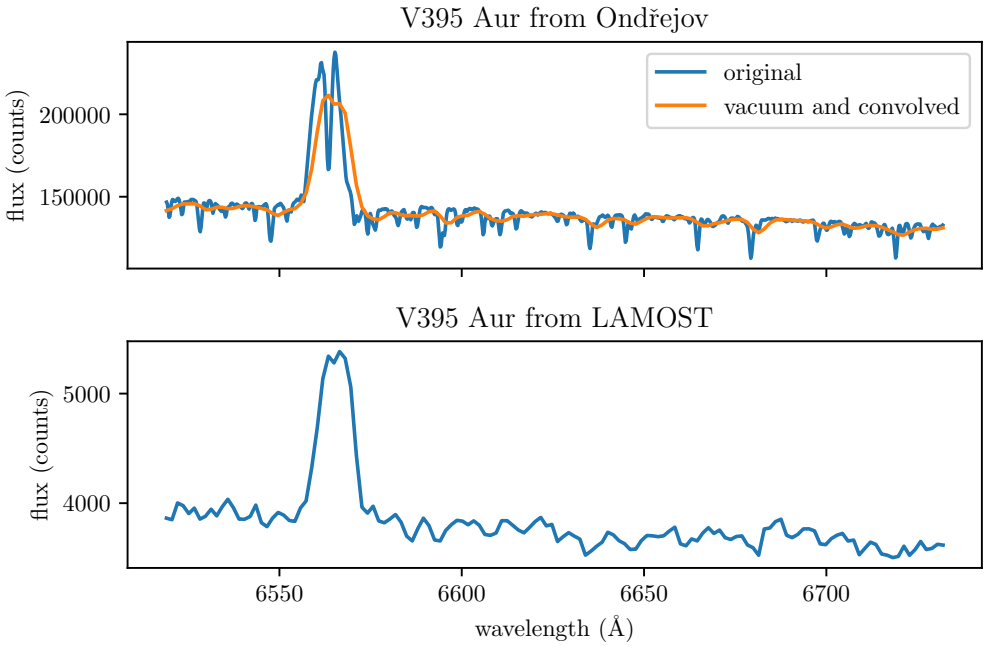

Figure 3.2: Comparison of a spectrum from the LAMOST pool with a spectrum from the CCD700 set transferred to the LAMOST lower resolution and vacuum wavelengths

when solving the domain adaptation problem.

Firstly, we applied air-to-vacuum wavelength conversion to spectra from the CCD700 set using formulas provided in Heiter (2014) because spectra from the CCD700 set are in air wavelengths, but the LAMOST spectra use vacuum wavelengths. Additionally, we converted the vacuum wavelengths of spectra from the LAMOST pool from the logarithmic to linear scale. Secondly, because spectra from the CCD700 set have a higher spectral resolution than the spectra from the LAMOST pool, we applied the spectral resolving power degradation to spectra from the CCD700 set, roughly approximated by the convolution with the Gaussian kernel of a given pixel width to reduce the high-resolution details. Comparison figures of simulated spectra from the CCD700 set and LAMOST pool of all 22 objects mentioned above showed that the standard deviation of seven pixels works best. Figure 3.2 shows the comparison of a spectrum from the CCD700 set, cross-matched spectrum from the LAMOST pool, and the trasferred spectrum.

Next, the CNN requires a vector of features as an input. To have the same features for all spectra, they need to be resampled to obtain the measure-

ment in the same wavelengths across all spectra. We decided to use a linear interpolation to 140 uniformly distributed wavelengths in the spectral range between 6 519 and 6 732 Å. We used this number of points because the LAMOST spectra mostly have this number of measurements in the range. We derived the range from the fact that our classification is based on the Hα line and most of the spectra from CCD700 set are exposed in this range. This range also contains He I 6 678 Å line, which is important in Be stars. Having resampled all spectra in the same wavelength points, we can create a matrix required for training, where rows are 140-dimensional feature vectors of spectra and columns contain fluxes in specified wavelengths.

The last step of data preparation is the min-max normalisation of the spectral flux into a unit-less range $[-1, 1]$ using the equation

$$\boldsymbol{x}'_i = 2\frac{\boldsymbol{x}_i - \min(\boldsymbol{x}_i)}{\max(\boldsymbol{x}_i) - \min(\boldsymbol{x}_i)} - 1,$$

where $\boldsymbol{x}_i$ is an original spectrum and $\boldsymbol{x}'_i$ is its scaled spectrum. Thus, each spectrum has a maximum flux of value 1 and a minimum of value -1. We applied this data preparation procedure for two reasons: 1. we would like to classify the spectra according to their shapes (this procedure effectively suppresses the differences in intensities); and 2. it transforms values in the comfortable small-valued range that is suitable for a neural network training (this is not a feature scaling, but a scaling across each spectrum).

The transffered spectra from the CCD700 set were manually annotated by **Podsztavek** (2017) according the visual shape of the Hα into three classes: single-peak, double-peak, and absorption. The annotated spectra resulted in a data set of 12 936 annotated spectra (hereafter Ondřejov data set) (**Podsztavek** and Škoda 2019) that is suitable for machine learning. The counts of spectra in classes are the following:

- single-peak: 5 301 spectra (40.98%),
- double-peak: 1 533 spectra (11.85%), and
- absorption: 6 102 spectra (47.17%).

Figure 3.3 displays representatives of each class. In both single-peak and double-peak spectra the Hα line is in emission, and the difference between the two classes is in the number of peaks, which are clearly visible in the spectrum. Spectra in the single-peak and double-peak classes are the target spectra of our interest, and as expected, their number is smaller than the number of non-target absorption spectra, which are not interesting for us.

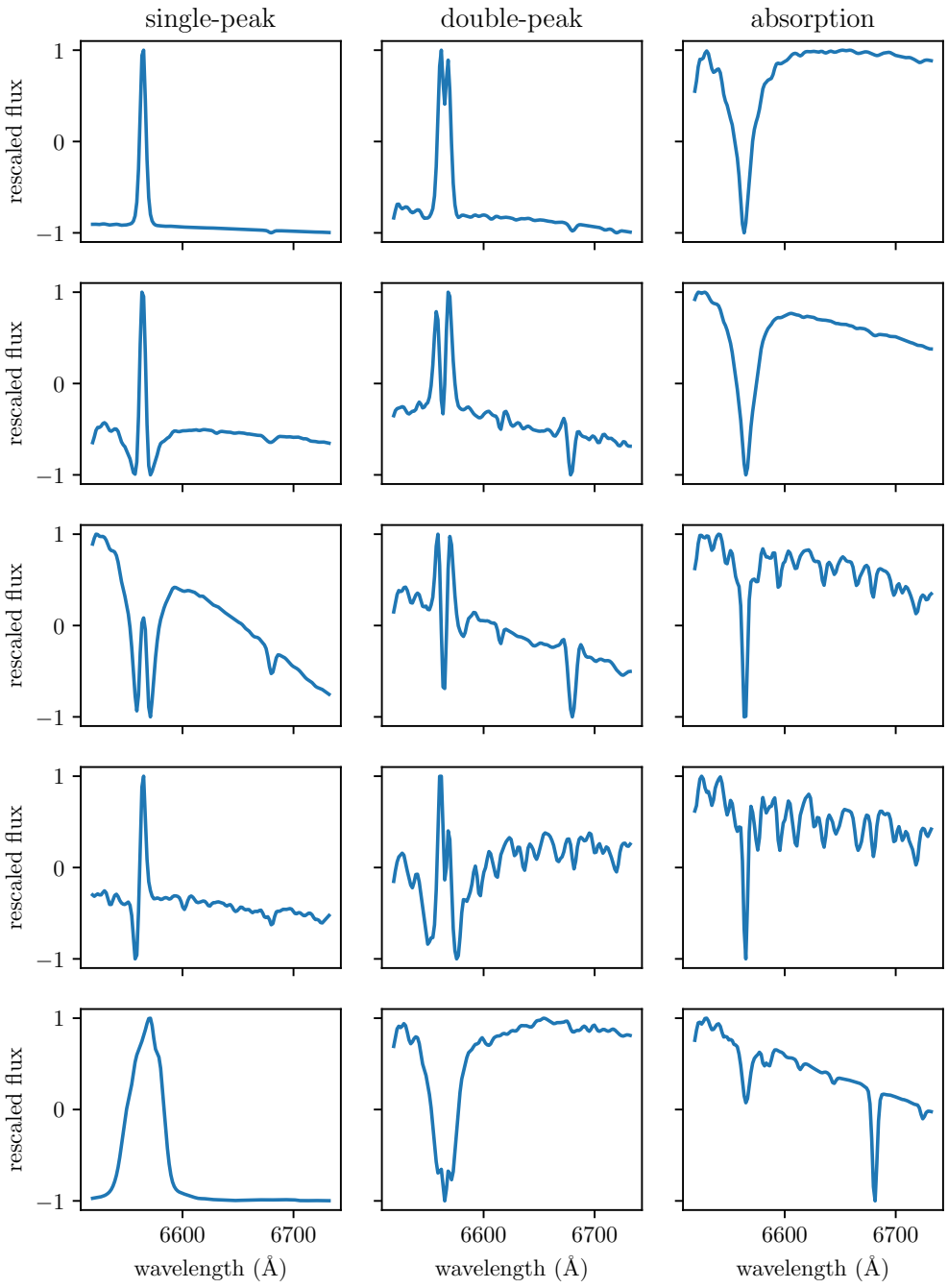

Figure 3.3: Exemplary spectra from Ondřejov data set

The Ondřejov data set contains only well-exposed spectra, while the LAMOST pool contains many noisy spectra with instrumental and reduction artefacts, spectra without peaks or absorption, and spectra with low signal-to-noise ratio. During our experiment, we placed all these spectra into the non-target uninteresting class. Therefore, the non-target-not-interesting class contains bad and absorption spectra, which are both uninteresting for us.

3.3.3 Application of active deep learning method

When the data were ready, we applied our method. We chose the architecture of a CNN as developed in previous work that proved to be working well (see **Podsztavek** 2017). This CNN architecture was inspired primarily by VGG Nets (Simonyan and Zisserman 2015). However, VGG Nets were designed to process multi-channel two-dimensional images. Therefore, we adapted the architecture to our one-dimensional spectra (replace two-dimensional convolutions with one-dimensional convolutions). After several experiments, we converged to the architecture shown in Table 3.1. This CNN was implemented using TensorFlow (Abadi et al. 2016) through the Keras (Chollet et al. 2015) high-level interface and was run on an NVIDIA GTX980 GPU (4 GB memory, 2 048 CUDA cores). The network was trained with the Adam optimiser (Kingma and Ba 2015) in its default setting. The best-found weights were restored at the end of each training. We stopped the training when the categorical cross-entropy loss function was not improved by at least 10^{-4} during the last ten iterations.

After we trained the CNN with the Ondřejov data set (the initial training set) balanced with SMOTE, we used the model to predict classes and probabilities of classes for all spectra in the LAMOST pool. From all the classified spectra, a batch of 100 spectra with the highest information entropy computed from the probabilities of classes was selected (the uncertainty sampling strategy) and manually annotated by us. Then, all the 100 manually annotated spectra were moved to the training set and removed from the LAMOST pool. Hence, after the first iteration, the training set contained the spectra from the Ondřejov data set and 100 new spectra from the LAMOST pool.

To track the performance of our CNN, we estimate the precision (the ratio of correctly predicted single-peak and double-peak spectra in all predicted target spectra) in each iteration. Therefore, we randomly selected 30 spectra classified into single-peak and double-peak (target spectra) classes from the LAMOST pool (hereafter the performance estimation sample). The size of 30 was chosen as a good trade-off between confidence and the demands of visual verification. Then, we manually annotated the performance estimation

type of layer	hyperparameters
input	140 neurons
convolutional	64 kernels
convolutional	64 kernels
max pooling	
convolutional	128 kernels
convolutional	128 kernels
max pooling	
convolutional	256 kernels
convolutional	256 kernels
max pooling	
fully connected	512 neurons
fully connected	512 neurons
fully connected	3 neurons

Table 3.1: The CNN for the active deep learning method consists of 6 convolutional (with 3 pixels wide kernels), 3 max pooling (with pool size 2, stride 2, and no padding), and 3 fully connected layers. Dropout (Srivastava et al. 2014) with probability $p = 0.5$ is applied to the first two fully connected layers as a regulariser.

sample and compared our annotations with the annotations predicted by our CNN. Thus, we estimated the precision after each iteration. The performance estimation sample of 30 spectra functions as a test set. In standard machine learning, a test set is a random sample of all unseen data that could be put into the CNN. In our case, all possible data for our CNN are in the so far unannotated LAMOST pool. Therefore, the performance estimation sample will provide an unbiased estimate of precision. We would like to point out that the annotations of the performance estimation sample is different from the manual annotations of batches for active learning. The annotations of the performance estimation sample are forgotten after the precision estimation, and the spectra are left in the LAMOST pool.

Finally, we stopped our experiment in the 17th iteration when the estimated precision reached more than the predefined threshold (in our case 80 %) for the third time. We chose the values of these parameters as a trade-off between time and performance requirements, and it can be chosen differently for different data sets. Figure 3.4 displays the precision of our CNN over 17 active learning iterations.

Because the training of our CNN was time-consuming, we sped up the method by training the CNN during the active learning phase for a smaller number of epochs. Then, after the active learning phase, we ran the Adam optimisation algorithm of the CNN for a longer time (the training was stopped when the loss function did not improve by 10^{-5} during 100 training iterations) to ensure that good convergence was achieved, and thus fewer false candidates will be produced. In the following text, we refer to this step as *long training*.

3.3.4 Results

Our method identified 4 379 candidate spectra with signatures of emission-line profiles including candidates found by the manual annotations in all the 4 136 482 LAMOST DR2 spectra. The last CNN predicted 3 574 spectra as single-peak and 587 as double-peak profiles, while we found 157 single-peak candidates and 61 double-peak candidates during manual annotations of batches. As explained earlier, it also includes absorption profiles with small visible disturbances that may be caused by additional circumstellar emissions. After visual inspection of the predicted candidates, we rejected 58 as bad (partly destroyed, noisy, or with pure absorption profiles) and computed the partial confusion matrix in Table 3.2. Finally, we had a set of

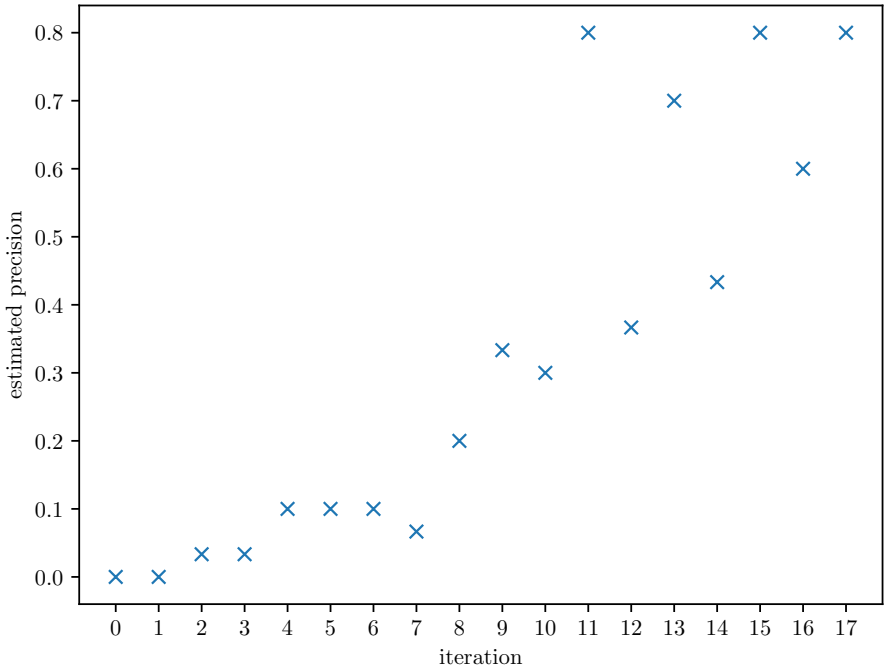

Figure 3.4: Estimated precision from a sample of 30 single-peak and double-peak spectra for each iteration (the zeroth iteration is estimated when the CNN is trained only with the initial Ondřejov data set)

predicted class	actual class	percentage	count
single-peak	single-peak	97.5 %	3 484
	double-peak	1.5 %	53
	uninteresting	1.0 %	37
double-peak	double-peak	93.4 %	548
	single-peak	3.1 %	18
	uninteresting	3.6 %	21

Table 3.2: Partial confusion matrix of the final classification of our experiment (excluding candidates found by manual annotation). The numbers show the percentage and counts of correctly predicted spectra of all spectra predicted for a given class. The 4 161 spectra in this table are all the candidates predicted as single or double-peaks after the long training. After we visually reviewed all of them, we found that 58 of candidates are uninteresting spectra (37 predicted as single-peaks and 21 predicted as double-peaks). The target classes also include some misclassification: 53 double-peaks are classified as single-peaks, and 18 single-peaks are classified as double-peaks. We could not compute the last row of the uninteresting class because it would mean manual annotation of all the four million spectra that are predicted as uninteresting.

4 321 spectra of about[2] 3 788 individual objects.

This set includes 2 644 spectra of 2 291 objects that have been found previously by Hou et al. (2016), and 664 new spectra of 549 objects that are listed in Set of Identifications, Measurements, and Bibliography for Astronomical Data (SIMBAD) which were not found by Hou et al. (2016). Our method proved to be reliable (with an error smaller than 6.5%) because most of the candidates are classified in SIMBAD as various cases of emission-line objects, such as cataclysmic variables, young stellar objects, dwarf novae, symbiotic binaries, infrared excess objects from InfraRed Astronomical Satellite (IRAS), classic Be stars and Herbig Ae/Be star (HAeBe) stars. In addition, our method found 1 013 spectra of 948 new objects that are neither known in SIMBAD nor discovered by Hou et al. (2016).

The newly discovered objects span almost all spectral classes as assigned by the LAMOST pipeline, but also many unclassified ones. The visual inspection has confirmed that all of them have signatures of emission in their line profiles. Some have even prominent strong emissions. These include three

[2]The exact number of individual objects is difficult to estimate because of cross-matching problems.

class	run	estimated precision	predicted spectra count
single-peak	no. 1	4.1 %	343 988
	no. 2	3.0 %	301 396
	no. 3	4.0 %	167 545
double-peak	no. 1	2.0 %	248 336
	no. 2	2.0 %	409 908
	no. 3	0.0 %	342 230

Table 3.3: The table shows results of three runs of passive learning, specifically the precision estimated from a random sample of 100 spectra from each target class, and the numbers of spectra classified into each target class.

supernovae candidates, an unknown Wolf-Rayet star, and many Be stars and young stellar objects. Moreover, through the visual preview of candidates, several normal and Seyfert galaxies and a high-velocity star (LAMOST HVS1) were also identified.

