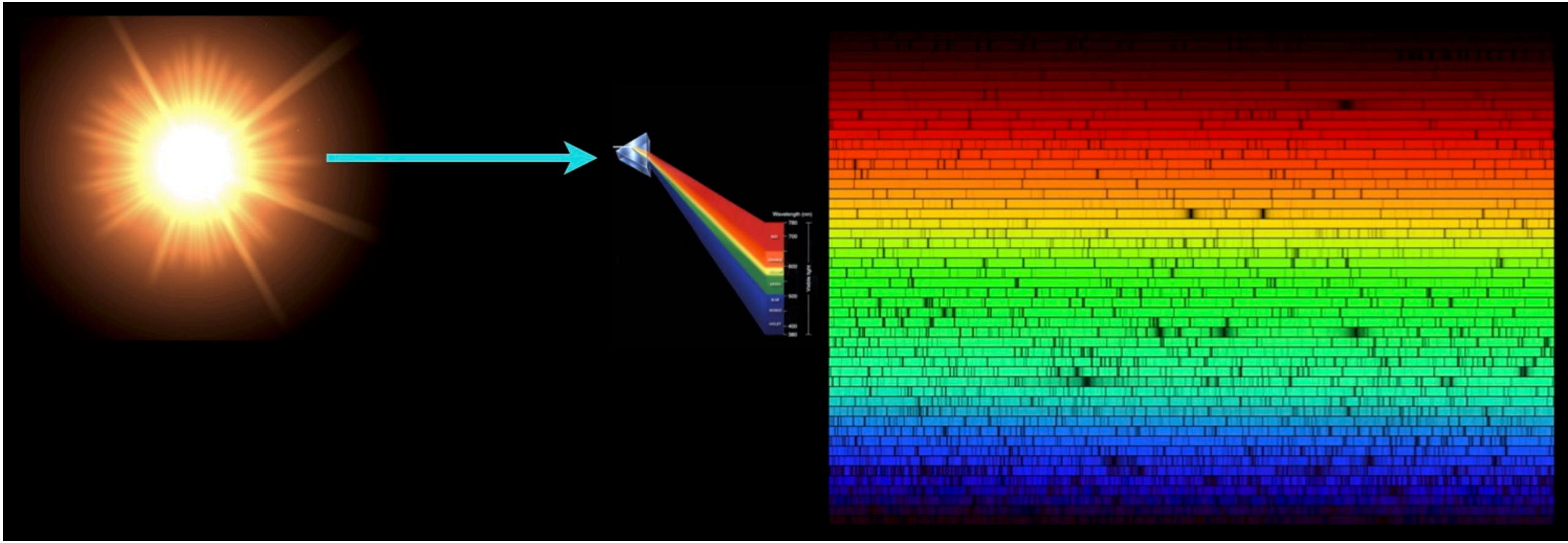
Stellar Classification with Neural Networks

