SDSS redshift prediction based on Bayesian Deep Learning

O.Podsztavek[1] P. Škoda[2,1]

[1]Faculty of Information Technology, Czech Technical University in Prague
[2]Astronomical Institute of the Czech Academy of Sciences, Ondřejov

IVOA Interoperability Meeting, KDD session,
Worldwide , 3rd November 2021

Uncertainties

“Lack of knowledge about the truth”

Aleatoric :

• Due to the random nature of getting data (noise in measurements]

• Cannot be reduced by better understanding

Epistemic :

• Ignorance about he model that generated the data

• We can improve our knowledge by more experiments
(e.g. different network architecture)

• Bayesian deep learning

Bayesian Deep Learning coming to astronomy

Standard deep network classification

Classification with neural networks

$$\mathscr{P}\left(\mathbf{y} \,|\, \mathbf{x}, \mathbf{w}\right)$$

classification data weights

$$NLL = \min_{\mathbf{w}} \sum_{i=1}^{N} - \log \mathscr{P}(\mathbf{y}_i \,|\, \mathbf{x}_i, \mathbf{w})$$

Bayesian Deep Classification

Bayesian neural networks Variational inference

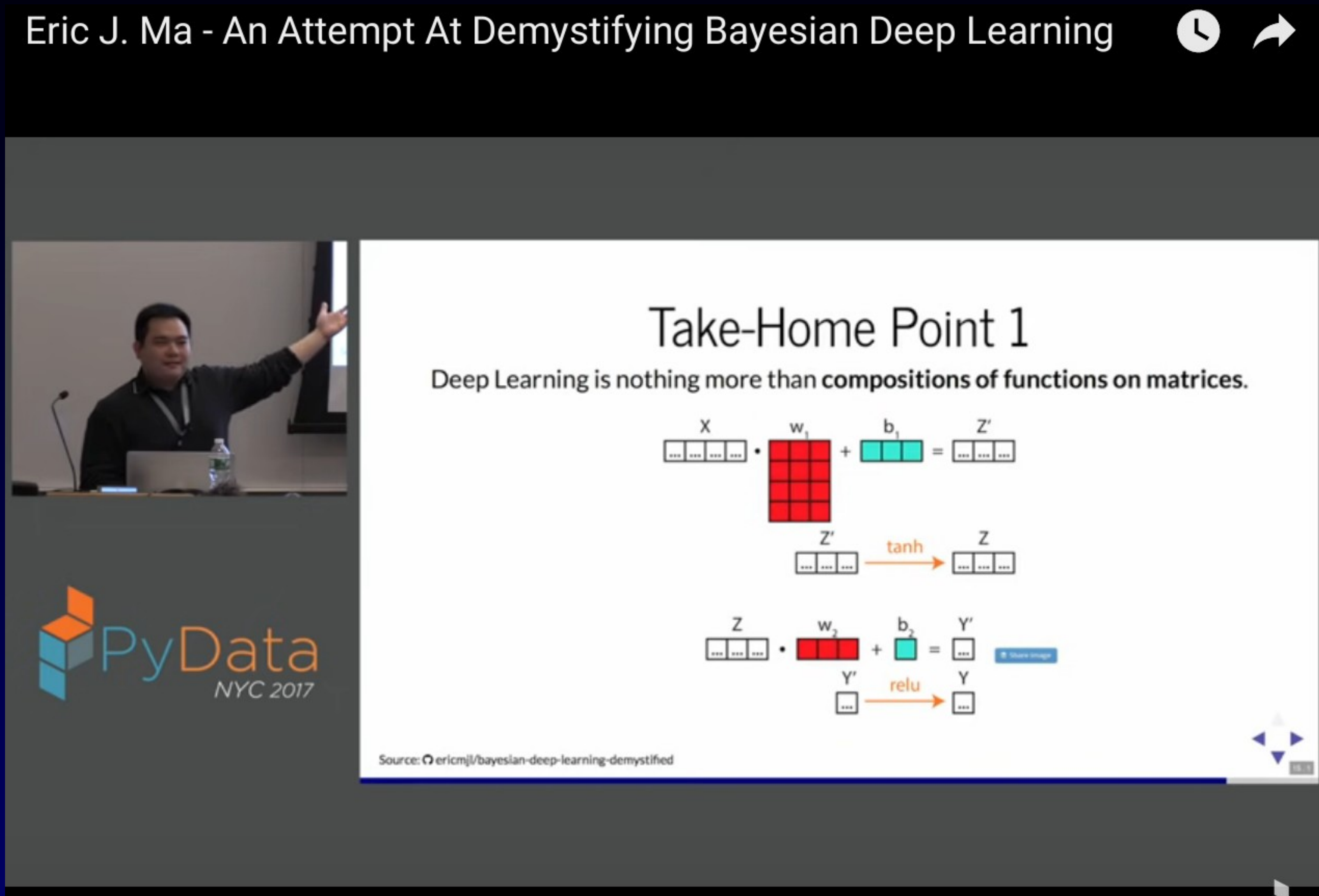
$$\mathscr{P}(\mathbf{y}\,|\,\mathbf{x}) = \int \mathscr{P}(\mathbf{y}\,|\,\mathbf{x}, \mathbf{w})\,\mathscr{P}\left(\mathbf{w}\,|\,\mathscr{D}\right) d\mathbf{w}$$

Posterior is intractable for deep NNs

$$\mathscr{P}(\mathbf{w}\,|\,\mathscr{D}) \approx q(\mathbf{w}\,|\,\theta)$$ variational distribution

Training minimisation $$\hat{\theta} = \min_{\theta} \mathbf{KL}\left(q(\mathbf{w}\,|\,\theta)\,|\,|\,\mathscr{P}(\mathbf{w}\,|\,\mathscr{D})\right)$$