3.3.5 Comparison with passive learning

To clarify the real gain of active learning, we compare our active deep learning method to a passive learning dual scenario. The passive learning can be considered the zeroth iteration of our active deep learning method. However, the zeroth iteration in our application is carried out in the accelerated regime.

We carried out an independent experiment to prove the benefits of active learning. We trained our CNN using the setting of the long training with the initial training set of our active deep learning method (the Ondřejov data set), and we used the trained CNN to classify all the spectra in the LAMOST pool. Then, we estimated precision of the CNN from random samples of 100 spectra from target classes. In order to make a more reliable conclusion, we ran the experiment three times. The results are shown in Table 3.3.

The comparison of Table 3.2 and Table 3.3 shows that the three CNNs were unable to learn without the support of spectra from the LAMOST pool added to the training set by active learning. Therefore we conclude that the gain of our active deep learning method is significant.

Chapter 4

Method for Prediction of Spectroscopic Redshift[1]

The second goal (see Section 1.2) was to to develop methods that allow us to select samples with high predictive uncertainty. In Section 2.2, we identified Monte Carlo (MC) dropout and deep ensembles as such candidate methods. Here, we research the MC dropout on the task of prediction of spectroscopic redshift, while in Chapter 5, we researched deep ensembles on the task of prediction of atmospheric properties of exoplanets. Due to the advantages of deep learning and provision of predictive uncertainties, deep learning models obtained by these methods can be used for both the active deep learning and consistency check (see Section 1.1).

4.1 Astronomical motivation

Quasars or quasi-stellar objects (QSOs) are the most luminous objects in the universe, however, due to their enormous distances, they appear in the optical telescope like faint stars. Their spectra were not understood well as they were showing sets of unknown spectral lines. While analysing spectra of object 3C 273, Schmidt (1963) realised that the strange QSO lines were in fact Balmer emission lines shifted by a large offset to the much more redder wavelengths. Ratio of this wavelength offset to the original laboratory wavelength is called a (cosmological) redshift, as we believe that it is caused by the global expansion of the universe. QSOs with high redshifts representing very early stages of the cosmic history are important for studies of large-scale

[1]This chapter is based on **O. Podsztavek**, P. Škoda, and P. Tvrdík (2022). "Spectroscopic redshift determination with Bayesian convolutional networks". In: *Astronomy and Computing*. DOI: 10.1016/j.ascom.2022.100615.

structure of the early universe (Hennawi and Prochaska 2007) and namely the epoch of reionization (Becker et al. 2001). QSOs belong to the wider class of active galactic nuclei (AGN) where the energy is produced by the accretion onto the supermassive black hole and their different spectral line shapes are explained by the unified model introduced by Urry and Padovani (1995).

As shown above, the spectroscopic redshift of QSOs is a very important parameter for cosmological studies and therefore huge efforts have been undertaken to create catalogues of QSO redshifts measured in large surveys, such as Sloan Digital Sky Survey (SDSS) and Large Sky Area Multi-Object Fiber Spectroscopic Telescope (LAMOST). As there are millions of spectra available in such surveys, automatic pipelines are used for this task, usually based on a pattern matching (Glazebrook, Offer, and Deeley 1998) of a given spectrum with a pixel-by-pixel shifted set of templates of various QSO classes (Vanden Berk et al. 2001). This method is understood to be straightforward and redshifts from such pipelines are often taken as a ground truth for training of various machine learning models like Kügler, K. Polsterer, and Hoecker (2015) or Rastegarnia et al. (2022). Unfortunately, the similar structures of emission lines of QSOs are repeated in multiple spectral ranges, and, namely in case of noisy spectra, the best matching with template may be identified at a completely different wavelength region, yielding severe errors in redshift determination. As the amount of measurements is huge, the human may visually inspect only a small subsample of results. To increase the reliability of pipeline predictions, a Bayesian convolutional neural network (hereafter Bayesian CNN) may be used for performing the consistency check (see Section 1.1) of results and for identifying the incorrect values.

We illustrate the consistency check on the task of spectroscopic redshift determination in the SDSS with a Bayesian CNN. All objects in SDSS catalogues of QSOs up to data release (DR) 12 were visually inspected (Pâris et al. 2017). However, since there are more than half a million QSOs in the SDSS DR14 QSO catalogue (Pâris et al. 2018) and about three-quarters of a million QSOs in the SDSS DR16 QSO catalogue (Lyke et al. 2020), only a small subset of spectra is visually inspected, and thus astronomers have to rely on automated methods. A standard method for spectroscopic redshift determination is template fitting used by the SDSS pipeline (Bolton et al. 2012). The redshift of a spectrum is measured by comparing the spectrum with all predefined templates at almost all pixels using χ^2 minimisation. The SDSS pipeline associates each measurement with a statistical error `Z_ERR` and confidence flag `ZWARNING`. Methods based on principal component analysis (PCA), such as the `redvsblue` algorithm (Mas des Bourboux 2021), are also traditional. The `redvsblue` algorithm uses predefined templates of the SDSS

pipeline and measures redshifts by fine-tuning previous redshift determinations.

There were attempts to approach spectroscopic redshift determination by predicting redshifts with CNNs. QuasarNET (Busca and Balland 2018) is a CNN inspired by the You Only Look Once (YOLO) (Redmon and Farhadi 2017) system for object detection. The YOLO system is a CNN for classification and regression simultaneously. The CNN of YOLO can detect and categorise an object in an image and draw a bounding box around the object. Accordingly, QuasarNET is trained to localise and classify several spectral lines in spectra. However, QuasarNET is outperformed by our proposed Bayesian CNN. Stivaktakis et al. (2020) trained a CNN for classification to predict redshift on *simulated* Euclid spectra. They formulated the redshift prediction as a classification task by mapping real redshift values into bins. However, the redshift prediction is a regression task, so we use a CNN for regression. Moreover, we experiment with *real* SDSS spectra. D'Isanto and K. L. Polsterer (2018) researched a probabilistic redshift prediction on photometric data with mixture density networks (MDNs) (Bishop 1994). MDNs are neural network models (including CNNs) that produce a Gaussian mixture model (GMM) as their output. GMMs represent predictive probability density functions (PDFs) of photometric redshifts allowing us to get predictive uncertainties. In this work, our approach to get predictive uncertainties is different from this method because we use a Bayesian CNN. Bayesian CNNs do not represent predictive distributions explicitly in the form of predictive PDFs as MDNs. But we can still sample from the predictive distributions and thus get predictive uncertainties.

4.2 SZNet: CNN for redshift prediction

The CNN for the redshift prediction is a modification of the VGG Net-A CNN (Simonyan and Zisserman 2015). Hereafter we denote this CNN as SZNet, where the letters "SZ" stand for spectroscopic redshift because astronomers denote redshift z. The reason we chose VGG Net-A is because it belongs to the family of VGG CNNs that achieved state-of-the-art results on the object localisation (i.e. a kind of a regression) task and object classification task of the the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) 2014 (Russakovsky et al. 2015).

Since VGG CNNs are designed to process images with 3 RGB channels but spectra have only 1 flux value for each pixel, SZNet is a reduced modification of VGG Net-A by a factor of $8 = 2^3$. For example, we reduced $2^{12} = 4096$ neurons or kernels to $2^{12-3} = 2^9 = 512$ neurons or kernels. Equally like VGG

type of layer	hyperparameters
convolutional	8 kernels
max pooling	
convolutional	16 kernels
max pooling	
convolutional	32 kernels
convolutional	32 kernels
max pooling	
convolutional	64 kernels
convolutional	64 kernels
max pooling	
convolutional	64 kernels
convolutional	64 kernels
max pooling	
fully connected	512 neurons
fully connected	512 neurons
fully connected	1 neuron

Table 4.1: SZNet consists of 11 layers with model parameters (8 convolutional and 3 fully connected layers) and 5 max pooling layers.

CNNs, SZNet is based on the principle of stacking convolutional layers with small 3 pixels wide kernels. The stride hyperparameter of convolutional layers is 1 pixel, and the padding hyperparameter of convolutional layers is set so that the input and output sizes are the same. It contains 5 max pooling layers with 2 pixels wide filters that are moved by (i.e. their stride is) 2 pixels each time. All convolutional and fully connected layers (excluding the last layer) apply the ReLU non-linear activation function (see Subsection 2.1.1. SZNet has 1 neuron in its last layer that produces a scalar value as the redshift prediction. It is trained with the mean squared error (MSE) loss $\mathcal{L}_{\text{MSE}}(\theta)$ defined by equation (2.1). Table 4.1 summarises the architecture of SZNet.

To get Bayesian SZNet, SZNet has to use the L_2 regularisation during training, and dropout has to be applied to each layer with model parameters θ (see Subsection 2.2.1). However, we applied dropout only to the first two fully connected layers of SZNet as in the VGG Net-A. One can think of this approach as first extracting a proper representation with the convolutional and max pooling part and then feeding such a representation into a Bayesian fully connected neural network (FCNN). Gal et al. (2017) used the same

method to diagnose cancer with a VGG-like Bayesian CNN.

Following the guidelines by Goodfellow et al. (2016), we trained Bayesian SZNet with the Adam optimiser in its default setting, i.e. learning rate $\eta = 0.001$, $\beta_1 = 0.9$, and $\beta_2 = 0.999$ (Kingma and Ba 2015). We stopped training with the early stopping algorithm, i.e. if there is no improvement in the mean continuous ranked probability score (CRPS) (see Section 2.3) during the last 32 epochs. The number of 32 epochs is a trade-off between the convergence speed and training time. At the end of the training of Bayesian SZNet, we restored the best-found model parameters θ^*. The original batch size (i.e. a hyperparameter of the training algorithm) in the training of VGG CNNs was 256 images, so we kept the batch size of 256 spectra.

4.3 Experiments

4.3.1 SDSS QSO data

The SDSS DR12 QSO catalogue[2] (hereafter DR12Q) (Pâris et al. 2017) is the final catalogue of QSOs from the Baryon Oscillation Spectroscopic Survey (BOSS) of SDSS-III. The catalogue is the result of a visual inspection of 546 856 QSO candidates, including stars and galaxies. The QSO candidates are stored in the DR12Q superset (also denoted as the DR12Q parent sample) together with *redshifts from visual inspection* and their confidences (columns denoted as `Z_VI` and `Z_CONF_PERSON` respectively). The DR12Q superset is suitable for machine learning because it provides redshifts from visual inspection for almost all its spectra. We used as target values only the redshifts from visual inspection `Z_VI` > -1 (`Z_VI` $= -1$ stands for redshifts that do not have visual inspection available) with confidences `Z_CONF_PERSON` $= 3$ (`Z_CONF_PERSON` $\in \{1, 2\}$ stands for uncertain redshifts). Furthermore, we excluded 65 corrupted spectra[3] that have all flux values equal to zero in their Flexible Image Transport System (FITS) files. We ended up with 523 331 spectra.

To evaluate the generalisation capability of Bayesian SZNet, we evaluated the model not only on a separate test set from the DR12Q superset but also on the superset of the SDSS DR16 QSO catalogue.[4] The SDSS DR16 QSO catalogue (hereafter DR16Q) (Lyke et al. 2020) is the final SDSS-IV QSO catalogue of extended BOSS. Its superset (hereafter the DR16Q superset)

[2] `https://sdss.org/dr12/algorithms/boss-dr12-quasar-catalog/`

[3] We list the corrupted spectra in the `dr12q_superset.err` file on GitHub, at `https://github.com/podondra/bayesian-redshift`.

[4] `https://sdss.org/dr16/algorithms/qso_catalog/`

contains 1 440 615 spectra. Due to its size, approximately only one third of its spectra have redshifts from visual inspection. As FITS files of 42 spectra are missing,[5] we evaluated the generalisation capability of Bayesian SZNet on 1 440 573 spectra from the DR16Q superset.[6]

We trained Bayesian SZNet on spectra from the DR12Q superset. Then, we applied it to spectra from the DR16Q superset to illustrate its generalisation capability.

Data preparation consisted of pseudo-continuum normalisation, spectral range cutting, and zero padding. We got each spectrum from individual FITS files (named "optical spectra per-object lite files") that are available on the Science Archive Server (SAS) of SDSS.[7] Figure 4.1 illustrates data preparation on an example of an SDSS spectrum.

To do the pseudo-continuum normalisation, we applied the *density of the least squares* (DLS) method[8] in its simplest version (Bukvić et al. 2008). Firstly, we standardised flux values to ensure numerical stability of the DLS method:

$$\boldsymbol{x}_i' = \frac{\boldsymbol{x}_i - \mu(\boldsymbol{x}_i)}{\sigma(\boldsymbol{x}_i)},$$

where the function μ returns the mean of an array, function σ returns the standard deviation of an array, and $\boldsymbol{x}_i'$ is the standardised spectrum. With the DLS method, we fit the third-order polynomial to flux values of each spectrum using ordinary least squares (i.e. inverse variances of each flux value were not used). We found the parameters of the DLS method through experimentation. We set its exponent parameter $k = 2$ that additionally simplifies the DLS method, and its removal parameter $r = 0.9$ that assures both fast processing and good fit. We subtracted[9] the continuum from the standardised spectrum. This pseudo-continuum normalisation makes spectra invariant to scale, intensities, and continuum shape, i.e. we focus mainly on spectral lines. Moreover, it is more convenient to pad with zero spectra with subtracted continuum.

[5]Missing spectra from the DR16Q superset are listed in the `dr16q_superset.err` file on GitHub, at `https://github.com/podondra/bayesian-redshift`.

[6]The DR16Q superset spectra can be identified using the catalogue available on Zenodo, at `https://doi.org/10.5281/zenodo.5173824`.

[7]Visit `https://sdss.org/dr12/data_access/bulk/` for SDSS DR12 download instruction and `https://sdss.org/dr16/data_access/bulk/` for SDSS DR16 download instruction.

[8]The Julia implementation of the DLS method is available on GitHub, at `https://github.com/podondra/DLSMethod.jl`.

[9]The spectrum is not divided by the continuum as in methods of rectification commonly employed in stellar astronomy.

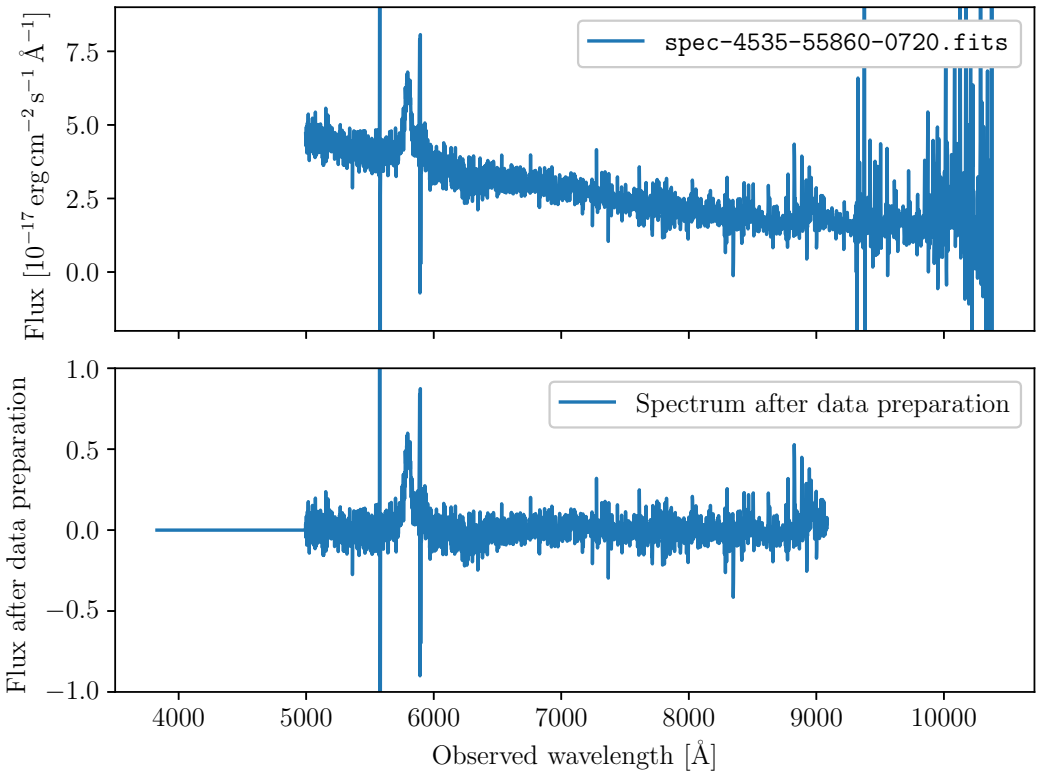

Figure 4.1: The top plot shows an original SDSS spectrum. The bottom plot displays the same spectrum after pseudo-continuum normalisation, cut into the predefined wavelength coverage, and zero padding (visible on the left side).

After the pseudo-continuum normalisation, we cut each spectrum into the logarithmic wavelength coverage 3.5832–3.9583 log Å (3830.01–9084.48 Å).[10] We calculated it from the DR16Q superset wavelength coverage because it has smaller wavelength coverage, and we want to show the generalisation capability of the examined model. We selected the minimal wavelength to be the 99.9 quantile of all minimal wavelengths (i.e. 3830.01 Å) and the maximal wavelength to be the 0.01 quantile of all maximal wavelengths (i.e. 9084.48 Å) in the DR16Q superset. The wavelength coverage 3.5832–3.9583 log Å covers 3752 flux values. We tried to maintain a wide wavelength coverage, but still some parts of spectra were cut off. Therefore, models might not have available those parts of spectra that led to a decision in the visual inspection.

Spectra that do not cover the full range of wavelengths are padded with zero so that we can make predictions for all spectra. This padding introduces a straight structure (see Figure 4.1) similar to straight structures of missing flux values in SDSS spectra (see Figure 4.2). Therefore, zero padding also indicates missing flux values.

Finally, we split the DR12Q superset spectra into training, validation, and test sets.[11] The DR12Q superset does not contain more spectra of a single object, so a spectrum of an object used for training or validation cannot be in the test set. With inspiration from the split sizes in ILSVRC (Russakovsky et al. 2015), the sizes of the validation and test sets are 50 000 spectra each. The remaining 423 331 spectra are left in the training set. Data preparation resulted in matrices in which rows contain spectra and columns flux values for corresponding wavelengths and target vectors with redshifts from visual inspection.