Zlatko Saldic & Alexander Rathcke

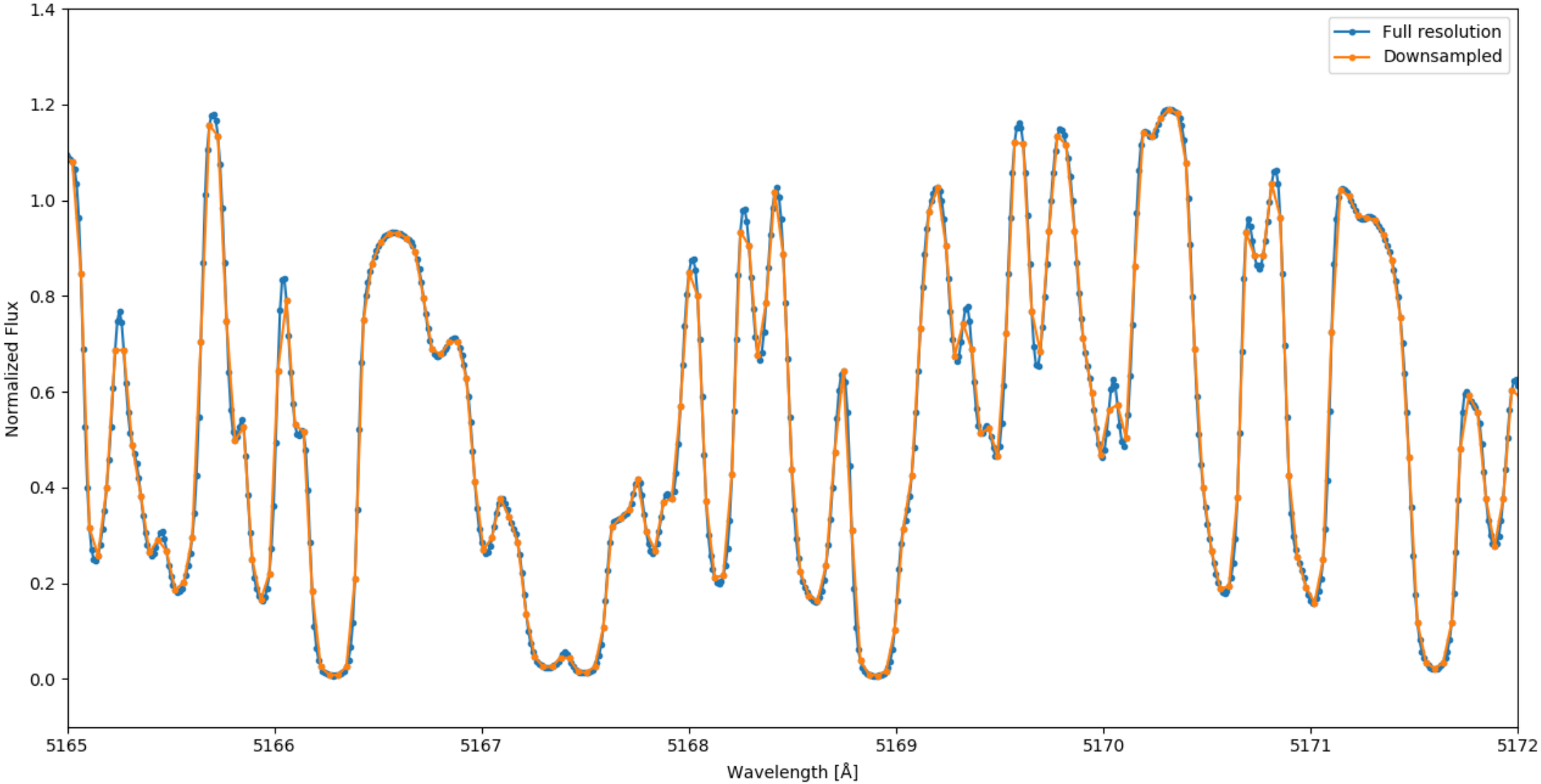
(All group members have contributed evenly to the project)


Wavelength (nm)
780
700
600
500
400
380
Visible light

Data:

- Model grid consisting of ~50k synthetic stellar spectra
- ~30k pixels covering a spectral range of 300 Å (very high resolution)
- 4 target parameters (labeled)

- Preprocessing:
  - Normalizing spectrum and labels
  - Spectrum downsampling (X.shape: (51359, 29788) -> X.shape(51359, 7447))


Full resolution
Downsampled
Normalized Flux
Wavelength [Å]
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
5165
5166
5167
5168
5169
5170
5171
5172