Eric J. Ma on Youtube

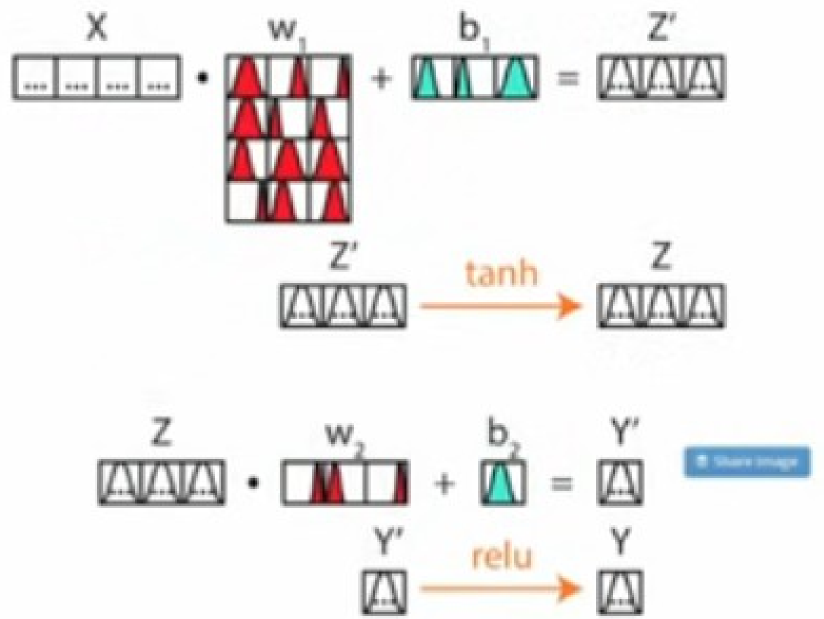


https://www.youtube.com/watch?v=s0S6HFdPtlA

Eric J. Ma on Youtube

Take-Home Point 2

Bayesian deep learning is grounded on **learning a probability distribution for each parameter.**



Source: ericmjl/bayesian-deep-learning-demystified

Bayesian Deep Learning

UNIVERSITY OF CAMBRIDGE

Uncertainty in Deep Learning

Yarin Gal

Department of Engineering
University of Cambridge

This dissertation is submitted for the degree of
Doctor of Philosophy

Gonville and Caius College — September 2016

ARTICLE

Dropout as a Bayesian approximation: representing model uncertainty in deep learning

Authors: Yarin Gal, Zoubin Ghahramani Authors Info & Claims

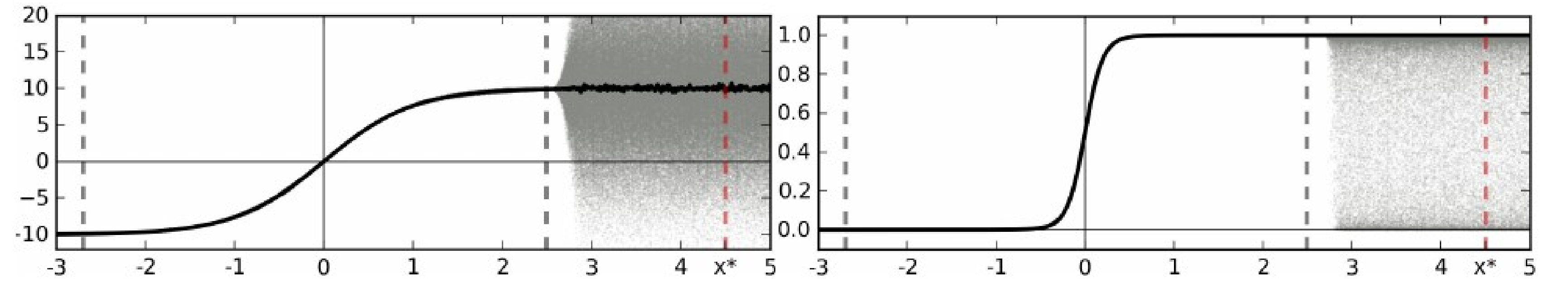
ICML'16: Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48 • June 2016 • Pages 1050–1059

Published: 19 June 2016

On ACM : Gal16.pdf

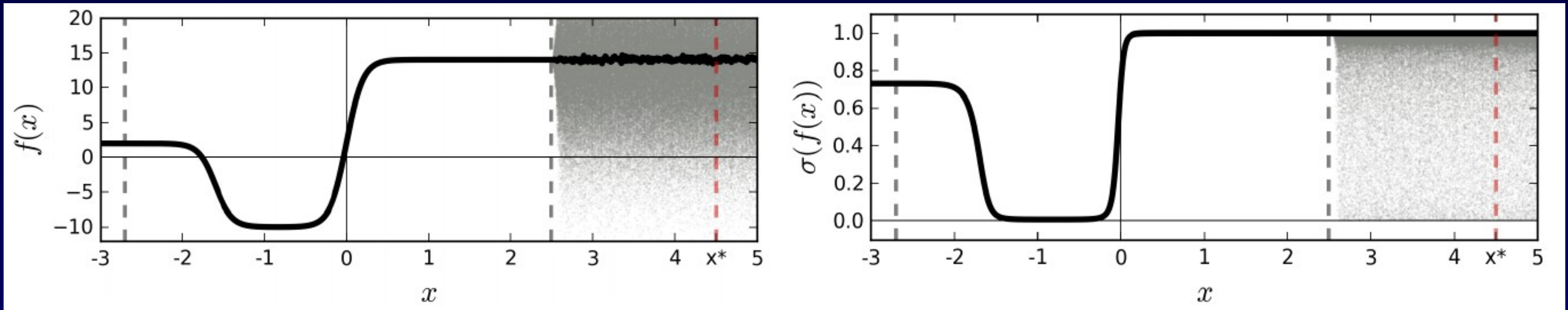
Google: NIPS_2015_deep_learning_uncertainty.pdf

Different picture of softmax



(a) Softmax *input* as a function of data $\mathbf{x}$: $f(\mathbf{x})$ (b) Softmax *output* as a function of data $\mathbf{x}$: $\sigma(f(\mathbf{x}))$