4.3.2 Evaluation metrics and tools

The most common metric for an evaluation of redshift prediction methods is the *root-mean-squared error* (RMSE):

$$\mathrm{RMSE}(\{(y_i, \hat{y}_i)\}_{i=1}^{N}) = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2},$$

[10] We choose wavelengths with logarithmic spacing because original SDSS spectra are in logarithmic wavelengths.

[11] Lists of the DR12Q superset training, validation, and test spectra are in the `dr12q_superset_train.lst`, `dr12q_superset_valid.lst`, and `dr12q_superset_test.lst` files respectively on GitHub, at `https://github.com/podondra/bayesian-redshift`.

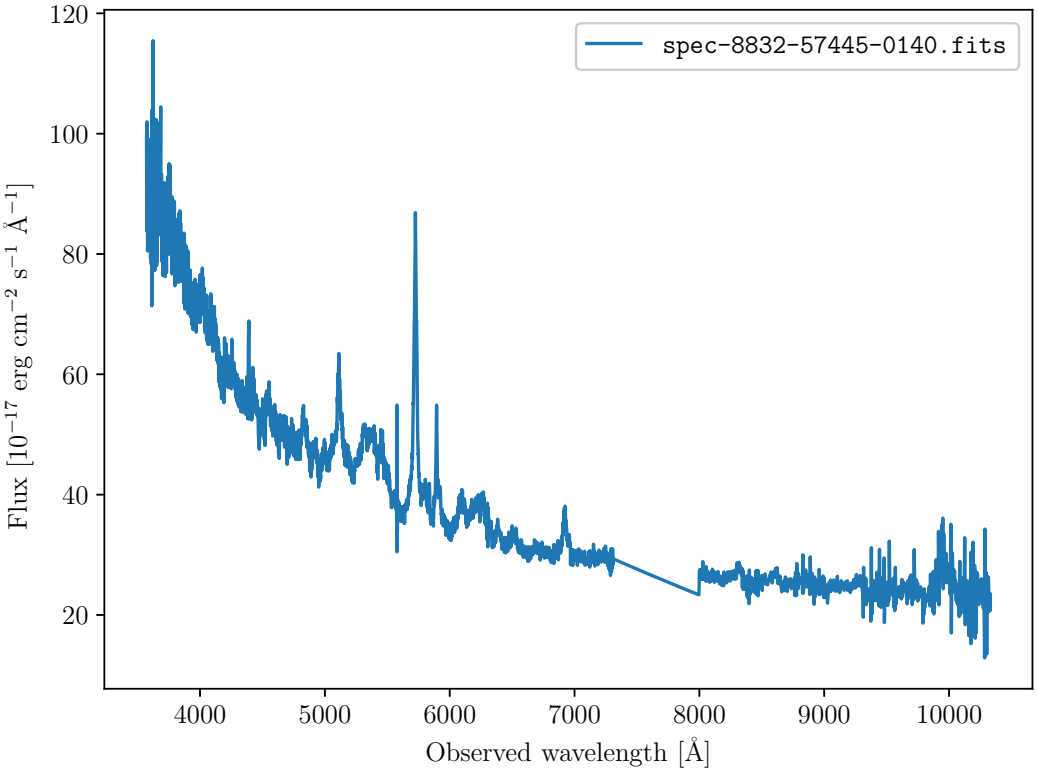

Figure 4.2: Example of SDSS spectrum with missing flux values

where y_i is a redshift from visual inspection (i.e. a target value) of spectrum $\boldsymbol{x}_i$, $\hat{y}_i$ is a redshift determination (i.e. prediction or measurement) from spectrum $\boldsymbol{x}_i$, and N is the size of the data set used for the metric computation. The redshift determination is a regression task, so the RMSE is a natural metric because the RMSE is the square root of the MSE loss $\mathcal{L}_{\mathrm{MSE}}(\theta)$ without the L_2 regularisation $\lambda\|\theta\|^2$ defined by equation (2.1).

However, the RMSE evaluates only point estimates (e.g. predictive means or deterministic predictions) and not predictive distributions themselves. With Bayesian CNNs, we can sample from predictive distributions, so we have to use proper scoring rules and tools to evaluate them (see Section 2.3).

One such proper scoring rule, introduced to probabilistic redshift predictions by D'Isanto and K. L. Polsterer (2018), is CRPS introduced in Section 2.3. With Bayesian CNNs, we cannot get true predictive CDFs, but we can approximate them with empirical predictive CDFs (Hersbach 2000). The CRPS values for individual spectra are usually aggregated and the mean value is reported.[12] Moreover, if predictions $\hat{y}_i$ are deterministic, then the mean CRPS equals the *mean absolute error* (MAE):

$$\mathrm{MAE}(\{(y_i, \hat{y}_i)\}_{i=1}^N) = \frac{1}{N}\sum_{i=1}^{N} |\hat{y}_i - y_i|.$$

Therefore, we can compare probabilistic and deterministic methods with the mean CRPS.

Furthermore, we employ probability integral transform (PIT) histograms (see Section 2.3) to evaluate the calibration of predictive distributions. We again approximate true predictive CDFs with empirical predictive CDFs.

Different but related evaluation metric is *coverage*. Coverage is the ratio of the number of spectra for which we accept the prediction of a Bayesian CNN. Predictive uncertainties from a Bayesian CNN allow us to reject predictions with predictive uncertainties greater than a chosen threshold.

4.3.3 Evaluation on the DR12Q superset

Firstly, we had to determine values of the weight decay λ, dropout probability p, and number of samples T of Bayesian SZNet. We set $T = 256$ because it is large enough to produce consistent results. Then, we used a grid search (a search over a finite set of values) to determine optimal values of the weight decay λ and dropout probability p. Figure 4.3 presents the result

[12] We used the `properscoring` package available on GitHub, at `https://github.com/TheClimateCorporation/properscoring`, to compute CRPS.

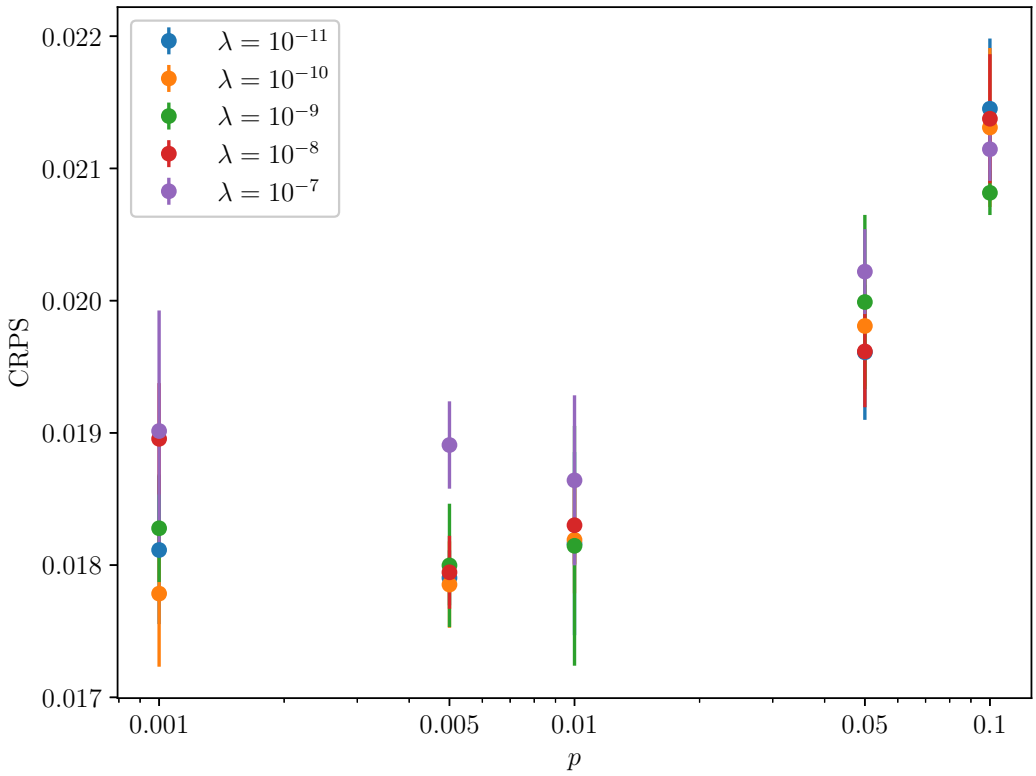

Figure 4.3: Grid serch result of Bayesian SZNet for weight decay λ and dropout probability p with 95% confidence intervals

of the grid search over weight decays $\lambda \in \{10^{-11}, 10^{-10}, 10^{-9}, 10^{-8}, 10^{-7}\}$ and dropout probabilities $p \in \{0.1, 0.05, 0.01, 0.005, 0.001\}$. With each hyperparameter setting, we trained five Bayesian SZNets so that we could provide 95% confidence intervals. We determined optimal values according to the mean CRPS evaluated on the DR12Q superset validation set so that we prefer calibrated and sharp predictive distributions. The optimal value of weight decay appeared to be $\lambda = 10^{-9}$ because smaller values do not provide any improvement while higher values are worse. We can extrapolate this to other values of weight decay λ. However, results in terms of the mean CRPS are inconclusive with respect to the optimal dropout probability p. Therefore, we inspected PIT histograms in Figure 4.4. They indicate that Bayesian SZNet with dropout probability $p = 0.01$ provides the most calibrated predictive distributions. Therefore, we selected the best Bayesian SZNet out of the five with dropout probability $p = 0.01$ trained with weight decay $\lambda = 10^{-9}$ for further evaluation.

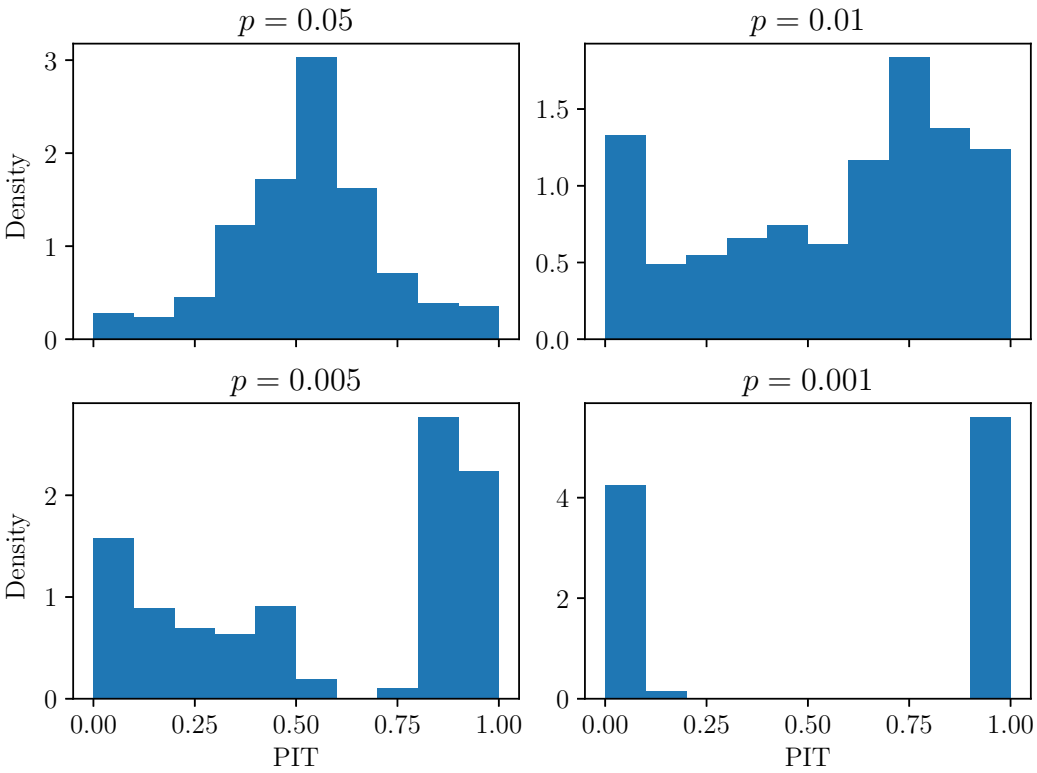

Figure 4.4: PIT histograms of Bayesian SZNet for different dropout probabilities p with weight decay $\lambda = 10^{-9}$

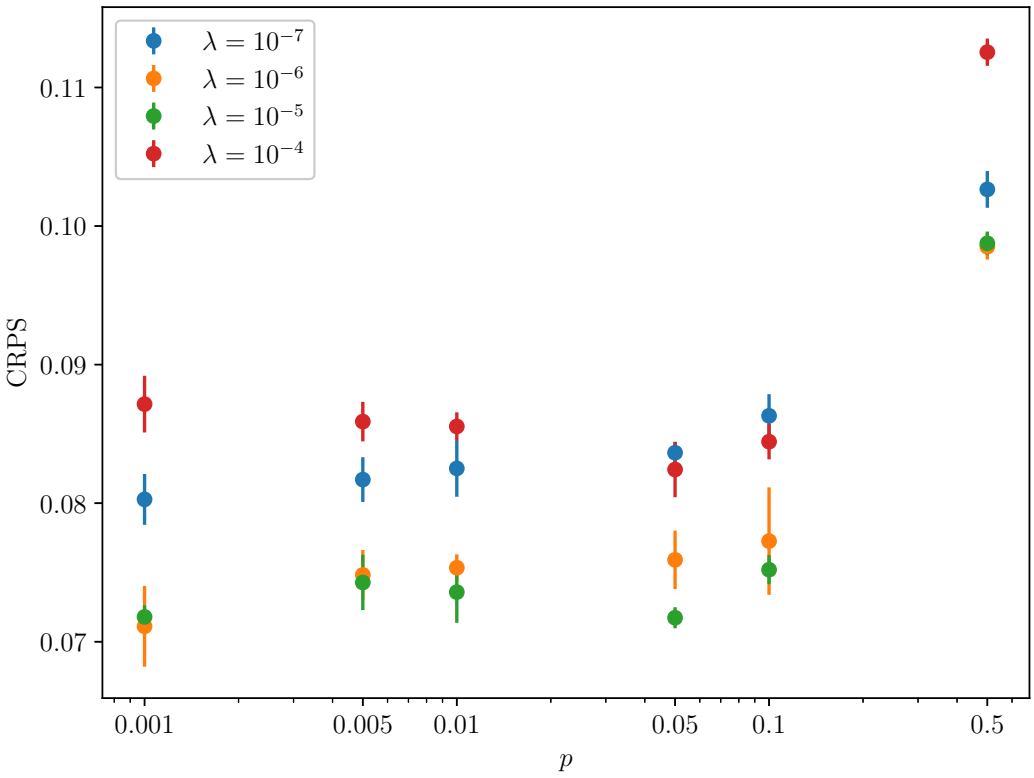

Figure 4.5: Grid search result of Bayesian FCNN for weight decay λ and dropout probability p with 95% confidence intervals

To have a simple machine learning baseline for reference, we separated the fully connected part from Bayesian SZNet and used it as a Bayesian FCNN. It was trained in the same way as Bayesian SZNet (see the description in Section 4.2). We again used grid search to determine optimal weight decay $\lambda = 10^{-5}$ and optimal dropout probability $p = 0.05$ (see Figures 4.5 and 4.6).

Next, we evaluated Bayesian SZNet on the DR12Q superset test set. The DR12Q superset test set was neither used to train Bayesian SZNet nor to optimise its hyperparameters. To put the result of Bayesian SZNet into a relevant context, we compare it with the Bayesian FCNN and 4 other baselines. The first baseline is the SDSS pipeline that processed the DR12Q superset (the `Z_PIPE` column in the DR12Q superset). The DR16Q superset provides other 3 baselines. The second baseline is the SDSS pipeline that processed the DR16Q superset (its measurements are in the `Z_PIPE` column in the DR16Q superset). The third baseline is QuasarNET (its predictions are in the `Z_QN` column in the DR16Q superset). The fourth baseline is

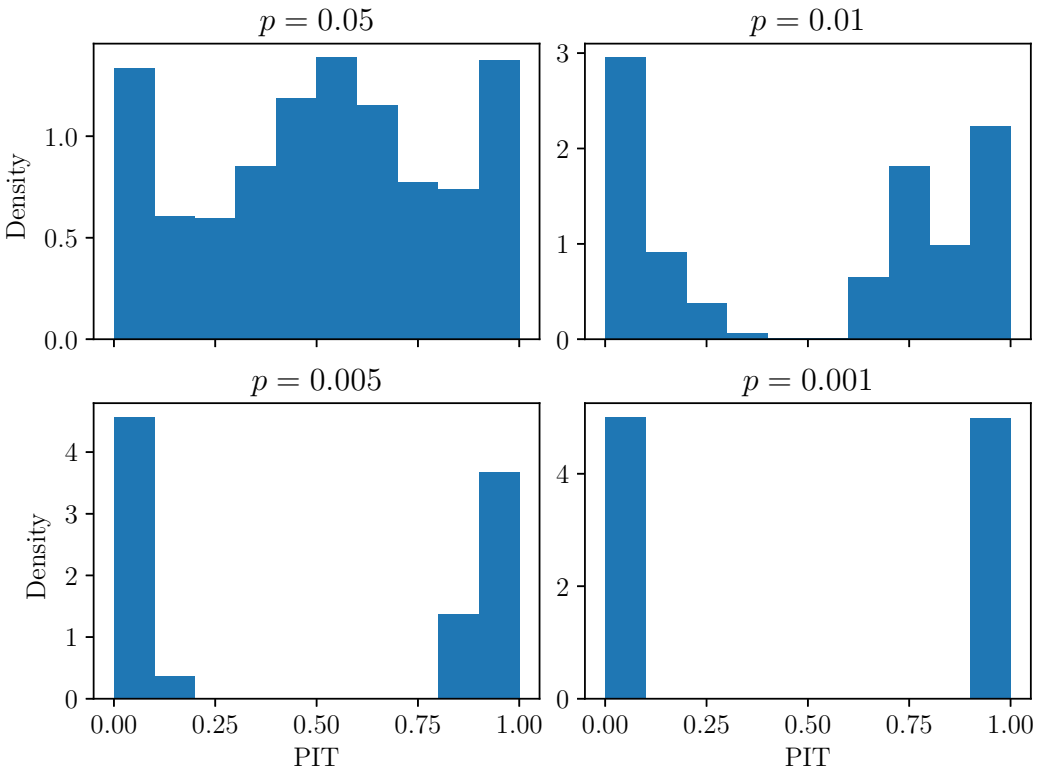

Figure 4.6: PIT histograms of Bayesian FCNN for different dropout probabilities p with weight decay $\lambda = 10^{-5}$

model	RMSE	mean CRPS
Bayesian SZNet	**0.1083**	**0.0171**
Bayesian FCNN	0.2106	0.0712
`Z_PCA` (DR16Q superset)	0.4118	0.0724
`Z_PIPE` (DR16Q superset)	0.4518	0.0830
`Z_PIPE` (DR12Q superset)	0.5002	0.0969
`Z_QN` (DR16Q superset)	1.1530	0.6812

Table 4.2: Evaluation on the DR12Q superset test set (the values in rows with "DR16Q superset" in parentheses are computed from redshift determinations cross-matched from the DR16Q superset)

the `redvsblue` algorithm (its measurements are in the `Z_PCA` column in the DR16Q superset). We cross-matched the `Z_PIPE`, `Z_QN`, and `Z_PCA` columns from the DR16Q superset to the DR12Q superset according to `PLATE`, `MJD` and `FIBERID` identifiers that uniquely identify each SDSS spectrum. Because of cross-matching, we can compare Bayesian SZNet with these three DR16Q baselines. However, we could not cross-match 113 spectra, so we used only a test set of 49 887 (out of 50 000) spectra.