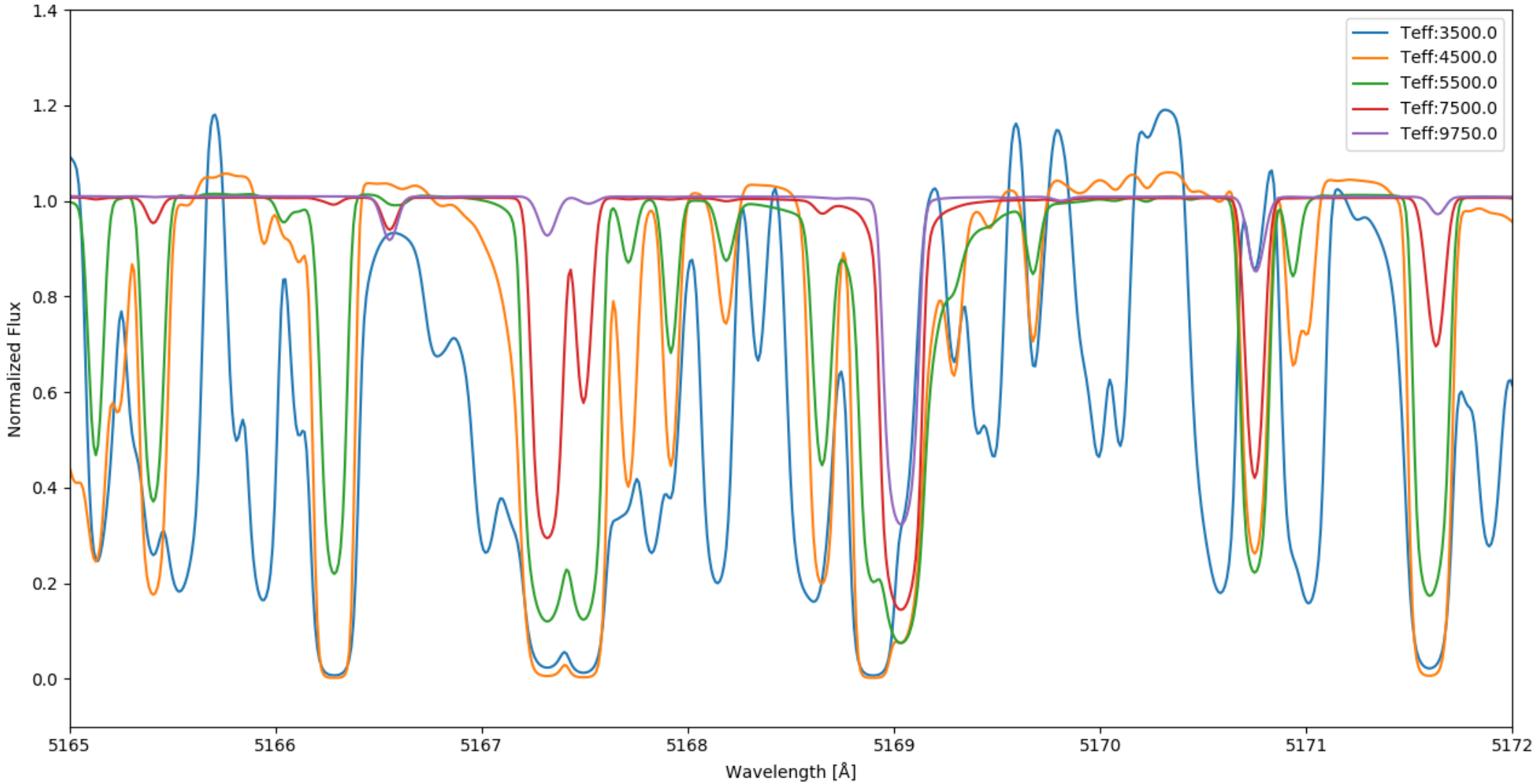
Teff 3500 to 9750 by 250 (26)