NIPS_2015_deep_learning_uncertainty.pdf



gal16.pdf

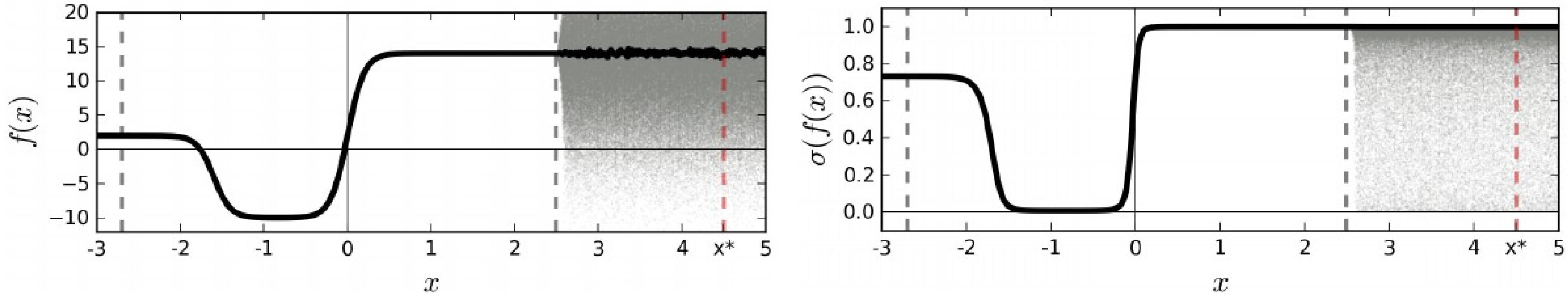
Predictive probability
IS NOT
predictive uncertainty

Standard deep learning tools for regression and classification do not capture model uncertainty.

In classification, predictive probabilities obtained at the end of the pipeline (the softmax output) are often erroneously interpreted as model confidence.

A model can be uncertain in its predictions even with a high softmax output (fig. 1).

Passing a point estimate of a function (solid line 1a) through a softmax (solid line 1b) results in extrapolations with unjustified high confidence for points far from the training data. x* for example would be classified as class 1 with probability 1.



(a) Arbitrary function $f(\mathbf{x})$ as a function of data $\mathbf{x}$ (softmax *input*) (b) $\sigma(f(\mathbf{x}))$ as a function of data $\mathbf{x}$ (softmax *output*)

Figure 1. **A sketch of softmax input and output for an idealised binary classification problem.** Training data is given between the dashed grey lines. Function point estimate is shown with a solid line. Function uncertainty is shown with a shaded area. Marked with a dashed red line is a point x^* far from the training data. Ignoring function uncertainty, point x^* is classified as class 1 with probability 1.

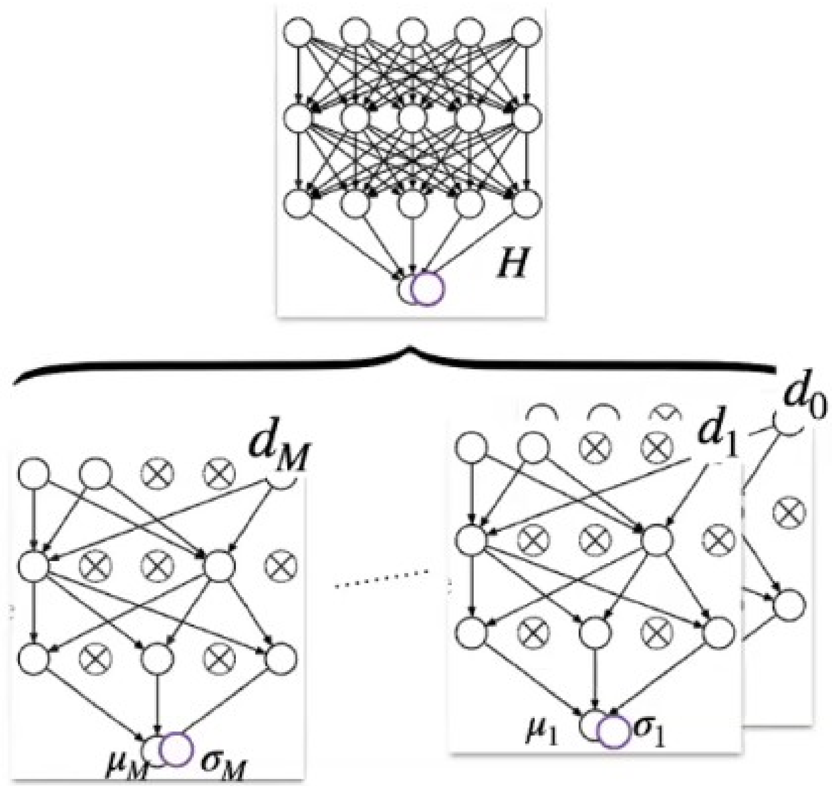
MC Dropout method

Monte Carlo Dropout

- Gal & Ghahramani 2016
- Gal & Ghahramani 2016 (RNNs)
- Gal+2017
- Corina+2018
- Cortes-Ciriano, Andreas 2019

- Premise:
  - Obtain mean and variance from an ensemble of predictions that are generated by **applying dropout at test time**.
  - Considered *approximately* Bayesian via Deep Gaussian Processes
  - *Concrete Dropout* provides a procedure to optimize the dropout in each layer during training.

- Recipe:
  - **Training**
    - Train model H maximizing the log-likelihood
  - **Prediction**
    - Make a prediction μ_i, σ_i from the model H that has a dropout d_i applied
    - Repeat for $i \in \{0,1,\ldots,M\}$ for M



Ensemble of M **predictions** generated from independent samplings via dropout.