Table 4.2 shows the evaluation on the DR12Q superset test set and compares performance metrics of Bayesian SZNet and other baselines. The two rows that evaluate measurements of pipelines differ because the SDSS pipeline has changed between SDSS DR12 and DR16 (Lyke et al. 2020). Bayesian SZNet is the best in both the RMSE and mean CRPS by a significant margin.

4.3.4 Generalisation to the DR16Q superset

Evaluation of generalisation capability of Bayesian SZNet to the DR16Q superset is not as straightforward as the evaluation on the DR12Q superset. These two supersets have different distributions of observations, so we cannot extrapolate the results from the DR12Q superset test set to the DR16Q superset. Histograms in Figure 4.7 illustrate that the redshift distributions differ between the two supersets. Not all spectra have the redshift from visual inspection in the DR16Q superset. Nonetheless, the DR16Q superset contains a *primary redshift* (hereafter primary `Z`) that is the redshift from visual inspection, if available, else it is the redshift of the SDSS pipeline. The primary `Z` is in the column denoted `Z`, while the column denoted `SOURCE_Z` indicates its source. The DR16Q superset contains more spectra with primary $\texttt{Z} \in [0, 2.15)$. Moreover, Bayesian SZNet was trained with the DR12Q super-

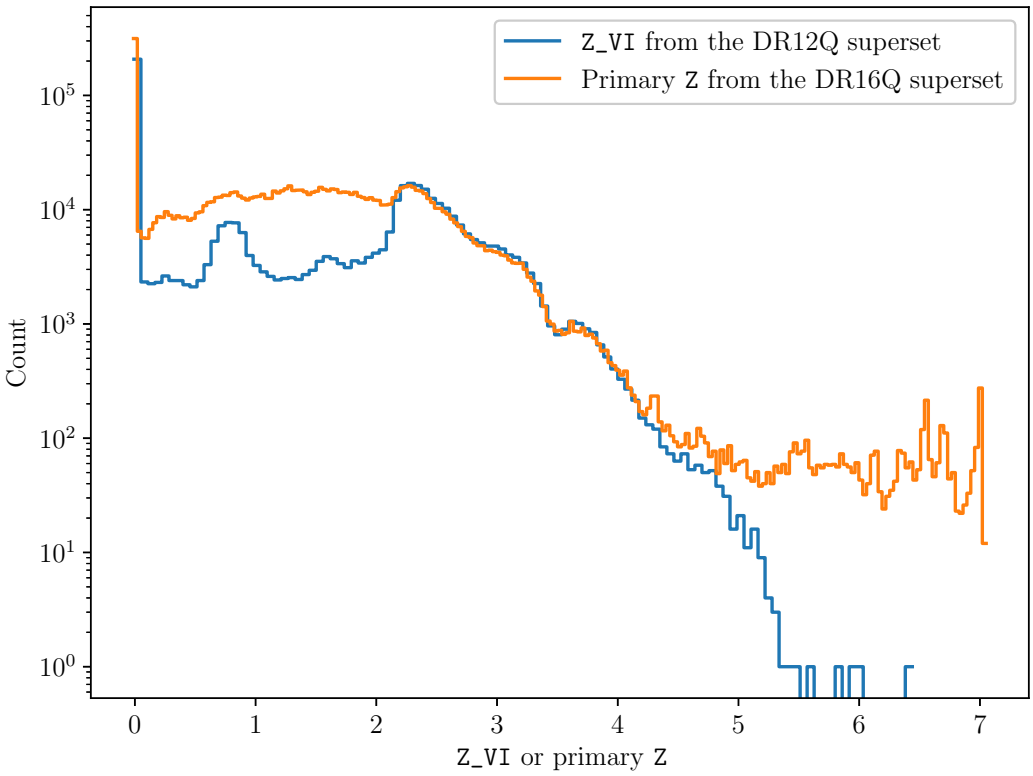

Figure 4.7: Redshift histograms of the DR12Q and DR16Q supersets prove the distribution discrepancy.

set training set, where the minimal and maximal redshifts are -0.008 and 5.216 respectively. Redshifts might be negative real numbers because of blue-shifted sources, e.g. Andromeda (Marel and Guhathakurta 2008). However, many spectra have primary Z > 5.216 in the DR16Q superset. Therefore, we cannot state that the RMSE of Bayesian SZNet is 0.2106 in the DR16Q superset since the DR12Q superset test set is not a random subsample of the DR16Q superset.

We want Bayesian SZNet to generalise to spectra with correct redshifts greater than 5.216. On the other hand, we expect Bayesian SZNet to provide high predictive uncertainties for spectra with a high redshift. There are 3645 spectra with primary Z > 5.216 in the DR16Q superset. Only 32 of them have the redshift from visual inspection, while the SDSS pipeline measured the rest. According to Lyke et al. (2020), spectra with primary Z > 5 and SOURCE_Z $=$ PIPE should be considered suspect. Figure 4.8 displays a sample spectrum with a high redshift measured by the SDSS pipeline (primary Z

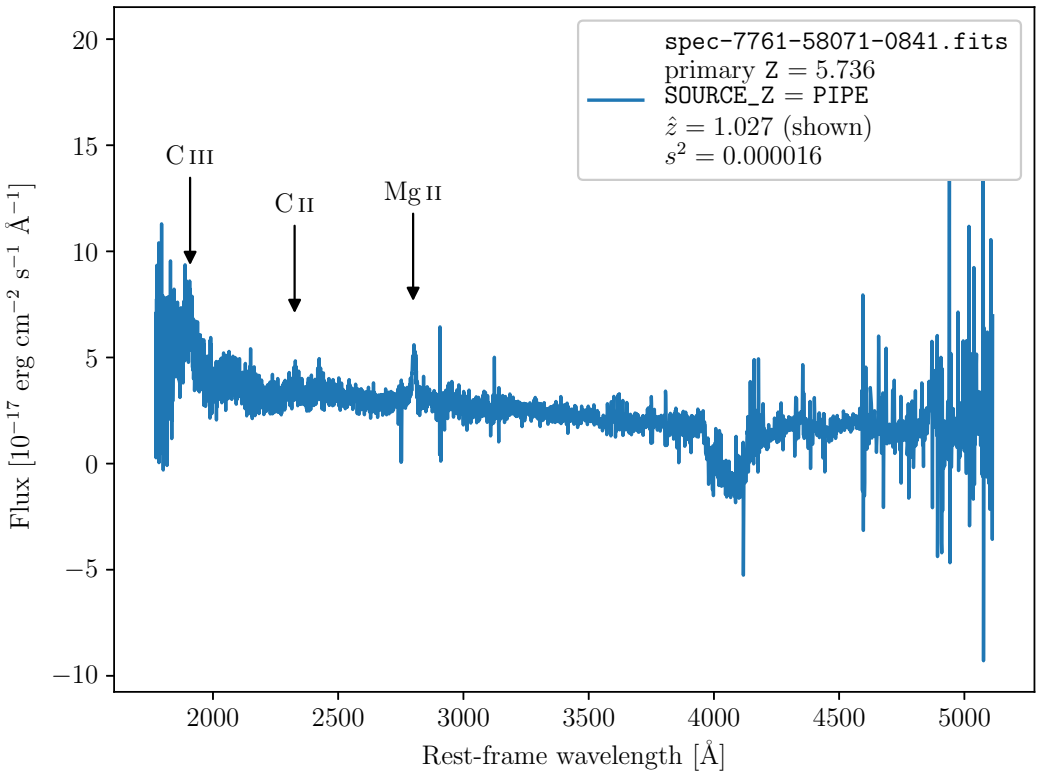

Figure 4.8: Example of a spectrum with a high primary Z = 5.230 measured by the SDSS pipeline while Bayesian SZNet predicted $\hat{y} = 0.38$.

= 5.736). Bayesian SZNet is almost sure ($s^2 = 0.000016$ is a relatively low predictive variance confronted with the predictive variance distribution in Figure 4.9) that the correct redshift of the spectrum is $\hat{y} = 1.027$. We visually inspected the spectrum, and we confirmed that the prediction of Bayesian SZNet is correct.

The only way to numerically estimate the performance of Bayesian SZNet on the DR16Q superset is to use redshifts from the random visual inspection of 10 000 spectra by Lyke et al. (2020). These redshifts are stored in the column denoted Z_10K in the DR16Q superset. Z_10K redshifts with their corresponding spectra constitute a subsample (hereafter the Z_10K subsample) that was originally used to evaluate measurements of the SDSS pipeline. We indeed found 10 000 spectra using the PIPE_CORR_10K column in the DR16Q superset. However, 304 spectra from the Z_10K subsample have Z_10K redshifts either −999 or −1 (the value of −999 stands for potential blazars and −1 is a missing value). We tried to annotate such spectra manually, but

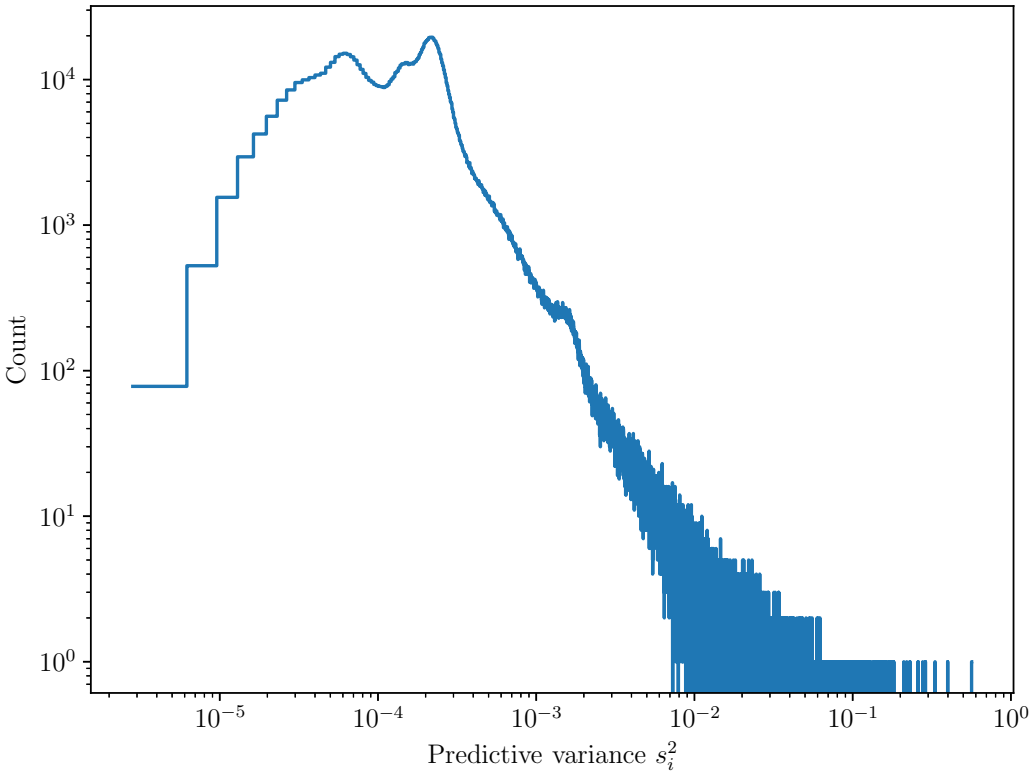

Figure 4.9: Histogram of predictive variances of Bayesian SZNet

they were challenging to annotate. Therefore, we decided to exclude them from the Z_10K subsample. Furthermore, we cross-matched spectra from the Z_10K subsample and DR12Q superset training and validation sets using TOPCAT (Taylor 2005) and the radius of 0.5″. This ensures that objects in the Z_10K subsample were used to neither train nor optimise hyperparameters of Bayesian SZNet, and thus are unseen by it. We found 1 028 matching spectra that we also had to exclude from the Z_10K subsample, so the Z_10K subsample contains 8 668 spectra.

We base the performance estimation on these 8 668 spectra, i.e. we use the Z_10K redshifts as target values. Table 4.3 summarises the results of Bayesian SZNet and other baselines. Bayesian SZNet is the best concerning the RMSE. The SDSS pipeline and `redvsblue` algorithm beat Bayesian SZNet in terms of the mean CRPS. However, this is not a fair comparison because there is a bias in Z_10K redshifts towards redshift measurements of the SDSS pipeline. Namely, Z_10K redshifts were set to be the same as redshift measurements of the SDSS pipeline (i.e. the Z_PIPE column) if the absolute value of velocity difference $|\Delta v_i|$ of these two was less than or equal to $3\,000\,\mathrm{km\,s^{-1}}$ (Lyke et al. 2020). Velocity difference Δv_i is defined as:

$$\Delta v_i = c \cdot \frac{\hat{y}_i - y_i}{1 + y_i},$$

where c is the speed of light, $\hat{y}_i$ is the redshift measurement of the SDSS pipeline, and y_i is the correct redshift. Otherwise, Z_10K redshifts are determined by visual inspection. Therefore, the performance of the SDSS pipeline is overestimated. The same applies to Z_PCA because the `redvsblue` algorithm fine-tunes redshift measurements of the SDSS pipeline. This also explains why Bayesian SZNet is better in terms of the RMSE. The RMSE is more sensitive to outliers than the mean CRPS. The mean CRPS is the MAE for deterministic predictions (see Subsection 4.3.2). This corresponds precisely to the way Z_10K redshifts were determined because it suppresses minor errors (i.e. $|\Delta v_i| \leq 3\,000\,\mathrm{km\,s^{-1}}$), but it does not suppress outliers. Additionally, we provide scatter plots comparing redshift determinations to Z_10K redshifts in Appendix A.1. The scatter plots reveal that measurements of the SDSS pipeline and `redvsblue` algorithm have more outliers than predictions of Bayesian SZNet. Finally, the PIT histogram for the Z_10K subsample in Figure 4.10 shows that predictive distributions of Bayesian SZNet are slightly biased and underdispersed.

model	RMSE	mean CRPS
Bayesian SZNet	**0.1894**	0.0387
`Z_PIPE`	0.2289	0.0260
`Z_PCA`	0.2114	**0.0245**
`Z_QN`	0.5406	0.1584

Table 4.3: Generalisation evaluation on `Z_10K` redshifts of 8 668 spectra from the random visual inspection of the DR16Q superset

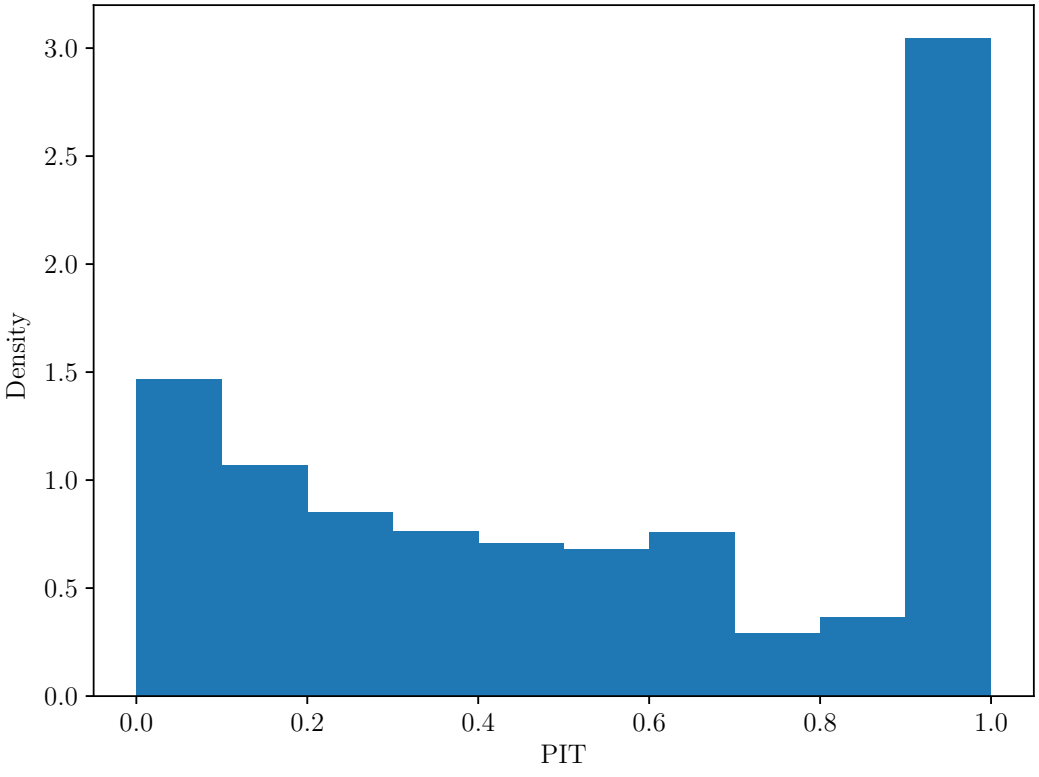

Figure 4.10: PIT histogram of Bayesian SZNet for the `Z_10K` subsample

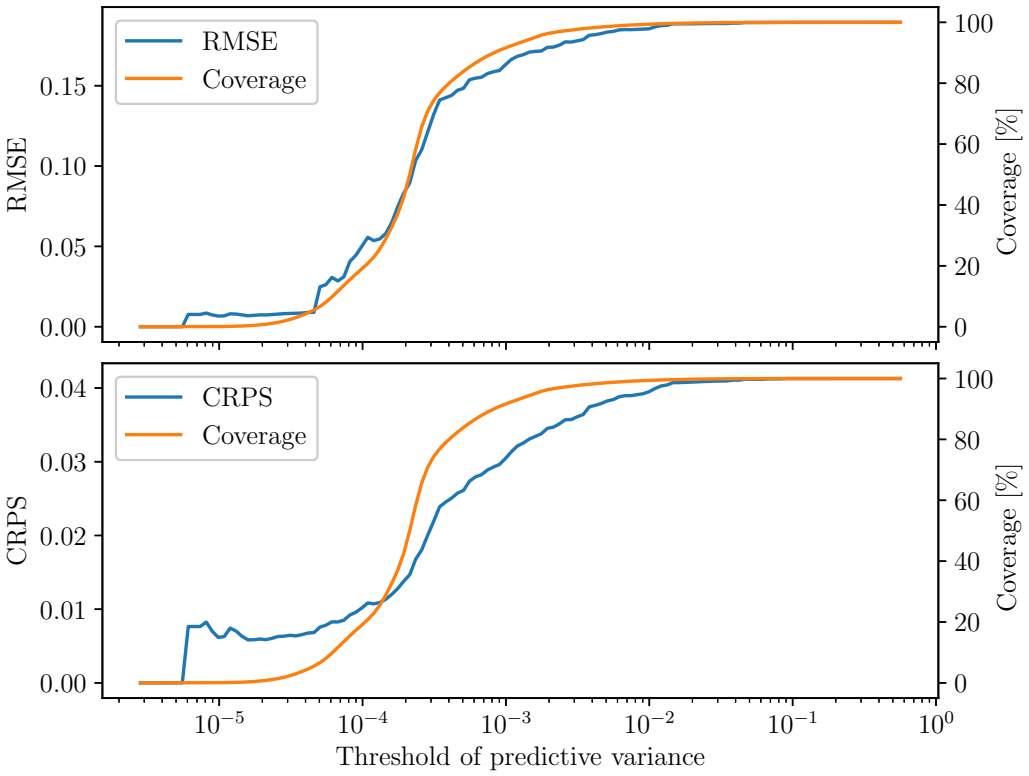

Figure 4.11: Impact of a predefined threshold of predictive variance on the RMSE and mean CRPS in comparison to coverage