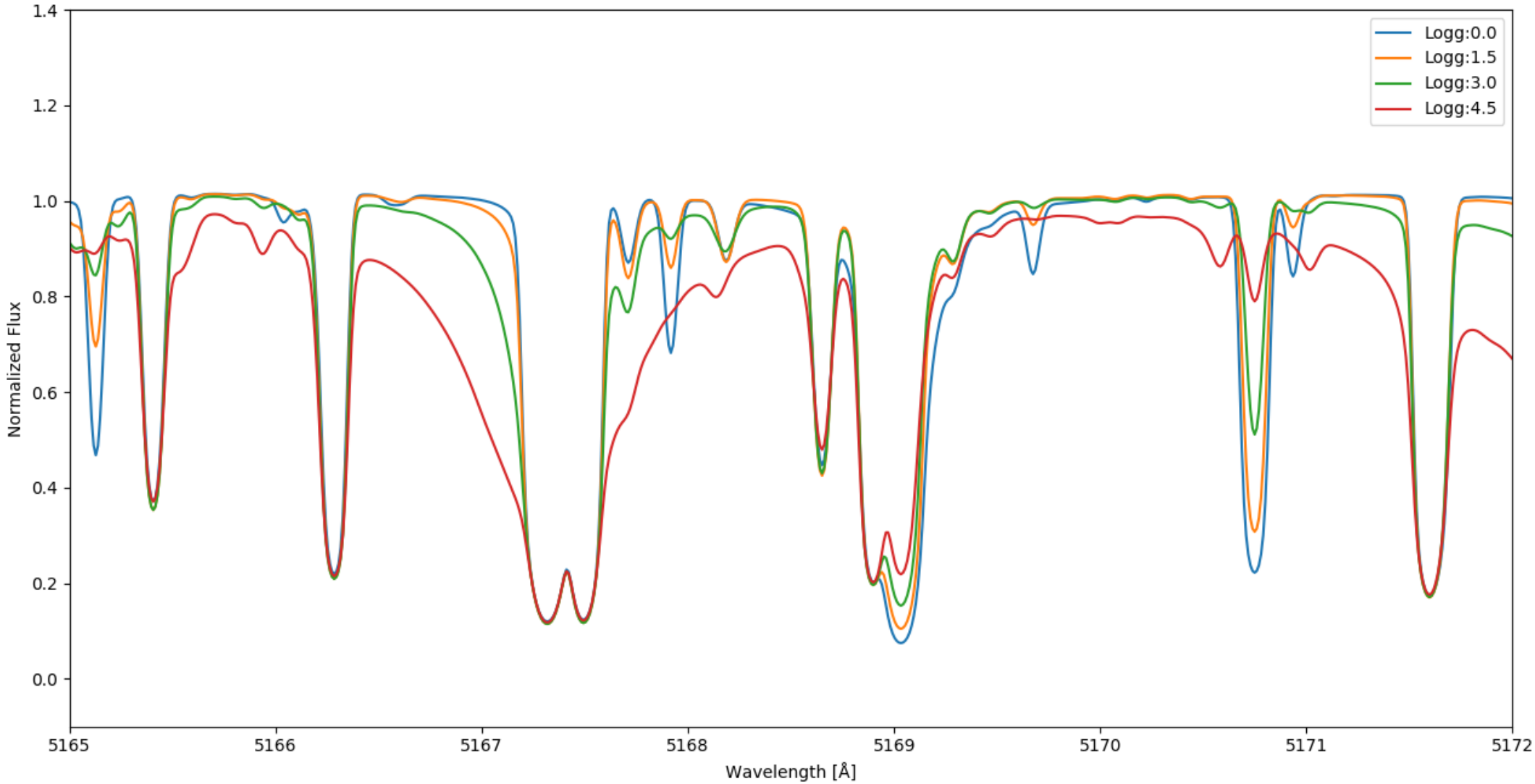
log g 0.0 to 5.0 by 0.5 (11)
not all values at every Teff