Brian Nord
SCMA VII 2021

Spectroscopic redshift experiment

- Prediction of QSO redshift - emission line pattern
- Formulation as regression or classification
- Classification in bins – interval width 0.01

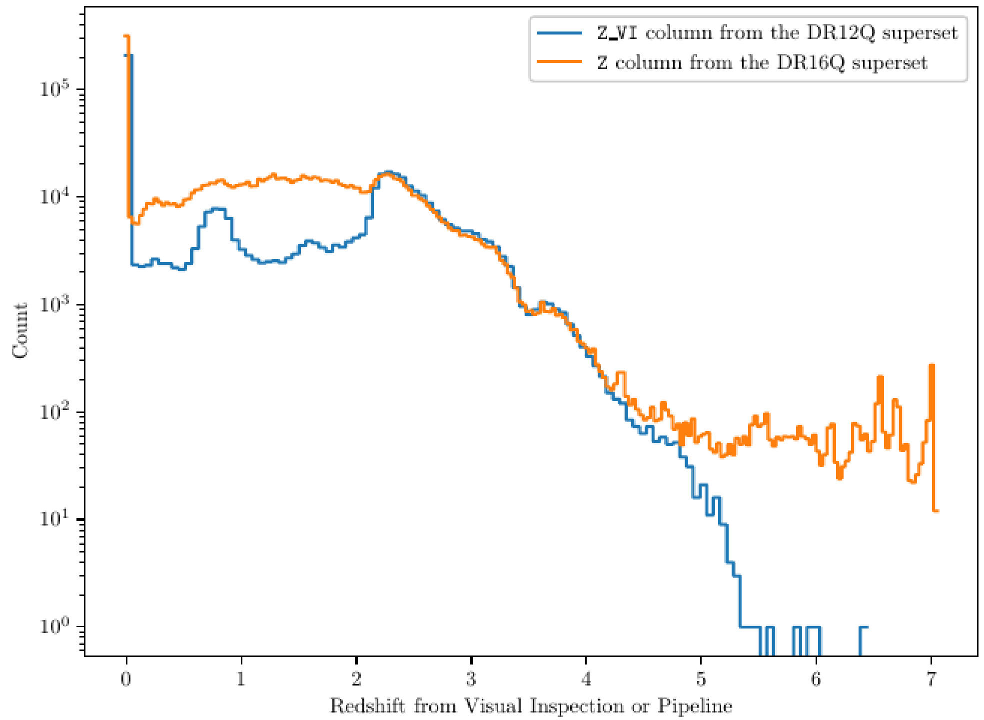
Inspired by Stivaktakis R. et al. (Convolutional Neural Networks for Spectroscopic Redshift Estimation on Euclid Data. In IEEE Transactions on Big Data, vol. 6, no. 3, 2020.)

- Preparation of spectra – continuum normalisation
- Cut and regridding to the same grid
- Rescaling to unit variance zero mean

Spectroscopic redshift experiment

Experimental Data of Bayesian Redshift Prediction

- Trained on *fully human-labelled* 12th Sloan Digital Sky Survey (SDSS) quasar superset (0.5 million human-labelled spectra).
- Generalisation capability is evaluated on the 16th SDSS quasar superset (1.5 million spectra).



Spectroscopic redshift experiment

Metrics to Evaluate Bayesian Redshift Prediction

Given N is the number of test spectra, z is the true redshift, $\hat{z}$ is the predicted redshift, and c is the speed of light:

Root-mean-square (RMS) error $E_{\mathrm{RMS}} = \sqrt{\frac{1}{N}\sum_{n=1}^{N}(\hat{z}_n - z_n)^2}$.

Median Δv Median of the velocity difference: $\Delta v = c \cdot \frac{|\hat{z}-z|}{1+z}$.

Catastrophic z ratio The ratio of redshift predictions with $\Delta v \geq 3000\,\mathrm{km\,s^{-1}}$.

Coverage The ratio of the count of spectra for which we accept predictions of the Bayesian CNN.

Predictive entropy

$$\hat{\mathbb{H}}(y|\boldsymbol{x}', \mathbf{X}, \boldsymbol{y}) = -\sum_{c=1}^{C} \left[\frac{1}{T} \sum_{t=1}^{T} p(y = c|\boldsymbol{x}', \hat{\Theta}_t) \right] \cdot \ln \left[\frac{1}{T} \sum_{t=1}^{T} p(y = c|\boldsymbol{x}', \hat{\Theta}_t) \right],$$