4.3.5 Utilisation of predictive uncertainties

Predictive variances associated with predictions are one of the main advantages of Bayesian SZNet for the redshift prediction. We can use the predictive variances to do thresholding. Firstly, we choose a threshold of the predictive variance, and we refuse predictions with predictive variances greater than the threshold. Therefore, we refuse uncertain predictions that are probably incorrect. This will improve the performance of Bayesian SZNet, but the coverage (see Subsection 4.3.2) will be lower. Figure 4.11 depicts dependencies of the RMSE, mean CRPS, and coverage on a predefined threshold. Table 4.4 expresses the same dependencies as Figure 4.11 for three levels of coverage: 99, 95, and 90 %. These coverages mean to refuse predictions for 14 406, 72 029, and 144 058 spectra from the DR16Q superset respectively.

coverage	RMSE	mean CRPS
99 %	0.1848	0.0389
95 %	0.1714	0.0334
90 %	0.1587	0.0293

Table 4.4: Comparison of the RMSE and mean CRPS for three levels of coverage

4.3.6 Suitability of Bayesian SZNet for consistency check

We verified the performance of Bayesian SZNet, so we can advance to an illustration of its suitability for the consistency check. For the illustration, we selected two spectra with signal-to-noise ratios greater than 12 shown in Figures 4.12 and 4.13. The threshold 12 of the signal-to-noise ratio is substantiated by random visual inspection that revealed that spectral features are difficult to identify in spectra with lower signal-to-noise ratios. We converted observed wavelengths to rest-frame wavelengths using the correct redshift verified by our visual inspection (marked in the legend by the word "shown"). The legend of each spectrum displays its primary `Z` and its source, the `IS_QSO_FINAL` flag that indicates QSOs that are in the DR16Q, its redshift $\hat{y}$ predicted by Bayesian SZNet with its associated predictive variance s^2. The spectrum in Figure 4.12 is a missed QSO (`IS_QSO_FINAL` $= 0$) with incorrect primary `Z` $= 0.236$. However, Bayesian SZNet predicts its redshift correctly $\hat{y} = 2.085$ and is certain, i.e. it provides the small predictive variance $s^2 = 0.000133$. On the contrary, Figure 4.13 displays a spectrum of a star that was incorrectly identified as a QSO (`IS_QSO_FINAL` $= 1$), probably because of emission lines. Bayesian SZNet predicted its redshift $\hat{y} = 0.008$, which is slightly off the correct value but with a higher predictive variance $s^2 = 0.001516$.

The whole DR16Q superset can be examined using the catalogue presented in Appendix A.2. Further examples are shown in Appendix A.3.

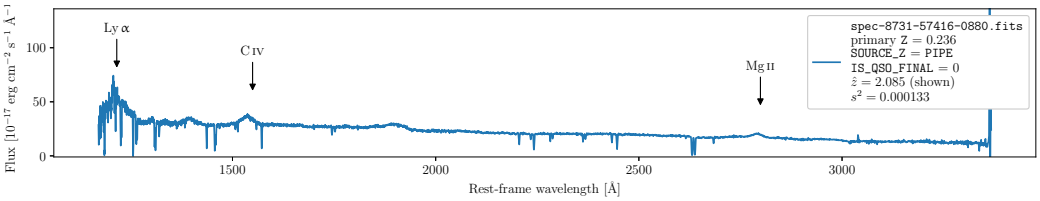

Figure 4.12: Spectrum of a QSO missed by the DR16Q (`IS_QSO_FINAL` = 0) because of the SDSS pipeline incorrect measurement

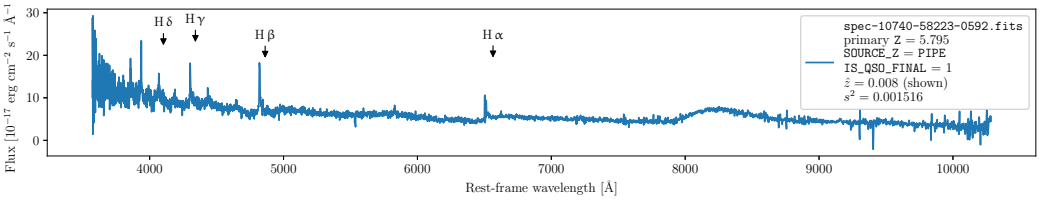

Figure 4.13: Spectrum of a star incorrectly included in the DR16Q (`IS_QSO_FINAL` = 1)

Chapter 5

Method for Prediction of Atmospheric Properties of Exoplanets[1]

As stated at the beginning of Chapter 4, we now move our attention to deep ensembles (see Subsection 2.2.2). We research deep ensembles on the task of predicting atmospheric properties of exoplanets. This research concerns the second goal (see Section 1.2) on methods that allow us to select samples with high predictive uncertainty.

5.1 Astronomical motivation

Exoplanets are planets that orbit stars other than our own Sun. The number of confirmed exoplanets is growing exponentially thanks to dedicated ground- and space-based missions, such as Wide Angle Search for Planets (WASP) (Pollacco et al. 2006), Kepler (Borucki et al. 2010), or Transiting Exoplanet Survey Satellite (TESS) (Ricker et al. 2015). The next task is characterising these exoplanets, i.e. understanding their atmospheric composition, dynamics and interior. This helps us understand how exoplanets evolve, how likely it is to find an Earth-like planet, and the conditions for life to emerge. Answering these questions is crucial to understanding our place in the universe.

Predicting atmospheric properties of exoplanets from astronomical spectra is a computationally demanding task. Astronomers have traditionally relied on statistical sampling methods such as nested sampling (Skilling 2006)

[1]This chapter is based on K. H. Yip et al. (2022b). "Lessons Learned from Ariel Data Challenge 2022 – Inferring Physical Properties of Exoplanets From Next-Generation Telescopes". In: *Proceedings of the NeurIPS 2022 Competitions Track*.

to approximate the distributions of different atmospheric properties, such as the temperature of the planet or trace gas abundances (e.g. Madhusudhan 2018). However, these methods, while precise, are not easily scalable to large data sets, and there have only been a few population-level analyses on the different classes of exoplanets (e.g. Sing et al. 2016; Barstow et al. 2017; Tsiaras et al. 2018; Fisher and Heng 2018; Pinhas et al. 2019; Mansfield et al. 2021; Roudier et al. 2021; Changeat et al. 2022; Edwards et al. 2022). Atmospheric Remote-Sensing Infrared Exoplanet Large-Survey (Ariel) launch in 2029 promises to provide thousands of high-quality spectra for a wide range of exoplanets (Tinetti et al. 2021). Conventional sampling methods will soon become a significant bottleneck to understanding planetary characteristics in our local galactic neighbourhood (Yip et al. 2022a; Ardevol Martinez et al. 2022; Matchev et al. 2022). We need scalable methods to analyse thousands of planets efficiently. The emergence of machine learning makes it possible to analyse thousands or even millions of planets at scale within a reasonable amount of time.

5.2 Ariel Data Challenge

Our method was developed to solve the Ariel Data Challenge 2022. The Ariel Data Challenge is an annual challenge that seeks innovative solutions to tackle pressing issues faced by the Ariel and exoplanet community. Each year, the challenge focuses on a different issue involving the technical or scientific aspects of the mission. A summary of the first challenge and its top-ranked solutions can be found in Nikolaou et al. (2023). Ariel Data Challenge 2022 focused on innovative solutions to the problem of probabilistic prediction of atmospheric properties of exoplanets.

5.2.1 Task

Specifically, the goal of the competition was to develop a method capable of predicting 6 atmospheric properties of exoplanets given simulated spectra from Ariel with corresponding auxiliary data comprising features of host stars (their distance, mass, radius, and temperature) and exoplanets themselves (their mass, orbital period, distance, radius, and surface gravity). The atmospheric properties to be predicted are relative molecular abundances of five gases (namely H_2O, CH_4, CO_2, CO, and NH_3), and the mean atmospheric temperature at the terminator of a planet (i.e. the line that separetes day and night also known as the twilight zone). The exact target values vary depending on the specific participation track chosen. The light track

asked participants to submit their predictions for the 16th, 50th and 84th percentile of each of the 6 properties. The regular track asked participants to submit weighted samples from a predictive distribution of those properties.

5.2.2 Data

Each spectrum is generated following a 3-step approach. First, a planet configuration is randomly selected from the catalogue of discovered planets. Based on the configuration of the chosen planet, a randomly generated atmospheric profile and trace gasses are produced. Second, the atmospheric modelling software `TauREx` (Al-Refaie et al. 2021) produces a theoretical atmospheric model of the exoplanet. Third, this model is processed by `ArielRad` (Mugnai et al. 2020) to generate a realistic spectrum expected by Ariel. The whole process is automatic via the software `Alfnoor` (Changeat et al. 2020; Mugnai et al. 2021). More than 100 000 simulated Ariel spectra were generated for this competition.

Target distributions of atmospheric properties were generated for around 26 % (21 988) of the simulated spectra using the Bayesian nested sampling method `MultiNest` (Feroz and Hobson 2008; Feroz et al. 2019). Thus, only these samples are annotated, while the rest is unannotated. More details are available in Changeat et al. (2022).

5.2.3 Scores

Submissions to the light track were evaluated based on squared relative errors. Squared relative errors were calculated between the 16th, 50th, and 84th percentiles of target distributions and their predictions. The final *light score* was calculated using the formula:

$$1000 - 10\sqrt{\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{6}\left[\left(\frac{\hat{a}_i^{(j)} - a_i^{(j)}}{a_i^{(j)}}\right)^2 + \left(\frac{\hat{b}_i^{(j)} - b_i^{(j)}}{b_i^{(j)}}\right)^2 + \left(\frac{\hat{c}_i^{(j)} - c_i^{(j)}}{c_i^{(j)}}\right)^2\right]},$$

where N is the size of the test set, i indexes exoplanets, j enumerates atmospheric properties, and $a_i^{(j)}$, $b_i^{(j)}$, $c_i^{(j)}$ are the 16th, 50th, 84th percentiles of their target distributions.

Submissions to the regular track were evaluated using the Earth mover's distance, also called the 2-Wasserstein distance. The distance was calculated between weighted samples from a predictive distribution and those generated by the nested sampling from the corresponding target distribution. The final

regular score was calculated using the formula:

$$1000 \left(1 - \frac{1}{N} \sum_{i=1}^{N} \mathrm{EMD}_i \right),$$

where EMD_i denotes the Earth mover's distance between those two weighted sets of samples.

5.3 Method

Our method is based on *deep ensembles* (see Subsection 2.2.2). The deep ensemble consists of $M = 20$ convolutional neural networks (CNNs). Therefore, its output is a probability density function (PDF), a mixture of 20 equally weighted normal distributions.[2]

The inputs of the CNNs are the spectra with corresponding auxiliary data comprising all individual features. Standardisation is applied to both spectra and auxiliary data. Each spectrum is standardised so that it has zero mean and unit variance. This standardisation reduces differences in the ranges of values of individual spectra. Therefore, CNNs can focus on the shapes of spectra since the relative molecular abundance in atmospheres determine them. Auxiliary data are standardised feature-wise, i.e. each auxiliary feature is subtracted by the mean and divided by standard deviation of the training set. The training set includes all 21 988 annotated exoplanets (i.e. spectra, auxiliary data, and annotations). The original annotations are weighted samples from target distributions. Such annotations would make the training of CNNs difficult. Therefore, the annotations are simplified to be 6 normal distributions fitted to the weighted samples independently for each atmospheric property.

Each CNN is a modification of the VGG Net-A CNN (Simonyan and Zisserman 2015). It consists of a convolutional part (6 convolutional and 4 max pooling layers) and fully connected part (7 fully connected layers, each with 1024 neurons). The convolutional part processed spectra; its output is concatenated with auxiliary data into a vector processed by the fully connected part. The activation function of all layers (except the last one) is the rectified linear unit (ReLU) introduced in Subsection 2.1.1. The last layer outputs 6 normal distributions, i.e. 6 means and 6 variances. The softplus function ($\mathrm{softplus}(a) = \log(1 + e^a)$) outputs these 6 variances, and a minimal variance of 10^{-6} is added for numerical stability. All CNNs are trained

[2]The code underlying this work is available online on GitHub, at `https://github.com/podondra/ariel-data-challenge`.

rank	team name	light score
1	**podondra**	987.44
2	gators	987.26
3	user1	986.01
4	MonsieurSolver	985.56
5	Stefan_Stefanov	985.31
6	LeoPulga	984.36
7	jhawkins515	983.33
8	asweet	982.56
9	ls	980.87
10	yl	979.15

Table 5.1: Final light scores of top-10 ranking solutions

with Kullback–Leibler divergence as the loss function using Adam optimiser (Kingma and Ba 2015) with a learning rate of 10^{-4} and batch size of 256. These and other hyperparameters are optimised on a separate validation set (20 % of the training set) using early stopping on the light score. However, after optimising them, data from both training and validation sets are used to train the final CNNs for 2048 epochs. This number of epochs ensures sufficient convergence of CNNs.

The deep ensemble of 20 CNNs generates samples from the predicted distributions: 250 samples are sampled from the 6 normal distributions outputted by each CNN. Therefore, there are 5000 samples in total for the regular track. Then, the sample percentiles are computed for the light track.

5.4 Results

Our method described above won first place in the light track and third place in the regular track. The final top-10 ranking solutions are listed in Table 5.1 for the light track and Table 5.2 for the regular track. Our solution is listed under the team name "podondra". These placings prove that it is a high-performing method for prediction of atmospheric properties of exoplanets.

rank	team name	regular score
1	gators	987.80
2	Stefan_Stefanov	987.26
3	**podondra**	987.25
4	LeoPulga	986.91
5	user1	984.26
6	asweet	984.01
7	MonsieurSolver	972.71
7	Weimin	967.35
9	yl	963.43
10	Ginqwerty	939.25

Table 5.2: Final regular scores of top-10 ranking solutions

Chapter 6

Method for Automatic Miscalibration Diagnosis[1]

Finally, the third goal (see Section 1.2) was to develop a method that can help us identify problems with the reliability of predictive uncertainties. We have an active deep learning method (see Chapter 3) and methods to associate predictions with uncertainties (see Chapter 4 and Chapter 5). However, how do we know that those predictive uncertainties (that we want to use to select samples for annotation in active deep learning) are reliable? Assessing the reliability of those predictive uncertainties is an essential task that we address next. Here, by reliable, we mean probabilistically calibrated (see Section 2.3). Moreover, we still focus on regression tasks as in Chapter 4 and Chapter 5. As concluded in Section 2.3, one should be able to diagnose miscalibration by visually inspecting a probability integral transform (PIT) histogram. However, understanding the *cause* of miscalibration from a PIT histogram requires a lot of experience. Therefore, we present method to an automatic interpretation of PIT histograms based on an interpreter trained with a synthetic data set.

6.1 Method

To facilitate an interpretation of a PIT histogram, we propose to perform a decomposition into a data-generating and a predictive distribution. These distributions allow us to reconstruct a PIT histogram that is close to the original PIT histogram. We achieve this decomposition using a machine

[1]This chapter is based on **O. Podsztavek** et al. (2024). "Automatic Miscalibration Diagnosis: Interpreting Probability Integral Transform (PIT) Histograms". In: *ESANN 2024 proceedings*.

learning model called an *interpreter*. Because the PIT is translation- and scale-invariant, an interpreter trained on a *synthetic data set of PIT histograms* can interpret a given PIT histogram independently of the original translation and scale of data-generating and predictive distribution pairs. Given the PIT histogram of a predictive model and data set, its interpretation allows us to diagnose miscalibration of the model by comparing the estimated data-generating and predictive distribution.

6.1.1 Synthetic data set of PIT histograms

A synthetic data set has to be relevant to the particular application, i.e. relevant to expected data-generating and predictive distributions. The synthetic data set consists of P PIT histograms with B bins, each generated from N pairs of data-generating and predictive distributions.

We generate the j-th PIT histogram, where $j \in \{1, \ldots, P\}$, by first generating a set of PIT values, and then assigning these PIT values to the predefined bins. Technically, that means choosing a pair of predictive and data-generating CDFs ($F_i^{(j)}$ and $G_i^{(j)}$) for each $i \in \{1, \ldots, N\}$, sampling a target value $y_i^{(j)}$ from $G_i^{(j)}$, and computing $F_i^{(j)}(y_i^{(j)})$. Then, we assign the PIT values into B bins, and calculate the corresponding relative frequencies, such that the area under the histogram integrates to 1 and is therefore independent of N.