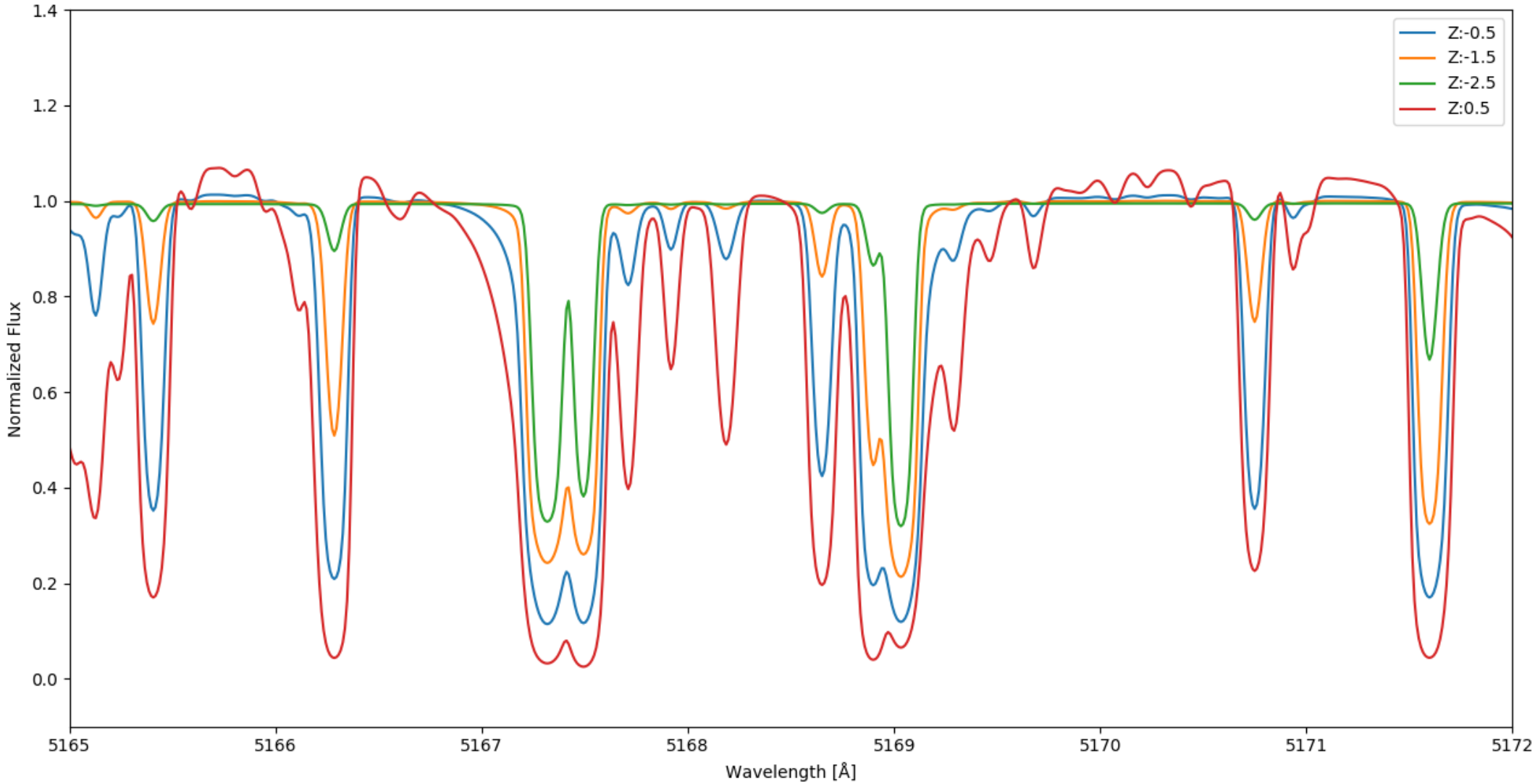
[m/H] -2.5 to +0.5 by 0.5 (7)