Thresholding

Utilisation of Uncertainty from the Bayesian CNN

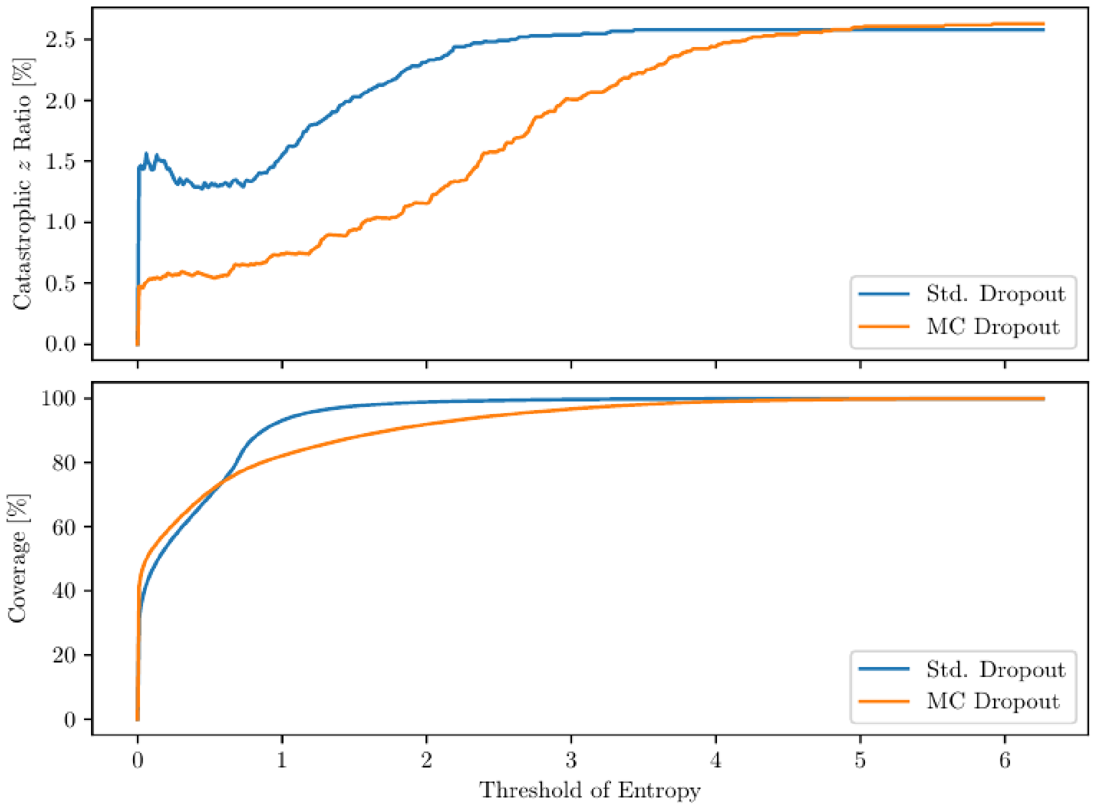
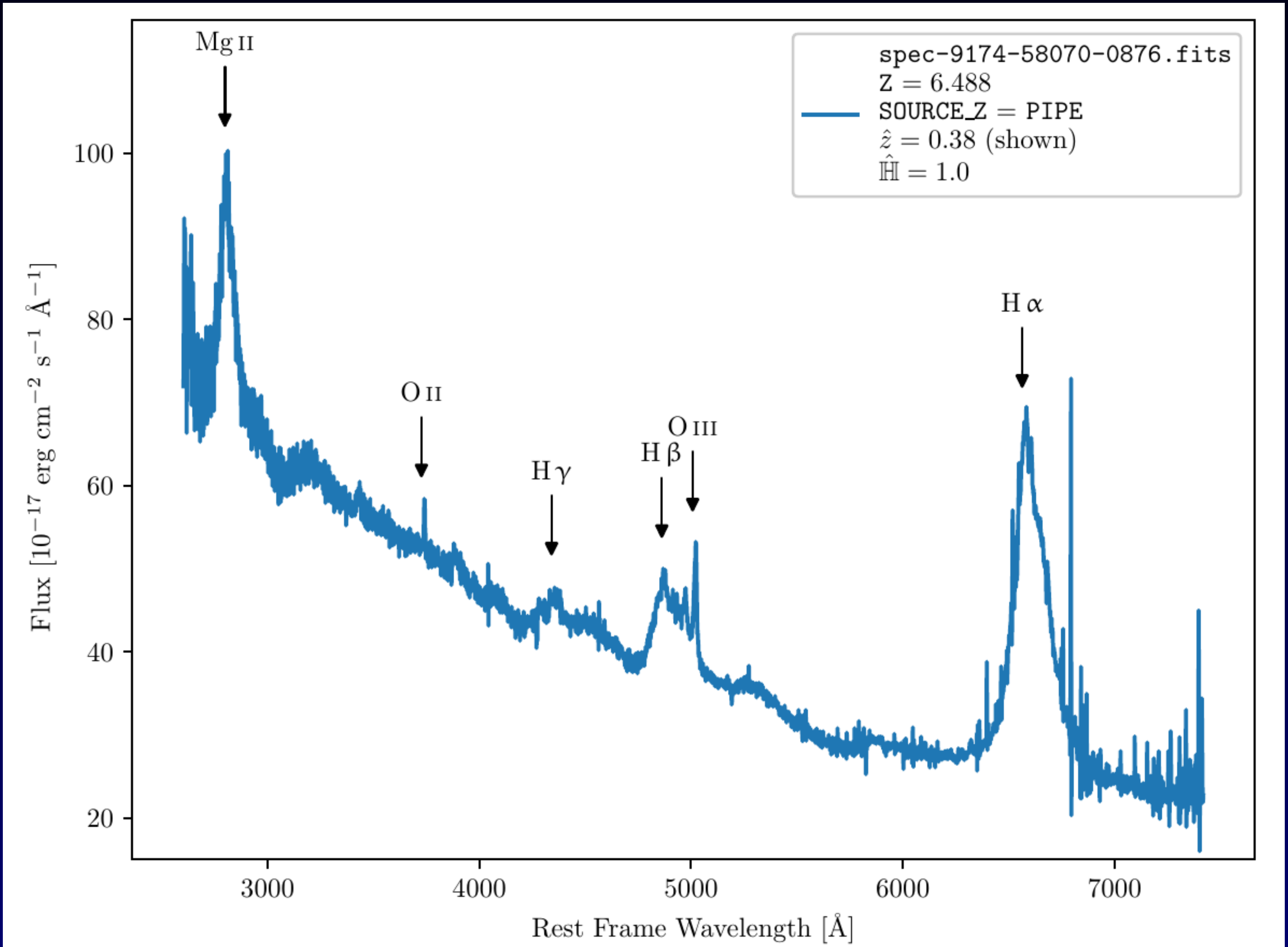
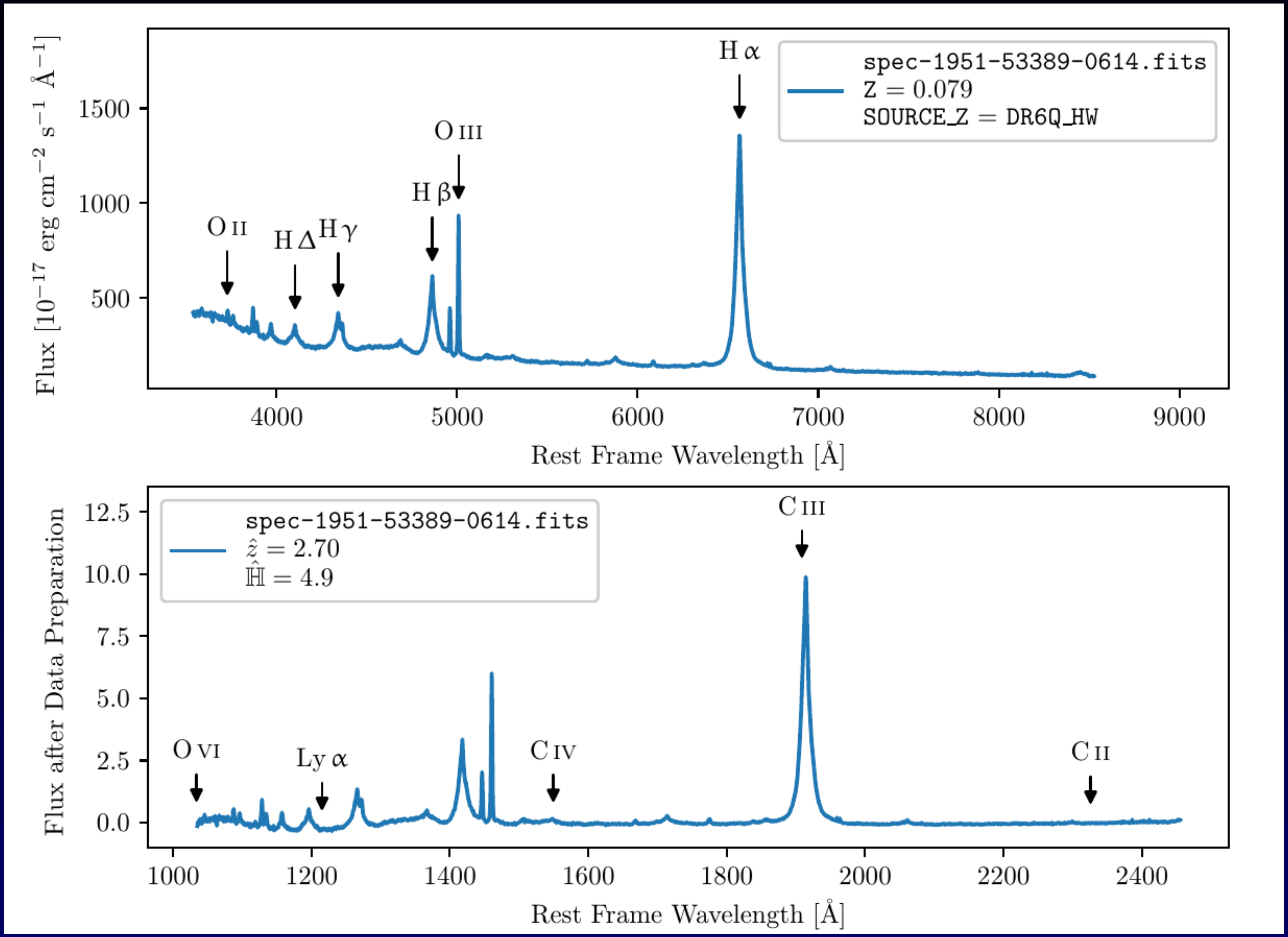