6.1.2 Interpreter

The input of the interpreter is a PIT histogram, and its output estimates the data-generating distribution that led to the PIT histogram. In particular, because a mixture of normal distributions can approximate any data-generating distribution if it has enough components, the interpreter is a *mixture density network* (MDN) (Bishop 1994). To allow data-generating distributions of the synthetic data set to be from any family of distributions, the interpreter is trained with a Monte Carlo approximation to 1-Wasserstein distance between true $G_i^{(j)}$ and predicted $\hat{G}^{(j)}$ data-generating CDFs:

$$\frac{1}{P}\sum_{j=1}^{P}\frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{K}|G_i^{(j)}(a_k) - \hat{G}^{(j)}(a_k)|,$$

here $a_1 < \ldots < a_K$ are equally spaced real numbers, a_1 and a_K are chosen according to the domain of the data-generating CDF $G_i^{(j)}$, and K is large enough to get a sufficiently accurate approximation.

6.2 Experiments

In probabilistic modelling, unimodal predictive distributions are often used to model multimodal data-generating distributions (e.g. Lakshminarayanan et al. 2017; Gal and Ghahramani 2016b). Therefore, we choose to experiment with a simple synthetic data set based on the normal family. For the j-th PIT histogram, every target value $y_i^{(j)}$ is a random number from a data-generation distribution $G_i^{(j)}$. For simplicity, we assume that $G_i^{(j)}$ is the same for all i. Specifically, $G^{(j)}$ is a mixture of two normal distributions, i.e. $y_i^{(j)}$ takes a random value from $\mathcal{N}(-d^{(j)}/2, t^{(j)})$ with probability $w^{(j)}$ or $\mathcal{N}(d^{(j)}/2, v^{(j)})$ with probability $1 - w^{(j)}$. By manipulating the parameters separation $d^{(j)}$, weight $w^{(j)}$, and variances $t^{(j)}$ and $v^{(j)}$, we can obtain PIT histograms of predictive models that are calibrated, under- and overestimated, under- and overdispersed, or have an incorrect number of modes. For simplicity, we fix the predictive distribution $F_i^{(j)}$ to $\mathcal{N}(0, 1)$ for all i and j. During our experiments, we observed that reconstructed PIT histograms match the original PIT histograms. This is already possible with the current choices of the fixed predictive distribution and the family of data-generating distributions. We will experiment with further distributions from various families with even more modes in the future.

In order to have a wide range of visually distinct PIT histograms in the synthetic data set of the interpreter, we decided to 1. define separation $d^{(j)} = 2(1 - a^{(j)}a^{(j)})$, where $a^{(j)}$ is sampled from the continuous uniform distribution $U(0.1, 1)$, 2. define variances $t^{(j)} = 2^{b^{(j)}}$ and $v^{(j)} = 2^{c^{(j)}}$, where $b^{(j)}$ and $c^{(j)}$ are sampled from $U(-2, 2)$, and 3. sample weight $w^{(j)}$ from $U(0, 1)$. Each generated PIT histogram has $B = 20$ bins containing a total of $N = 10^4$ PIT values per histogram.

Our experimental interpreter has a single hidden layer with 16 neurons and outputs a mixture of five normal distributions, which gives the interpreter enough flexibility with respect to our experimental synthetic data set.[2]

6.2.1 Evaluation on a simple synthetic inverse problem

First, we present a simple synthetic inverse problem for which a bimodal predictive distribution is adequate. The corresponding data set consists of 10^4 pairs of an input and target value (x_i, y_i), where $x_i = u_i'^2$, u_i' is sampled from $U(-1, 1)$, $y_i = u_i' + 0.25\epsilon_i$, and ϵ_i is sampled from $\mathcal{N}(0, 1)$. We train on it a *density network* (DN) (Nix and Weigend 1994) as a simple model with a unimodal normal predictive distribution.

[2]For more details, see `https://github.com/podondra/calibration`.

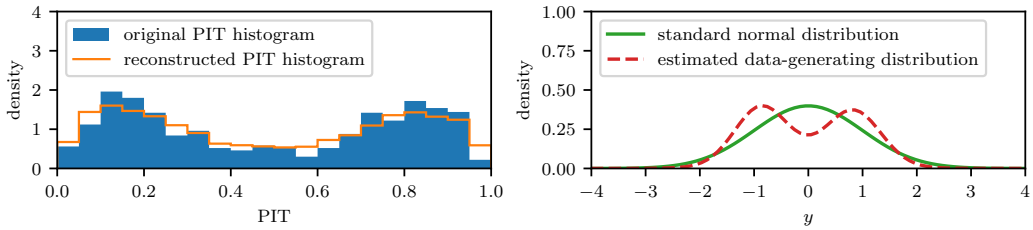

Figure 6.1: The PIT histogram (left) of a density network (DN) trained to solve the simple synthetic inverse problem and its interpretation (right).

Figure 6.1 displays the PIT histogram of the DN and the interpretation of the PIT histogram. The non-uniform PIT histogram reveals that the DN is miscalibrated. The interpretation clearly shows that the cause of miscalibration is that a unimodal predictive distribution is used to model a bimodal data-generating distribution.

6.2.2 Evaluation on real-world data sets

We choose the Year Prediction MSD, Physicochemical Properties of Protein Tertiary Structure, and Combined Cycle Power Plant (hereafter *year*, *protein*, and *power* respectively) data sets from University of California, Irvine (UCI) Machine Learning Repository, because they are commonly used for the evaluation of predictive uncertainties (e.g. Lakshminarayanan et al. 2017; Gal and Ghahramani 2016b).

Figure 6.2 displays PIT histograms of DNs trained on the data sets and their interpretations. In the case of the *year* data set, the PIT histogram of the DN is not uniform, indicating miscalibration, and its cause is more easily identified with the proposed decomposition. Our interpreter suggests that the normal predictive distribution is insufficiently flexible in its shape to model the data-generating distribution, and that it would be better to use a right-skewed predictive distribution. On the *protein* data set, the decomposition is similar to the one of the *year* data set. However, on the *power* data set, we observe that the PIT histogram of the DN exhibits some noise but is uniform. It is plausible that the data-generating distribution deviates only slightly from a normal distribution.

Table 6.1 reiterates the well-known fact that dealing with causes of miscalibration leads to tangible improvements in the predictive performance. We deal with the skewness by training MDNs that output mixtures of five normal distributions for simplicity. In real applications, an appropriate simple predictive distribution inferred from the interpretation should be used,

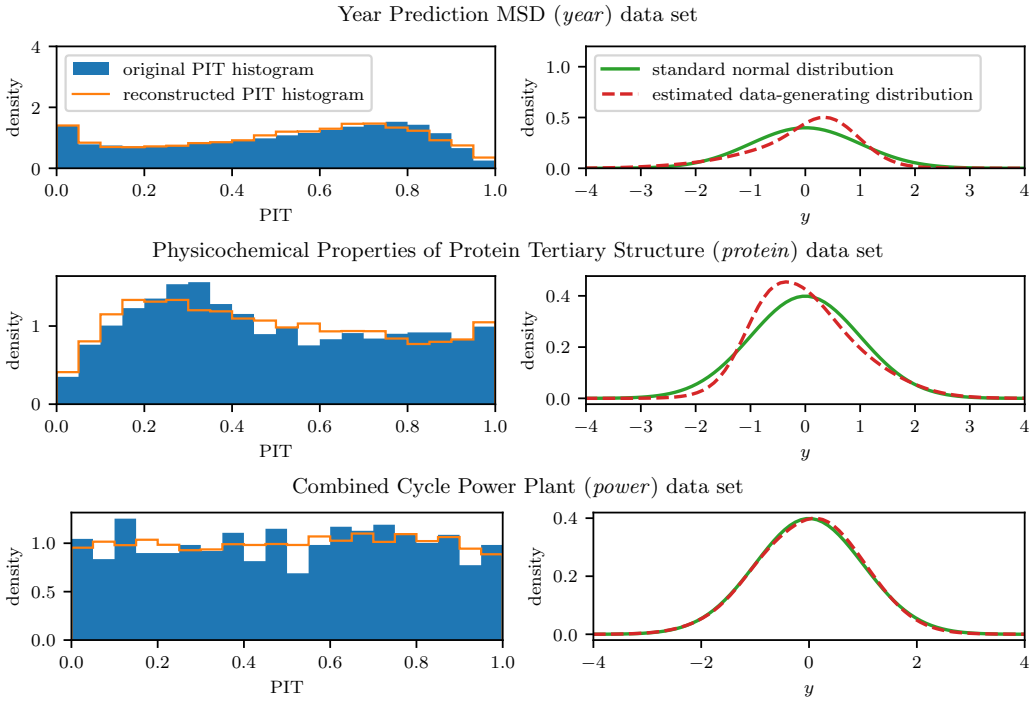

Figure 6.2: PIT histograms (left) of DNs trained on the data sets from UCI Machine Learning Repository and interpretations of those PIT histograms (right).

data set	model	mean NLL	mean CRPS
year	DN	3.373 ± 0.003	4.322 ± 0.013
	MDN	**3.094 ± 0.002**	**4.040 ± 0.007**
protein	DN	2.805 ± 0.039	2.342 ± 0.025
	MDN	**2.086 ± 0.017**	**1.940 ± 0.019**
power	DN	2.795 ± 0.018	2.175 ± 0.030
	MDN	**2.673 ± 0.023**	**2.093 ± 0.042**

Table 6.1: Comparison of models in terms of the mean NLL and mean CRPS

not a complex mixture of many distributions. We report the mean NLL and mean CRPS as performance metrics, accompanied by standard errors that are estimated from splitting the data sets into five train-test folds. The gap in predictive performance between DNs and MDNs is large for the *year* and *protein* data sets. This gap is mainly due to miscalibration when assuming a symmetric predictive distribution. For the *power* data set, the gap is small because both models are almost calibrated.

Chapter 7

Conclusion

This dissertation aimed toward better active deep learning for the annotation of large data sets with a particular focus on large data sets of astronomical spectra. Its first goal was to verify that active deep learning is a suitable set of methods for large data sets. Its second goal was to develop methods that allow us to select samples with high predictive uncertainty. Predictive uncertainties are essential for active deep learning to select a correct batch of samples for annotation by humans. Its third goal was to develop a method to help us identify problems with the reliability of predictive uncertainties.

Concerning the first goal, in Chapter 3, we have contributed a promising active deep learning method for the discovery of objects of interest in large data sets of astronomical spectra. This method, supported by interactive manual annotation of a small batch of predicted objects of interest, is very efficient and has led to the discovery of many new unknown stars with special physical properties. To the best of our knowledge, this was the first application of an active deep learning method to astronomical spectral classification. The main advantage of the method is that the objects of interest with characteristic features can be identified in cases where classical deep learning methods fail because a sufficiently large training set is not available. Our experiments identified many candidates that deserve more detailed examination because they may be rare astronomical objects with interesting physical properties.

Concerning the second goal, in Chapter 4, we developed a method based on Monte Carlo (MC) dropout that also quantifies predictive uncertainties on the spectroscopic redshift prediction task. Experiments confirmed that it can be well applied to the spectroscopic redshift prediction. It beats other baselines and generalises well. We can improve its performance further by thresholding, i.e. ignoring uncertain predictions. We also illustrated the consistency check and found several unrecognised or incorrectly identified quasi-stellar

objects (QSOs). A limitation of this work is that we only estimate predictive means and predictive variances of predictive distributions. A more sophisticated way would be to fit Gaussian mixture models (GMMs) to samples from predictive distributions. This would better account for multimodalities, but we leave this for future research.

Moreover, in Chapter 5, we designed a method based on a deep ensemble that predicts the atmospheric properties of exoplanets and also quantifies the uncertainties of these predictions. The method was one of the winning solutions of the Ariel Data Challenge 2022 competition, proving its performance. This method overcomes the limitation of the method based on MC dropout as it outputs GMMs that better account for multimodalities.

Concerning the third goal, in Chapter 6, we proposed a method that yields plots that essentially contain the same information as other tools for assessing the reliability of predictive uncertainties but in a form that makes problems with their reliability more obvious. By dealing with those problems, we get more reliable models. In turn, the overall performance of these models is superior in terms of other scalar scores.

All these contributions lead to better active deep learning. What remains is to integrate these methods and apply their integration to some large data sets of astronomical spectra. We leave this for the future work of some astronomers interested in discovering objects of interest, etc.

We took astronomy as an exemplary domain with a lot of data. The active deep learning method is developed for discovery in large data sets of astronomical spectra. Also, both methods for probabilistic prediction of spectroscopic redshift and atmospheric properties of exoplanets are tailored to astronomical spectra. However, the principles behind these methods can be generalised to any other domain with large data sets. On the other hand, the method for automatic miscalibration diagnoses is general to probabilistic models that produce predictive probabilistic distributions.

Bibliography

Abadi, M. et al. (2016). "TensorFlow: A System for Large-Scale Machine Learning". In: *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*.

Alger, M. J. et al. (2018). "Radio Galaxy Zoo: Machine learning for radio source host galaxy cross-identification". In: *Monthly Notices of the Royal Astronomical Society*. DOI: 10.1093/mnras/sty1308.

Alhassan, W. et al. (2018). "The FIRST Classifier: Compact and extended radio galaxy classification using deep Convolutional Neural Networks". In: *Monthly Notices of the Royal Astronomical Society*. DOI: 10.1093/mnras/sty2038.

Aniyan, A. K. and K. Thorat (2017). "Classifying Radio Galaxies with the Convolutional Neural Network". In: *The Astrophysical Journal Supplement Series*. DOI: 10.3847/1538-4365/aa7333.

Ardevol Martinez, F. et al. (2022). "Convolutional neural networks as an alternative to Bayesian retrievals for interpreting exoplanet transmission spectra". In: *Astronomy & Astrophysics*. DOI: 10.1051/0004-6361/202142976.

Barstow, J. K. et al. (2017). "A Consistent Retrieval Analysis of 10 Hot Jupiters Observed in Transmission". In: *The Astrophysical Journal*. DOI: 10.3847/1538-4357/834/1/50.

Becker, R. H. et al. (2001). "Evidence for Reionization at $z \sim 6$: Detection of a Gunn-Peterson Trough in a $z = 6.28$ Quasar". In: *The Astronomical Journal*. DOI: 10.1086/324231.

Bishop, C. M. (1994). *Mixture Density Networks*. Tech. rep. Aston University.

Bishop, C. M. (2006). *Pattern recognition and machine learning*. Springer.

Blanton, M. R. et al. (2017). "Sloan Digital Sky Survey IV: Mapping the Milky Way, Nearby Galaxies, and the Distant Universe". In: *The Astronomical Journal*. DOI: 10.3847/1538-3881/aa7567.

Bolton, A. S. et al. (2012). "Spectral classification and redshift measurement for the SDSS-III Baryon Oscillation Spectroscopic Survey". In: *The Astronomical Journal*. DOI: 10.1088/0004-6256/144/5/144.

Borucki, W. J. et al. (2010). "Kepler Planet-Detection Mission: Introduction and First Results". In: *Science*. DOI: 10.1126/science.1185402.

Bukvić, S. et al. (2008). "Advanced fit technique for astrophysical spectra". In: *Astronomy & Astrophysics*. DOI: 10.1051/0004-6361:20065969.

Busca, N. and C. Balland (2018). *QuasarNET: Human-level spectral classification and redshifting with Deep Neural Networks*. URL: https://arxiv.org/abs/1808.09955.

Calleja, J. de la et al. (2011). "Machine learning from imbalanced data sets for astronomical object classification". In: *2011 International Conference of Soft Computing and Pattern Recognition*. DOI: 10.1109/SoCPaR.2011.6089283.

Changeat, Q. et al. (2020). "Alfnoor: A Retrieval Simulation of the Ariel Target List". In: *The Astronomical Journal*. DOI: 10.3847/1538-3881/ab9a53.

Changeat, Q. et al. (2022). "Five Key Exoplanet Questions Answered via the Analysis of 25 Hot-Jupiter Atmospheres in Eclipse". In: *The Astrophysical Journal Supplement Series*. DOI: 10.3847/1538-4365/ac5cc2.

Chawla, N. V. et al. (2002). "SMOTE: Synthetic Minority Over-sampling Technique". In: *Journal of Artificial Intelligence Research*.

Chollet, F. et al. (2015). *Keras*. URL: https://keras.io.

D'Isanto, A. and K. L. Polsterer (2018). "Photometric redshift estimation via deep learning". In: *Astronomy & Astrophysics*. DOI: 10.1051/0004-6361/201731326.

Domínguez Sánchez, H. et al. (2018). "Improving galaxy morphologies for SDSS with Deep Learning". In: *Monthly Notices of the Royal Astronomical Society*. DOI: 10.1093/mnras/sty338.

Edwards, B. et al. (2022). "Exploring the Ability of Hubble Space Telescope WFC3 G141 to Uncover Trends in Populations of Exoplanet Atmospheres Through a Homogeneous Transmission Survey of 70 Gaseous Planets". In: *The Astrophysical Journal Supplement Series*. DOI: 10.3847/1538-4365/ac9f1a.

Feroz, F. et al. (2019). "Importance Nested Sampling and the MultiNest Algorithm". In: *The Open Journal of Astrophysics*. DOI: 10.21105/astro.1306.2144.

Feroz, F. and M. P. Hobson (2008). "Multimodal nested sampling: an efficient and robust alternative to Markov Chain Monte Carlo methods for astronomical data analyses". In: *Monthly Notices of the Royal Astronomical Society*. DOI: 10.1111/j.1365-2966.2007.12353.x.

Fisher, C. and K. Heng (2018). "Retrieval analysis of 38 WFC3 transmission spectra and resolution of the normalization degeneracy". In: *Monthly Notices of the Royal Astronomical Society*. DOI: 10.1093/mnras/sty2550.

Gaia Collaboration et al. (2016). "The Gaia mission". In: *Astronomy & Astrophysics*. DOI: 10.1051/0004-6361/201629272.

Gal, Y. et al. (2017). "Deep Bayesian Active Learning with Image Data". In: *Proceedings of the 34th International Conference on Machine Learning*.

Gal, Y. and Z. Ghahramani (2016a). "Bayesian convolutional neural networks with Bernoulli approximate variational inference". In: *4th International Conference on Learning Representations, Workshop Track Proceedings*.

Gal, Y. and Z. Ghahramani (2016b). "Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning". In: *Proceedings of the 33rd International Conference on Machine Learning*.