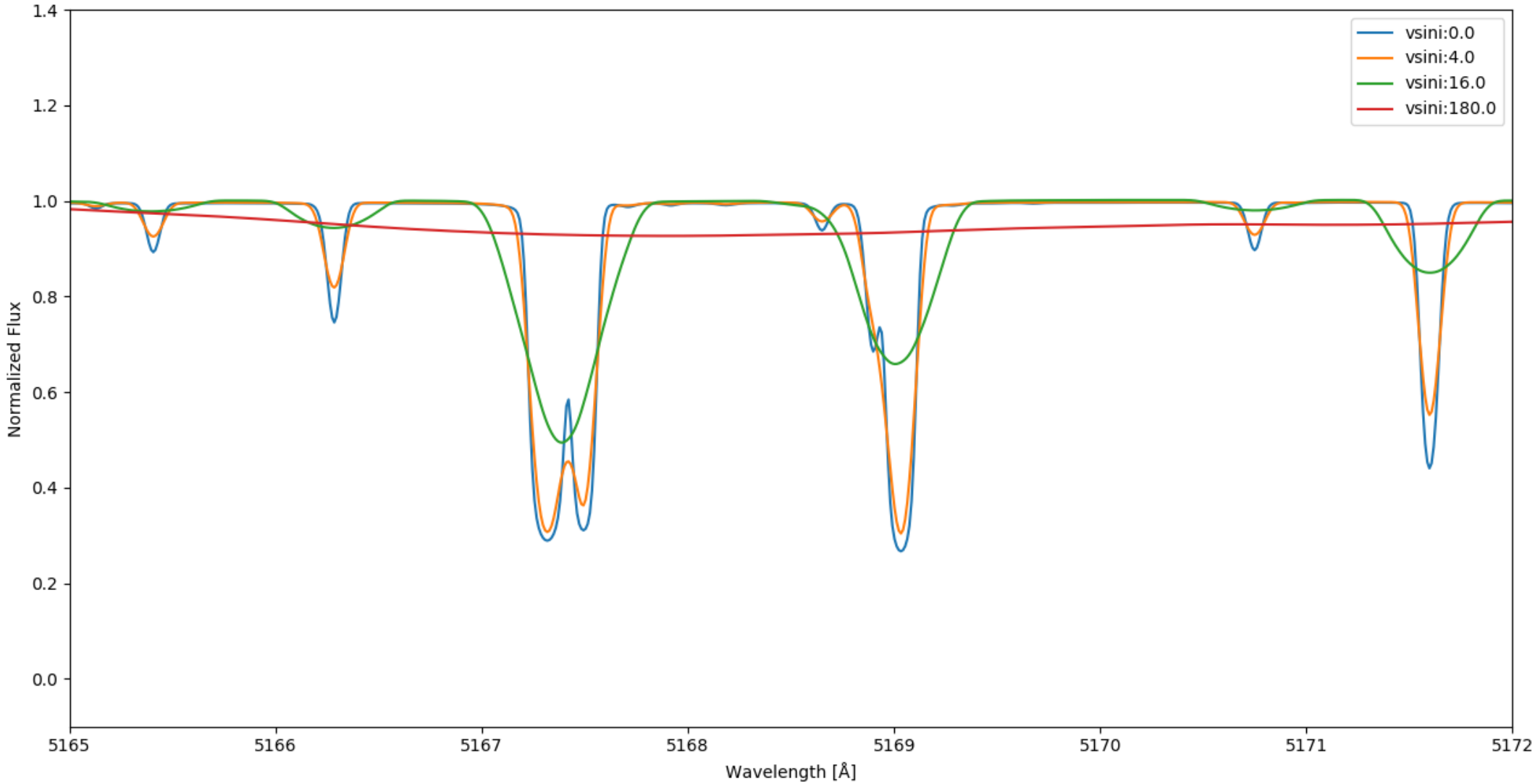
vsini 0 to 200 km/s (29)
0, 1, 2, 4, 6, 8, 10, 12, 16,
20, 25, 30, 35, 40, 45, 50,
55, 60, 65, 70, 80, 90, 100,
110, 120, 140, 160, 180, 200