Figure: Dependence of catastrophic z ratio and coverage on a predefined threshold. The plots compare uncertainty in the form of entropy from the Bayesian CNN (MC dropout) and classical CNN (std. dropout).

Wrong SDDS pipeline – high z QSO

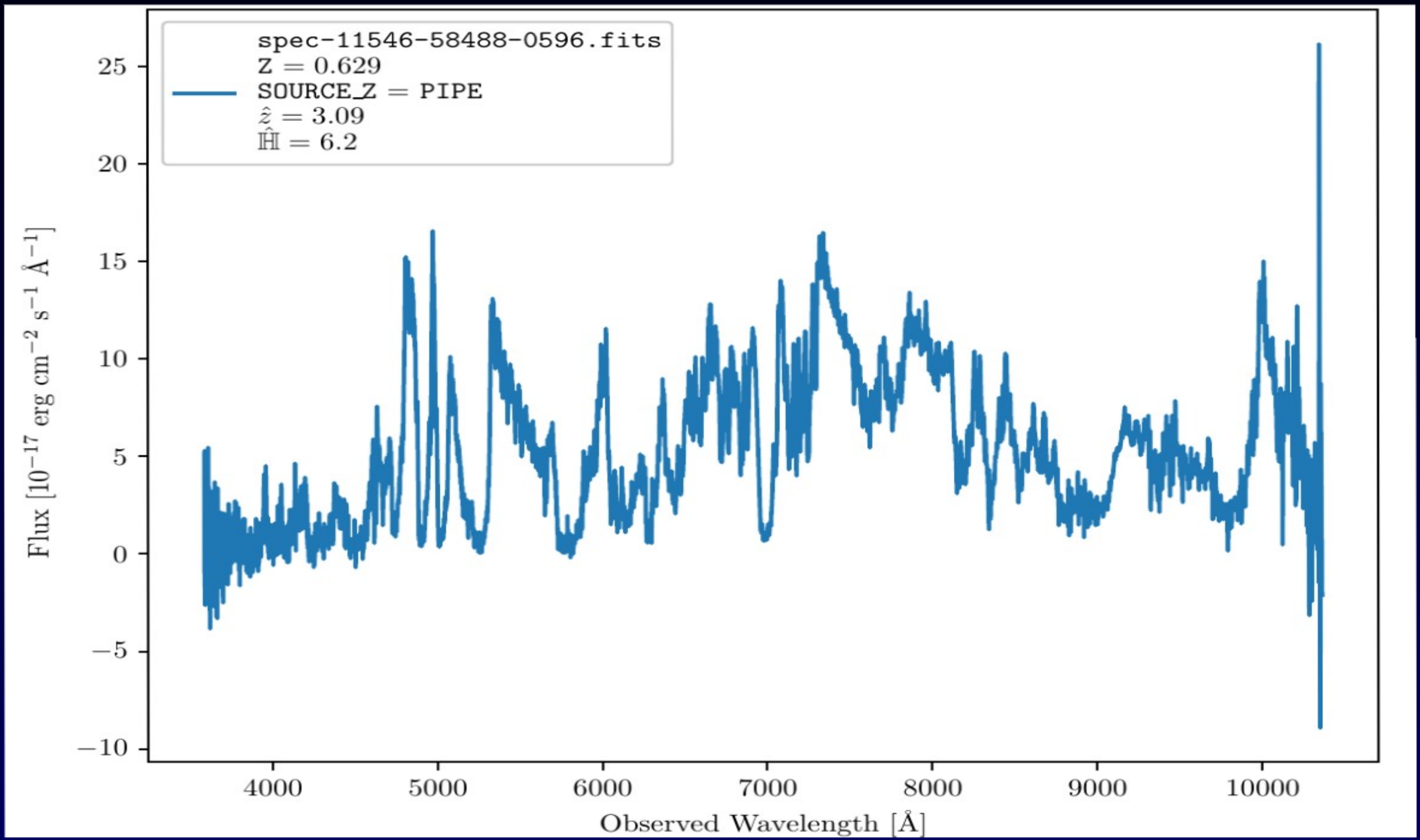


Hints for human decision



20 runs 7 times z=2.7 three times z=0.08 other 10 only once each

Highest entropy – wrong