Gang, Z. et al. (2012). "LAMOST spectral survey — An overview". In: *Research in Astronomy and Astrophysics*. DOI: 10.1088/1674-4527/12/7/002.

Gawlikowski, J. et al. (2023). "A survey of uncertainty in deep neural networks". In: *Artificial Intelligence Review*. DOI: 10.1007/s10462-023-10562-9.

George, D. and E. A. Huerta (2018). "Deep neural networks to enable real-time multimessenger astrophysics". In: *Physical Review D*. DOI: 10.1103/PhysRevD.97.044039.

Glazebrook, K., A. R. Offer, and K. Deeley (1998). "Automatic Redshift Determination by Use of Principal Component Analysis. I. Fundamentals". In: *The Astrophysical Journal*. DOI: 10.1086/305039.

Glorot, X. et al. (2011). "Domain Adaptation for Large-Scale Sentiment Classification: A Deep Learning Approach". In: *28th International Conference on Machine Learning*.

Gneiting, T. et al. (2007). "Probabilistic forecasts, calibration and sharpness". In: *Journal of the Royal Statistical Society Series B: Statistical Methodology*.

Goodfellow, I. et al. (2016). *Deep learning*. MIT Press.

Gray, R. O. and J. C. Corbally (2009). *Stellar Spectral Classification*. Princeton University Press.

Gupta, K. D. et al. (2016). "Automated supernova Ia classification using adaptive learning techniques". In: *2016 IEEE Symposium Series on Computational Intelligence (SSCI)*. DOI: 10.1109/SSCI.2016.7849951.

Hanuschik, R. W., J. R. Kozok, and D. Kaiser (1988). "High-resolution emission-line spectroscopy of Be stars: III. Balmer line profiles". In: *Astronomy & Astrophysics*.

Heiter, U. (2014). *Air-to-vacuum conversion*. URL: http://www.astro.uu.se/valdwiki/Air-to-vacuum%20conversion.

Hennawi, J. F. and J. X. Prochaska (2007). "Quasars Probing Quasars. II. The Anisotropic Clustering of Optically Thick Absorbers around Quasars". In: *The Astrophysical Journal*. DOI: 10.1086/509770.

Hersbach, H. (2000). "Decomposition of the Continuous Ranked Probability Score for Ensemble Prediction Systems". In: *Weather and Forecasting*. DOI: 10.1175/1520-0434(2000)015<0559:DOTCRP>2.0.CO;2.

Hewett, P. C. and V. Wild (2010). "Improved redshifts for SDSS quasar spectra". In: *Monthly Notices of the Royal Astronomical Society*. DOI: 10.1111/j.1365-2966.2010.16648.x.

Hou, W. et al. (2016). "A catalog of early-type emission-line stars and Hα line profiles from LAMOST DR2". In: *Research in Astronomy and Astrophysics*. DOI: 10.1088/1674-4527/16/9/138.

Ishida, E. E. O. et al. (2019a). *Active Anomaly Detection for time-domain discoveries*. DOI: 10.1051/0004-6361/202037709.

Ishida, E. E. O. et al. (2019b). "Optimizing spectroscopic follow-up strategies for supernova photometric classification with active learning". In: *Monthly Notices of the Royal Astronomical Society*. DOI: 10.1093/mnras/sty3015.

Ivezić, Ž. et al. (2019). "LSST: From Science Drivers to Reference Design and Anticipated Data Products". In: *The Astrophysical Journal*. DOI: 10.3847/1538-4357/ab042c.

Kang, W. and S.-G. Lee (2012). "Tool for Automatic Measurement of Equivalent width (TAME)". In: *Monthly Notices of the Royal Astronomical Society*. DOI: 10.1111/j.1365-2966.2012.21613.x.

Killestein, T. L. et al. (2021). "Transient-optimized real-bogus classification with Bayesian convolutional neural networks – sifting the GOTO candidate stream". In: *Monthly Notices of the Royal Astronomical Society*. DOI: 10.1093/mnras/stab633.

Kingma, D. P. and J. Ba (2015). "Adam: A Method for Stochastic Optimization". In: *3rd International Conference on Learning Representations*.

Kogure, T. and K.-C. Leung (2007). *The Astrophysics of Emission-Line Stars*. Springer. DOI: 10.1007/978-0-387-68995-1.

Krizhevsky, A. et al. (2012). "ImageNet Classification with Deep Convolutional Neural Networks". In: *Advances in Neural Information Processing Systems 25*.

Kügler, S. D., K. Polsterer, and M. Hoecker (2015). "Determining spectroscopic redshifts by using k nearest neighbor regression I. Description of method and analysis". In: *Astronomy & Astrophysics*. DOI: 10.1051/0004-6361/201424801.

Kuleshov, V. et al. (2018). "Accurate Uncertainties for Deep Learning Using Calibrated Regression". In: *Proceedings of the 35th International Confer-*

ence on Machine Learning. URL: https://proceedings.mlr.press/v80/kuleshov18a.html.

Kurosawa, R., T. J. Harries, and N. H. Symington (2006). "On the formation of Hα line emission around classical T Tauri stars". In: *Monthly Notices of the Royal Astronomical Society*. DOI: 10.1111/j.1365-2966.2006.10527.x.

Lakshminarayanan, B. et al. (2017). "Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles". In: *Advances in Neural Information Processing Systems 30*.

LeCun, Y. et al. (1989). "Backpropagation applied to handwritten zip code recognition". In: *Neural Computation*. DOI: 10.1162/neco.1989.1.4.541.

Lee, Y. S. et al. (2008). "The SEGUE Stellar Parameter Pipeline. I. Description and Comparison of Individual Methods". In: *The Astronomical Journal*. DOI: 10.1088/0004-6256/136/5/2022.

Lee, Y. S. et al. (2015). "Application of the SEGUE Stellar Parameter Pipeline to LAMOST Stellar Spectra". In: *The Astrophysical Journal*. DOI: 10.1088/0004-6256/150/6/187.

Leung, H. W. and J. Bovy (2018). "Deep learning of multi-element abundances from high-resolution spectroscopic data". In: *Monthly Notices of the Royal Astronomical Society*. DOI: 10.1093/mnras/sty3217.

Levasseur, L. P. et al. (2017). "Uncertainties in Parameters Estimated with Neural Networks: Application to Strong Gravitational Lensing". In: *The Astrophysical Journal Letters*. DOI: 10.3847/2041-8213/aa9704.

Lin, C.-C. et al. (2015). "Searching for classical Be stars in LAMOST DR1". In: *Research in Astronomy and Astrophysics*. DOI: 10.1088/1674-4527/15/8/015.

Liu, P. et al. (2022). "A Survey on Active Deep Learning: From Model Driven to Data Driven". In: *ACM Computing Surveys*. DOI: 10.1145/3510414.

Lyke, B. W. et al. (2020). "The Sloan Digital Sky Survey Quasar Catalog: Sixteenth Data Release". In: *The Astrophysical Journal, Supplement Series*. DOI: 10.3847/1538-4365/aba623.

Lyon, R. J. et al. (2016). "Fifty years of pulsar candidate selection: from simple filters to a new principled real-time classification approach". In: *Monthly Notices of the Royal Astronomical Society*. DOI: 10.1093/mnras/stw656.

Madhusudhan, N. (2018). "Atmospheric Retrieval of Exoplanets". In: *Handbook of Exoplanets*. Springer International Publishing.

Mansfield, M. et al. (2021). "A unique hot Jupiter spectral sequence with evidence for compositional diversity". In: *Nature Astronomy*. DOI: 10.1038/s41550-021-01455-4.

Marel, R. P. van der and P. Guhathakurta (2008). "M31 Transverse Velocity and Local Group Mass from Satellite Kinematics". In: *The Astrophysical Journal*. DOI: 10.1086/533430.

Mas des Bourboux, H. du (2021). *redvsblue: Quasar and emission line redshift fitting*. Astrophysics Source Code Library. URL: https://ascl.net/2106.017.

Matchev, K. T. et al. (2022). "Analytical Modeling of Exoplanet Transit Spectroscopy with Dimensional Analysis and Symbolic Regression". In: *The Astrophysical Journal*. DOI: 10.3847/1538-4357/ac610c.

Möller, A. and T. de Boissière (2019). "SuperNNova: An open-source framework for Bayesian, neural network-based supernova classification". In: *Monthly Notices of the Royal Astronomical Society*. DOI: 10.1093/mnras/stz3312.

Mugnai, L. V. et al. (2020). "ArielRad: the *Ariel* Radiometric Model". In: *Experimental Astronomy*. DOI: 10.1007/s10686-020-09676-7.

Mugnai, L. V. et al. (2021). "Alfnoor: Assessing the Information Content of Ariel's Low-resolution Spectra with Planetary Population Studies". In: *The Astronomical Journal*. DOI: 10.3847/1538-3881/ac2e92.

Nikolaou, N. et al. (2023). "Lessons learned from the 1st Ariel Machine Learning Challenge: Correcting transiting exoplanet light curves for stellar spots". In: *RAS Techniques and Instruments*. DOI: 10.1093/rasti/rzad050.

Nix, D. A. and A. S. Weigend (1994). "Estimating the mean and variance of the target probability distribution". In: *Proceedings of 1994 IEEE International Conference on Neural Networks*. DOI: 10.1109/ICNN.1994.374138.

Pan, S. J. and Q. Yang (2010). "A Survey on Transfer Learning". In: *IEEE Transactions on Knowledge and Data Engineering*. DOI: 10.1109/TKDE.2009.191.

Pâris, I. et al. (2017). "The Sloan Digital Sky Survey Quasar Catalog: Twelfth data release". In: *Astronomy & Astrophysics*. DOI: 0004-6361/201527999.

Pâris, I. et al. (2018). "The Sloan Digital Sky Survey Quasar Catalog: Fourteenth data release". In: *Astronomy & Astrophysics*. DOI: 10.1051/0004-6361/201732445.

Pinhas, A. et al. (2019). "H_2O abundances and cloud properties in ten hot giant exoplanets". In: *Monthly Notices of the Royal Astronomical Society*. DOI: 10.1093/mnras/sty2544.

Podsztavek, **O.** (2017). "Deep Learning in Large Astronomical Spectra Archives". Bachelor's thesis. Czech Technical University in Prague.

Podsztavek, **O.** et al. (2024). "Automatic Miscalibration Diagnosis: Interpreting Probability Integral Transform (PIT) Histograms". In: *ESANN 2024 proceedings*.

Podsztavek, **O.** and P. Škoda (2019). *Ondřejov Dataset*. Zenodo. DOI: 10.5281/zenodo.2640971.

Podsztavek, **O.**, P. Škoda, and P. Tvrdík (2021). "Transfer Learning in Large Spectroscopic Surveys". In: *Astronomical Data Analysis Software and Systems XXX*.

Podsztavek, **O.**, P. Škoda, and P. Tvrdík (2022). "Spectroscopic redshift determination with Bayesian convolutional networks". In: *Astronomy and Computing*. DOI: 10.1016/j.ascom.2022.100615.

Podsztavek, **O.**, P. Škoda, and P. Tvrdík (2024). "Prototype of Interactive Visualisation Tool for Bayesian Active Deep Learning". In: *Astronomical Data Analysis Software and Systems XXXI*.

Pollacco, D. L. et al. (2006). "The WASP Project and the SuperWASP Cameras". In: *Publications of the Astronomical Society of the Pacific*. DOI: 10.1086/508556.

Porter, J. M. and T. Rivinius (2003). "Classical Be Stars". In: *Publications of the Astronomical Society of the Pacific*. DOI: 10.1086/378307.

Rastegarnia, F. et al. (2022). "Deep learning in searching the spectroscopic redshift of quasars". In: *Monthly Notices of the Royal Astronomical Society*. DOI: 10.1093/mnras/stac076.

Rastgoo, M. et al. (2016). "Tackling the problem of data imbalancing for melanoma classification". In: *3rd International Conference on Bioimaging*.

Redmon, J. and A. Farhadi (2017). "YOLO9000: Better, Faster, Stronger". In: *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition*.

Al-Refaie, A. F. et al. (2021). "TauREx 3: A Fast, Dynamic, and Extendable Framework for Retrievals". In: *The Astrophysical Journal*. DOI: 10.3847/1538-4357/ac0252.

Reipurth, B., A. Pedrosa, and M. T. V. T. Lago (1996). "Hα emission in pre-main sequence stars. I. An atlas of line profiles". In: *Astronomy and Astrophysics Supplement Series*. DOI: 10.1051/aas:1996286.

Ren, P. et al. (2021). "A Survey of Deep Active Learning". In: *ACM Computing Surveys*. DOI: 10.1145/3472291.

Richards, J. W. et al. (2012). "Active Learning to Overcome Sample Selection Bias: Application to Photometric Variable Star Classification". In: *The Astrophysical Journal*. DOI: 10.1088/0004-637X/744/2/192.

Ricker, G. R. et al. (2015). "Transiting Exoplanet Survey Satellite (TESS)". In: *Journal of Astronomical Telescopes, Instruments, and Systems*. DOI: 10.1117/1.JATIS.1.1.014003.

Roudier, G. M. et al. (2021). "Disequilibrium Chemistry in Exoplanet Atmospheres Observed with the Hubble Space Telescope". In: *The Astronomical Journal*. DOI: 10.3847/1538-3881/abfdad.

Russakovsky, O. et al. (2015). "ImageNet Large Scale Visual Recognition Challenge". In: *International Journal of Computer Vision*. DOI: 10.1007/s11263-015-0816-y.

Schmidt, M. (1963). "3C 273 : A Star-Like Object with Large Red-Shift". In: *Nature*. DOI: 10.1038/1971040a0.

Schneider, D. P. et al. (2010). "The Sloan Digital Sky Survey quasar catalog. V. Seventh data release". In: *The Astronomical Journal*. DOI: 10.1088/0004-6256/139/6/2360.

Silaj, J. et al. (2010). "A Systematic Study of Hα Profiles of Be Stars". In: *The Astrophysical Journal Supplement Series*. DOI: 10.1088/0067-0049/187/1/228.

Simonyan, K. and A. Zisserman (2015). "Very Deep Convolutional Networks for Large-Scale Image Recognition". In: *3rd International Conference on Learning Representations*.

Sing, D. K. et al. (2016). "A continuum from clear to cloudy hot-Jupiter exoplanets without primordial water depletion". In: *Nature*. DOI: 10.1038/nature16068.

Skilling, J. (2006). "Nested sampling for general Bayesian computation". In: *Bayesian Analysis*. DOI: 10.1063/1.1835238.

Škoda, P. and **O. Podsztavek** (n.d.). "Consistency check of automatic pipeline measurements of quasar redshifts with Bayesian convolutional networks". In: *Astronomical Data Analysis Software and Systems XXXII*. In print.

Škoda, P., **O. Podsztavek**, and P. Tvrdík (2020a). "Active deep learning method for the discovery of objects of interest in large spectroscopic surveys". In: *Astronomy & Astrophysics*. DOI: 10.1051/0004-6361/201936090.

Škoda, P., **O. Podsztavek**, and P. Tvrdík (2020b). "VO-supported Active Deep Learning as a New Methodology for the Discovery of Objects of Interest in Big Surveys". In: *Astronomical Data Analysis Software and Systems XXIX*.

Soboczenski, F. et al. (2018). *Bayesian deep Learning for exoplanet atmospheric retrieval*.

Solorio, T. et al. (2005). "An active instance-based machine learning method for stellar population studies". In: *Monthly Notices of the Royal Astronomical Society*. DOI: 10.1111/j.1365-2966.2005.09456.x.

Srivastava, N. et al. (2014). "Dropout: A Simple Way to Prevent Neural Networks from Overfitting". In: *Journal of Machine Learning Research*.

Stivaktakis, R. et al. (2020). "Convolutional neural networks for spectroscopic redshift estimation on Euclid Data". In: *IEEE Transactions on Big Data*. DOI: 10.1109/TBDATA.2019.2934475.

Taigman, Y. et al. (2017). "Unsupervised Cross-Domain Image Generation". In: *5th International Conference on Learning Representations*.

Taylor, M. B. (2005). "TOPCAT & STIL: Starlink Table/VOTable Processing Software". In: *Astronomical Data Analysis Software and Systems XIV*.

Tinetti, G. et al. (2021). *Ariel: Enabling planetary science across light-years*. Tech. rep. European Space Agency.

Traven, G. et al. (2015). "The Gaia-ESO Survey: Catalogue of Hα emission stars". In: *Astronomy & Astrophysics*. DOI: 10.1051/0004-6361/201525857.

Tsiaras, A. et al. (2018). "A Population Study of Gaseous Exoplanets". In: *The Astronomical Journal*. DOI: 10.3847/1538-3881/aaaf75.

Urry, C. M. and P. Padovani (1995). "Unified Schemes for Radio-Loud Active Galactic Nuclei". In: *Publications of the Astronomical Society of the Pacific*. DOI: 10.1086/133630.

Vanden Berk, D. E. et al. (2001). "Composite Quasar Spectra from the Sloan Digital Sky Survey". In: *The Astronomical Journal*. DOI: 10.1086/321167.

Vilalta, R. et al. (2019). "A General Approach to Domain Adaptation with Applications in Astronomy". In: *Publications of the Astronomical Society of the Pacific*. DOI: 10.1088/1538-3873/aaf1fc.

Walmsley, M. et al. (2020). "Galaxy Zoo: Probabilistic morphology through Bayesian CNNs and active learning". In: *Monthly Notices of the Royal Astronomical Society*. DOI: 10.1093/mnras/stz2816.

Wan, T. et al. (2023). "A Survey of Deep Active Learning for Foundation Models". In: *Intelligent Computing*. DOI: 10.34133/icomputing.0058.

Waters, C. Z. and J. K. Hollek (2013). "ROBOSPECT: Automated Equivalent Width Measurement". In: *Publications of the Astronomical Society of the Pacific*. DOI: 10.1086/673311.

Wu, M. et al. (2022). "Active Learning for Computer Vision Tasks: Methodologies, Applications, and Challenges". In: *Applied Sciences*. DOI: 10.3390/app12168103.

Wu, Y. et al. (2011). "Automatic determination of stellar atmospheric parameters and construction of stellar spectral templates of the Guoshoujing Telescope (LAMOST)". In: *Research in Astronomy and Astrophysics*. DOI: 10.1088/1674-4527/11/8/006.

Yip, K. H. et al. (2022a). "To Sample or Not To Sample: Retrieving Exoplanetary Spectra with Variational Inference and Normalising Flows". In: *The Astrophysical Journal*. DOI: 10.3847/1538-4357/ad063f.