	0.0	0.5	1.0	1.5	2.0	2.5	3.0	3.5	4.0	4.5	5.0	No.	Total spectra
3500	x	x	x	x	x	x	x	x	x	x	x	11	2,233
3750	x	x	x	x	x	x	x	x	x	x	x	11	2,233
4000	x	x	x	x	x	x	x	x	x	x	x	11	2,233
4250	x	x	x	x	x	x	x	x	x	x	x	11	2,233
4500	x	x	x	x	x	x	x	x	x	x	x	11	2,233
4750	x	x	x	x	x	x	x	x	x	x	x	11	2,233
5000	x	x	x	x	x	x	x	x	x	x	x	11	2,233
5250	x	x	x	x	x	x	x	x	x	x	x	11	2,233
5500	x	x	x	x	x	x	x	x	x	x	x	11	2,233
5750	x	x	x	x	x	x	x	x	x	x	x	11	2,233
6000	x	x	x	x	x	x	x	x	x	x	x	11	2,233
6250		x	x	x	x	x	x	x	x	x	x	10	2,030
6500		x	x	x	x	x	x	x	x	x	x	10	2,030
6750		x	x	x	x	x	x	x	x	x	x	10	2,030
7000		x	x	x	x	x	x	x	x	x	x	10	2,030
7250		x	x	x	x	x	x	x	x	x	x	10	2,030
7500		x	x	x	x	x	x	x	x	x	x	10	2,030
7750			x	x	x	x	x	x	x	x	x	9	1,827
8000			x	x	x	x	x	x	x	x	x	9	1,827
8250			x	x	x	x	x	x	x	x	x	9	1,827
8500				x	x	x	x	x	x	x	x	8	1,624
8750				x	x	x	x	x	x	x	x	8	1,624
9000				x	x	x	x	x	x	x	x	8	1,624
9250					x	x	x	x	x	x	x	7	1,421
9500					x	x	x	x	x	x	x	7	1,421
9750					x	x	x	x	x	x	x	7	1,421
												253	51,359


Teff:3500.0
Teff:4500.0
Teff:5500.0
Teff:7500.0
Teff:9750.0
Normalized Flux
Wavelength [Å]
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
5165
5166
5167
5168
5169
5170
5171
5172


Logg:0.0
Logg:1.5
Logg:3.0
Logg:4.5
Normalized Flux
Wavelength [Å]
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
5165
5166
5167
5168
5169
5170
5171
5172


Z:-0.5
Z:-1.5
Z:-2.5
Z:0.5
Normalized Flux
Wavelength [Å]
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
5165
5166
5167
5168
5169
5170
5171
5172


vsini:0.0
vsini:4.0
vsini:16.0
vsini:180.0
Normalized Flux
Wavelength [Å]
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
5165
5166
5167
5168
5169
5170
5171
5172

Approaching the problem & building the network

Similar problems exists

GOOD AUDIENCE + SIGNUP FOR OUR NEWSLETTER ✍ SUBMIT A STORY | 🚀 BINANCE CHAIN X RAVEN

Introduction to 1D Convolutional Neural Networks in Keras for Time Sequences

Nils Ackermann Follow

Sep 4, 2018 · 7 min read

Introduction

Many articles focus on two dimensional conv

When to Apply a 1D CNN?

A CNN works well for identifying simple patterns within your data which will then be used to form more complex patterns within higher layers. A 1D CNN is very effective when you expect to derive interesting features from shorter (fixed-length) segments of the overall data set and where the location of the feature within the segment is not of high relevance.