BNN corrects the SDSS pipeline

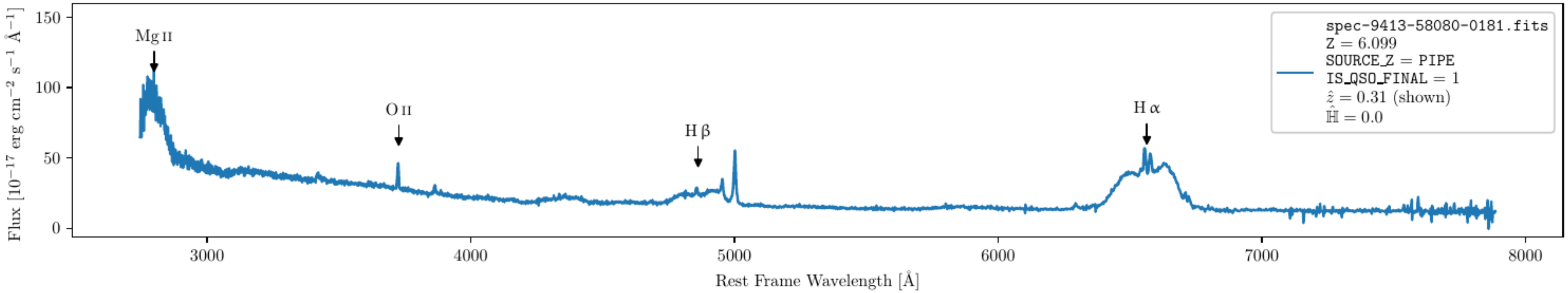


Figure B6. Spectrum with incorrectly high redshift prediction by the pipeline. The Bayesian CNN correctly predicted $\hat{z} = 0.31$ with $\hat{\mathbb{H}} = 0$.

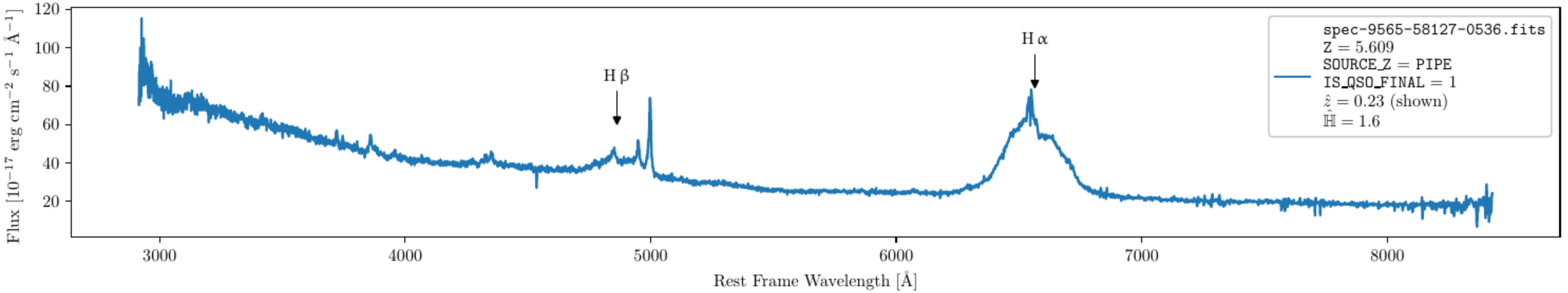
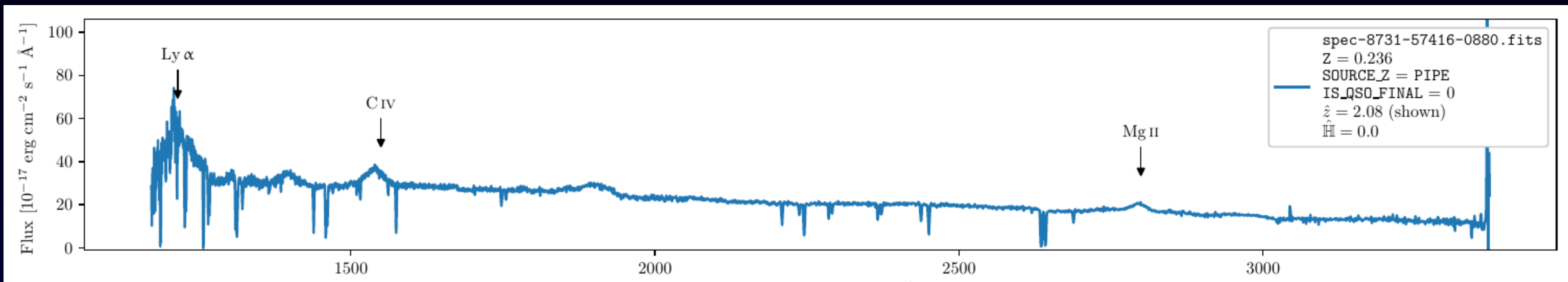
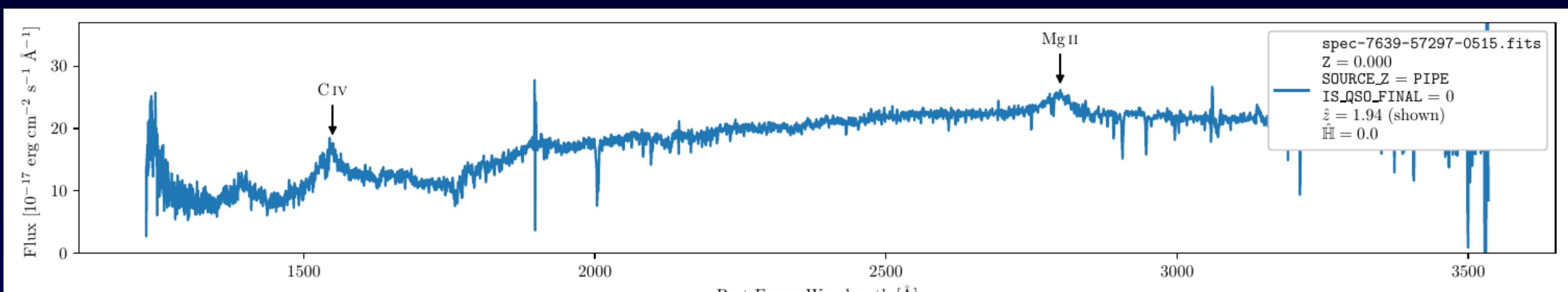
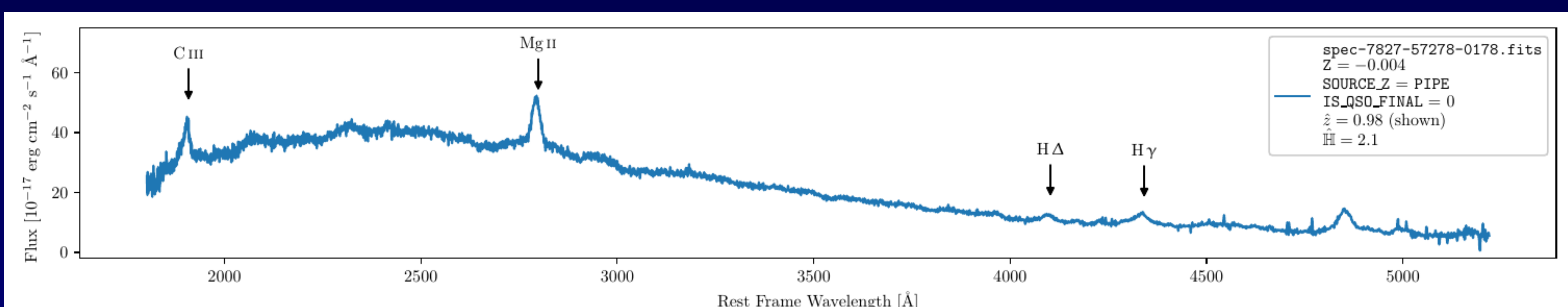