Yip, K. H., Q. Changeat, I. Waldmann, E. B. Unlu, R. T. Forestano, A. Roman, K. Matcheva, K. T. Matchev, S. Stefanov, **O. Podsztavek**, et al. (2022b). "Lessons Learned from Ariel Data Challenge 2022 – Inferring Physical Properties of Exoplanets From Next-Generation Telescopes". In: *Proceedings of the NeurIPS 2022 Competitions Track*.

Zickgraf, F.-J. (2003). "Kinematical structure of the circumstellar environments of galactic B[e]-type stars". In: *Astronomy & Astrophysics*. DOI: 10.1051/0004-6361:20030999.

Author's Bibliography

Refereed related to dissertation

Podsztavek, **O.** et al. (2024). "Automatic Miscalibration Diagnosis: Interpreting Probability Integral Transform (PIT) Histograms". In: *ESANN 2024 proceedings*.

Podsztavek, **O.**, P. Škoda, and P. Tvrdík (2022). "Spectroscopic redshift determination with Bayesian convolutional networks". In: *Astronomy and Computing*. DOI: 10.1016/j.ascom.2022.100615.

Škoda, P., **O. Podsztavek**, and P. Tvrdík (2020a). "Active deep learning method for the discovery of objects of interest in large spectroscopic surveys". In: *Astronomy & Astrophysics*. DOI: 10.1051/0004-6361/201936090.

Related to winning a competition and to dissertation

Yip, K. H., Q. Changeat, I. Waldmann, E. B. Unlu, R. T. Forestano, A. Roman, K. Matcheva, K. T. Matchev, S. Stefanov, **O. Podsztavek**, et al. (2022b). "Lessons Learned from Ariel Data Challenge 2022 – Inferring Physical Properties of Exoplanets From Next-Generation Telescopes". In: *Proceedings of the NeurIPS 2022 Competitions Track*.

Other related to dissertation

Podsztavek, **O.**, P. Škoda, and P. Tvrdík (2021). "Transfer Learning in Large Spectroscopic Surveys". In: *Astronomical Data Analysis Software and Systems XXX*.

Podsztavek, **O.**, P. Škoda, and P. Tvrdík (2024). "Prototype of Interactive Visualisation Tool for Bayesian Active Deep Learning". In: *Astronomical Data Analysis Software and Systems XXXI*.

Škoda, P. and **O. Podsztavek** (n.d.). "Consistency check of automatic pipeline measurements of quasar redshifts with Bayesian convolutional networks". In: *Astronomical Data Analysis Software and Systems XXXII*. In print.

Škoda, P., **O. Podsztavek**, and P. Tvrdík (2020b). "VO-supported Active Deep Learning as a New Methodology for the Discovery of Objects of Interest in Big Surveys". In: *Astronomical Data Analysis Software and Systems XXIX*.

Appendix A

Method for Prediction of Spectroscopic Redshift

A.1 Scatter plots of redshifts

Figure A.1 compares spectroscopic redshifts determined by Bayesian SZNet, SDSS pipeline, `redvsblue` algorithm, and QuasarNET to `Z_10K` redshifts. In the case of ideal determinations, all points should be on a diagonal line. However, all methods exhibit a kind of systematic errors, i.e. the lines with different angles, which reveal that spectral lines were misidentified. Bayesian SZNet systematically predicts $\hat{z} = 0$ for non-zero `Z_10K` redshifts less than 2. The SDSS pipeline (`Z_PIPE`) problem is that it systematically measures higher redshifts for some spectra. The `redvsblue` algorithm (`Z_PCA`) performs similarly to the SDSS pipeline because the algorithm fine-tunes its measurements. Lastly, QuasarNET (`Z_QN`) systematically predicts stars to have non-zero redshifts and predicts poorly `Z_10K` redshifts less than 2.

A.2 Redshift predictions catalogue

We provide the `dr16q_superset_redshift.csv` catalogue with redshifts from Bayesian SZNet on Zenodo, at `https://doi.org/10.5281/zenodo.5173824`. It lists 1 440 573 redshift predictions for DR16Q superset spectra by Bayesian SZNet in the `z_pred` column with their associated predictive variances in the `variance` column. Furthermore, we provide all 256 sampled redshift predictions in `z_pred_1`–`z_pred_256` columns. Other columns are from the DR16Q superset, where lowercase column names correspond to uppercase column names in the DR16Q superset (e.g. the `is_qso_final` column equals the `IS_QSO_FINAL` column in the DR16Q superset). All columns

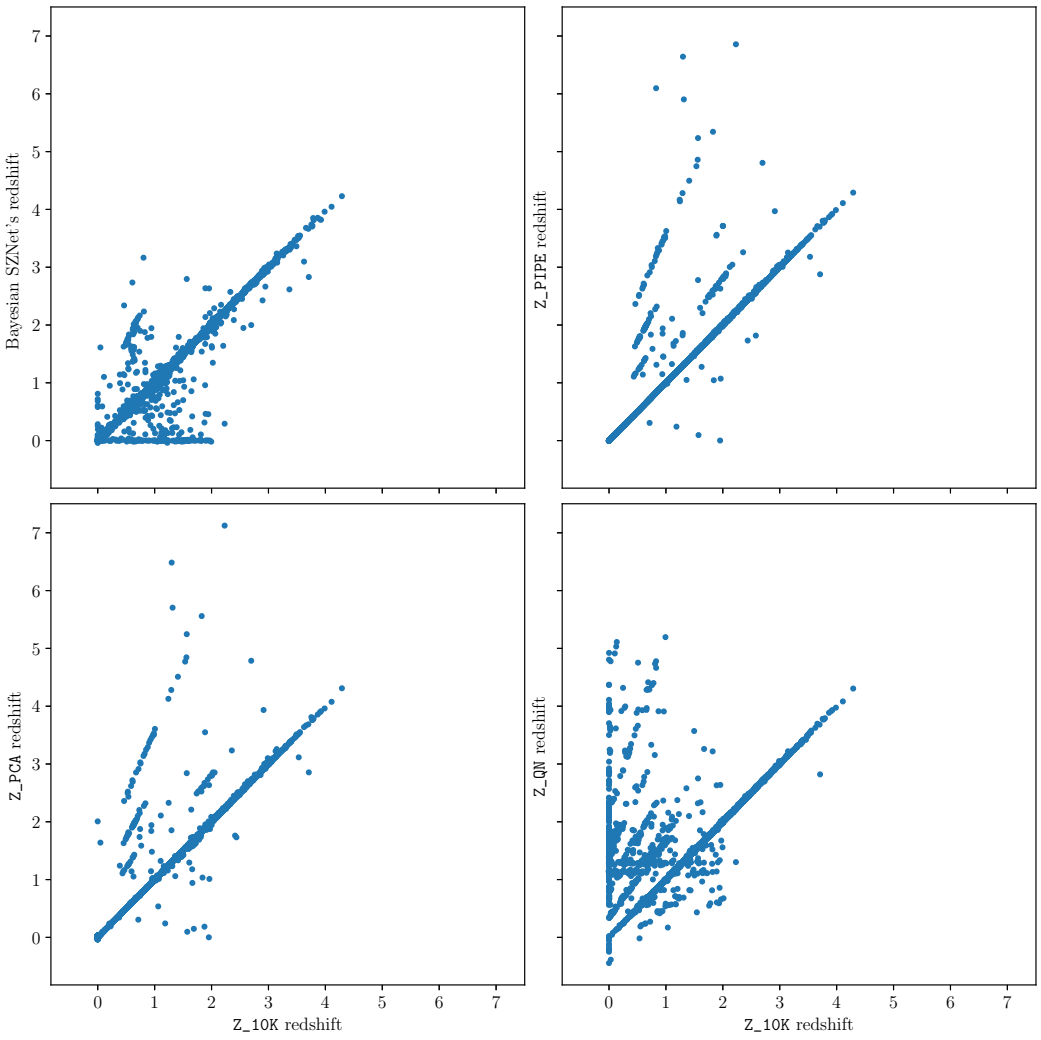

Figure A.1: Scatter plots comparing four different redshift determinations to `Z_10K` redshifts

column no.	name	description
1	plate	spectroscopic plate number
2	mjd	modified Julian day of the spectroscopic observation
3	fiberid	fibre identification number
4	z_pred	redshift from Bayesian SZNet
5	variance	predictive variance associated with redshift from Bayesian SZNet
6	z	primary redshift
7	source_z	origin of the reported redshift in the z column
8	is_qso_final	flag indicating QSOs included in the DR16Q
9	z_vi	redshift from visual inspection
10	z_pipe	redshift from the SDSS pipeline
11	zwarning	quality flag on the redshift from the SDSS pipeline
12	z_dr12q	redshift from the DR12Q visual inspection
13	z_dr7q_sch	redshift from the SDSS DR7 QSO catalogue (Schneider et al. 2010)
14	z_dr6q_hw	redshift from the SDSS DR6 QSO catalogue (Hewett and Wild 2010)
15	z_10k	redshift from the Z_10K subsample
16	z_pca	redshift from the redvsblue algorithm
17	z_qn	redshift from QuasarNET
18–273	z_pred_1–z_pred_256	sampled redshifts from Bayesian SZNet

Table A.1: Description of columns in dr16q_superset_redshift.csv catalogue

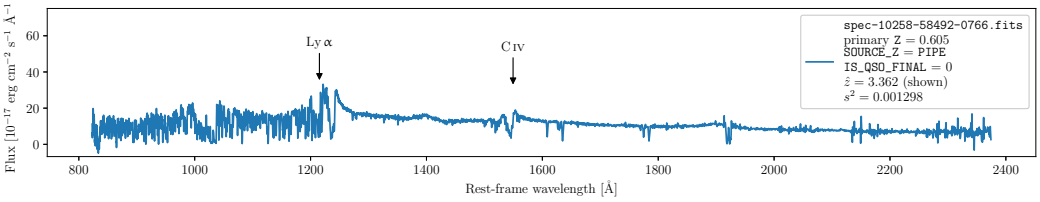

Figure A.2: Spectrum of QSO missed by DR16Q with incorrect redshift measurement of SDSS pipeline

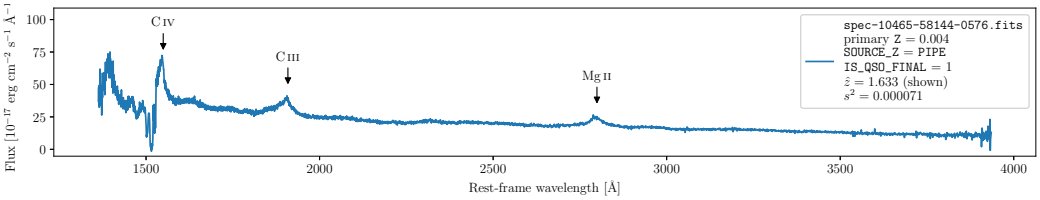

Figure A.3: Spectrum of QSO with incorrect redshift measurement of SDSS pipeline

are described in Table A.1. The catalogue is in the comma-separated values (CSV) format and is sorted according to the `variance` column so that the most certain predictions are at the top.

A.3 Consistency check examples

Figures A.2–A.6 show spectra of QSOs with incorrect primary `Z`. Moreover, three of them are not included in the DR16Q (`IS_QSO_FINAL` = 0) while they are QSOs. The SDSS pipeline measured the spectra in Figures A.2–A.5 to have an incorrectly low redshift, while the spectrum in Figure A.6 to have an incorrectly high redshift.

Figures A.7–A.9 display stars with incorrect primary `Z`. Figure A.9 shows a spectrum that exhibits emission features which might confuse the SDSS pipeline.

Finally, spectra in Figures A.10 and A.11 illustrate incorrect redshift predictions that Bayesian SZNet has made. However, the redshift prediction in Figure A.10 has a high predictive variance $s^2 = 0.009961$. Figure A.11 displays a spectrum with missing flux values that probably caused the incorrect redshift prediction.

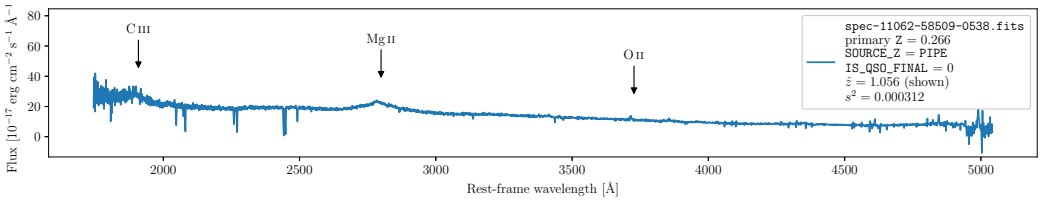

Figure A.4: Spectrum of QSO missed by DR16Q with incorrect redshift measurement of SDSS pipeline

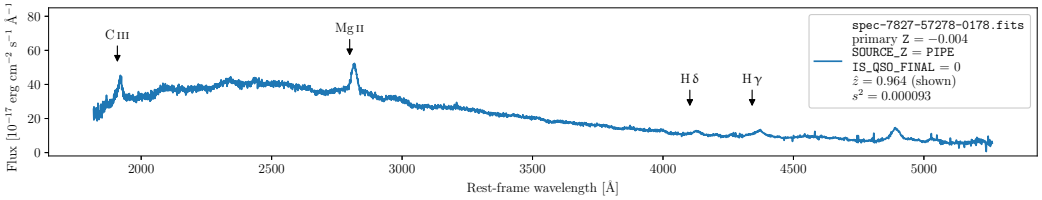

Figure A.5: Spectrum of QSO missed by DR16Q with incorrect redshift measurement of SDSS pipeline

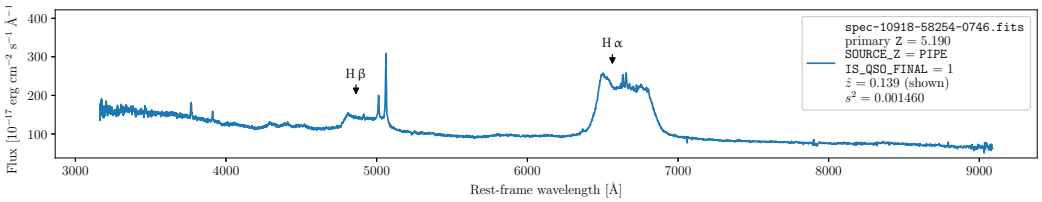

Figure A.6: Spectrum of QSO with incorrect redshift measurement of SDSS pipeline

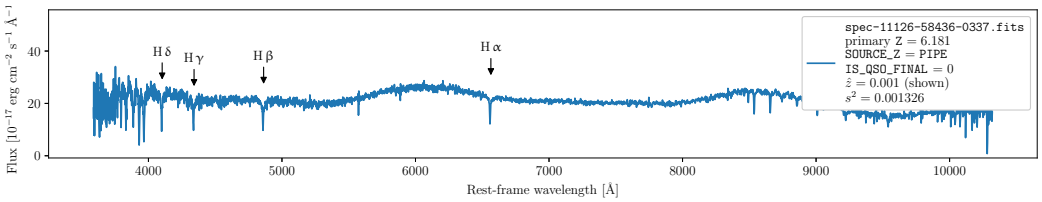

Figure A.7: Spectrum of star with incorrect redshift measurement of SDSS pipeline

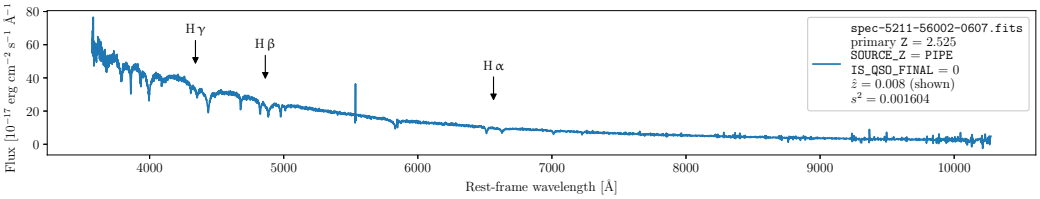

Figure A.8: Spectrum of star with incorrect redshift measurement of SDSS pipeline

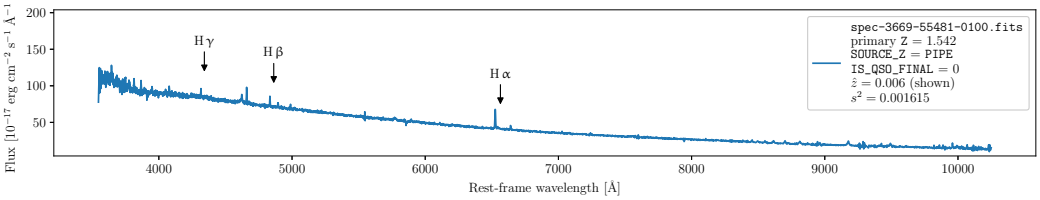

Figure A.9: Spectrum of star with incorrect redshift measurement of SDSS pipeline

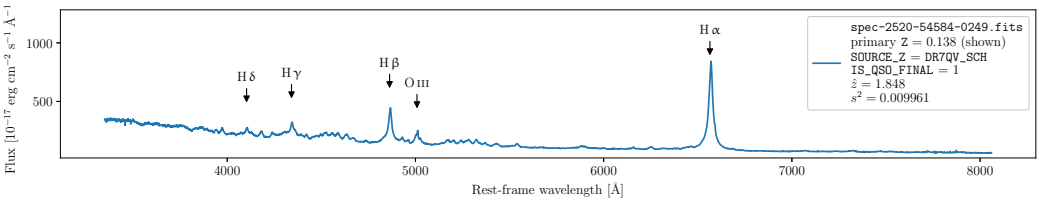

Figure A.10: Incorrect redshift prediction of Bayesian SZNet, but with high predictive variance $s^2 = 0.009961$

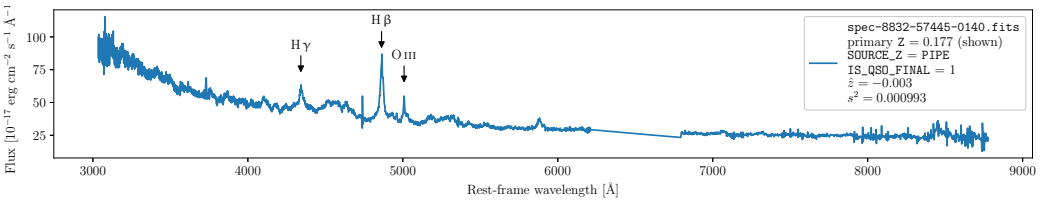

Figure A.11: Incorrect redshift prediction of Bayesian SZNet is probably caused by missing flux values