Figure B7. Spectrum with incorrectly high redshift prediction by the pipeline. The Bayesian CNN correctly predicted $\hat{z} = 0.23$ with $\hat{\mathbb{H}} = 1.6$.

QSOs missing due to SDSS pipeline error







Bayesian deep network errors

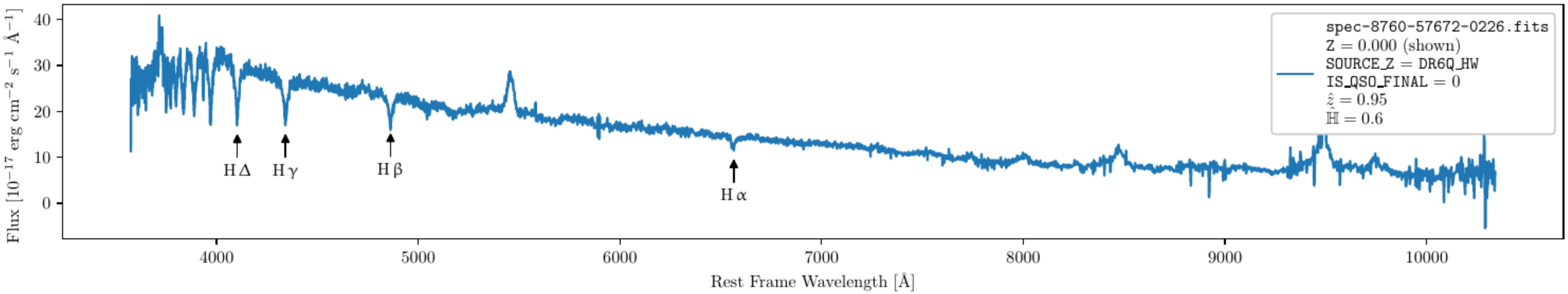


Figure B16. Error of the Bayesian CNN that does not recognise a star (primary Z = 0 is the true redshift) probably because of the emission lines.

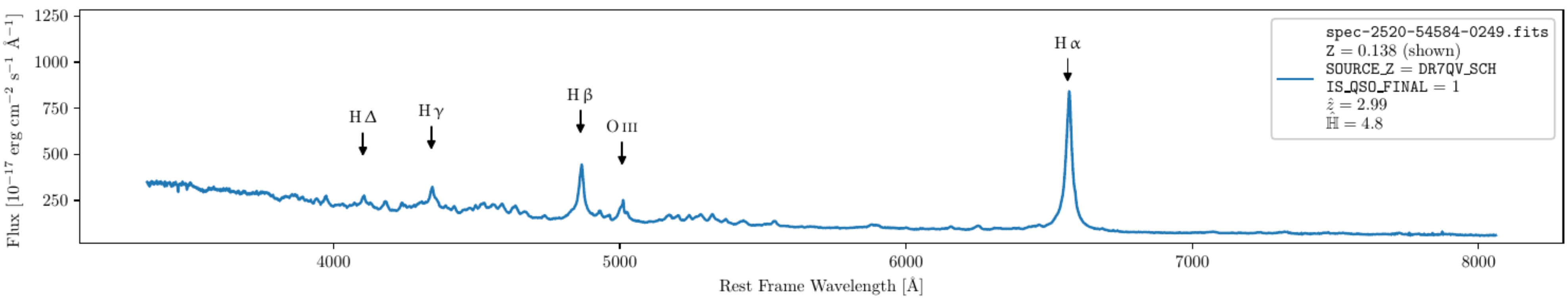


Figure B17. Error of the Bayesian CNN that probably misidentified a spectral line. However, the predictive entropy is high ($\hat{\mathbb{H}} = 4.8$).

Conclusions

- Bayesian deep learning is a relatively new method, it has just entered the astronomy as well
- Bayesian deep learning is a good way to get uncertainty
- Predictive entropy may identify wrong predictions or strange cases - hand it to expert for verification
- There is no simple threshold to decide !
- Can augment the decision of other pipelines (e.g. template based)
- Combination with Active learning - promissing future