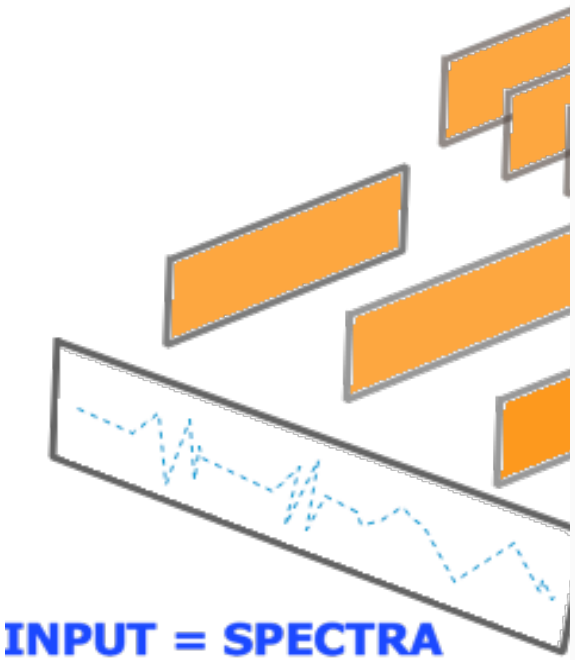
Initial structure for final network adopted from:
Fabbro, Sebastien, et al. (2017)

The Neural

FULLY CONNECT LAYERS

OUTPUT = TEMPERATURE

MAX POOLING



INPUT = SPECTRA

Layer (type)	Output Shape	Param #
Input (InputLayer)	(None, 7447)	0
reshape_1 (Reshape)	(None, 7447, 1)	0
conv1d_1 (Conv1D)	(None, 7447, 4)	132
conv1d_2 (Conv1D)	(None, 7447, 16)	2064
max_pooling1d_1 (MaxPooling1	(None, 1861, 16)	0
flatten_1 (Flatten)	(None, 29776)	0
dense_1 (Dense)	(None, 256)	7622912
dense_2 (Dense)	(None, 128)	32896
Output (Dense)	(None, 4)	516

Total params: 7,658,520
Trainable params: 7,658,520
Non-trainable params: 0

Some nice KERAS features

```
38016/38519 [============================>.] - ETA: 3s - loss: 0.0013 - mean_absolute_error: 0.0242
38048/38519 [============================>.] - ETA: 3s - loss: 0.0013 - mean_absolute_error: 0.0242
38080/38519 [============================>.] - ETA: 3s - loss: 0.0013 - mean_absolute_error: 0.0242
38112/38519 [============================>.] - ETA: 2s - loss: 0.0013 - mean_absolute_error: 0.0242
38144/38519 [============================>.] - ETA: 2s - loss: 0.0013 - mean_absolute_error: 0.0242
38176/38519 [============================>.] - ETA: 2s - loss: 0.0013 - mean_absolute_error: 0.0242
38208/38519 [============================>.] - ETA: 2s - loss: 0.0013 - mean_absolute_error: 0.0242
38240/38519 [============================>.] - ETA: 1s - loss: 0.0013 - mean_absolute_error: 0.0242
38272/38519 [============================>.] - ETA: 1s - loss: 0.0013 - mean_absolute_error: 0.0242
38304/38519 [============================>.] - ETA: 1s - loss: 0.0013 - mean_absolute_error: 0.0242
38336/38519 [============================>.] - ETA: 1s - loss: 0.0013 - mean_absolute_error: 0.0242
38368/38519 [============================>.] - ETA: 1s - loss: 0.0013 - mean_absolute_error: 0.0242
38400/38519 [============================>.] - ETA: 0s - loss: 0.0013 - mean_absolute_error: 0.0242
38432/38519 [============================>.] - ETA: 0s - loss: 0.0013 - mean_absolute_error: 0.0242
38464/38519 [============================>.] - ETA: 0s - loss: 0.0013 - mean_absolute_error: 0.0242
38496/38519 [============================>.] - ETA: 0s - loss: 0.0013 - mean_absolute_error: 0.0242
38519/38519 [==============================] - 287s 7ms/step - loss: 0.0013 - mean_absolute_error: 0.0242 -
val_loss: 0.0014 - val_mean_absolute_error: 0.0245
Epoch 00023: early stopping
trained_data.h5 saved.

Process finished with exit code 0
```

```
'min')
```

Hyper parameters

- Kernel size (filter length that convolves across spectrum) = 32 pixels

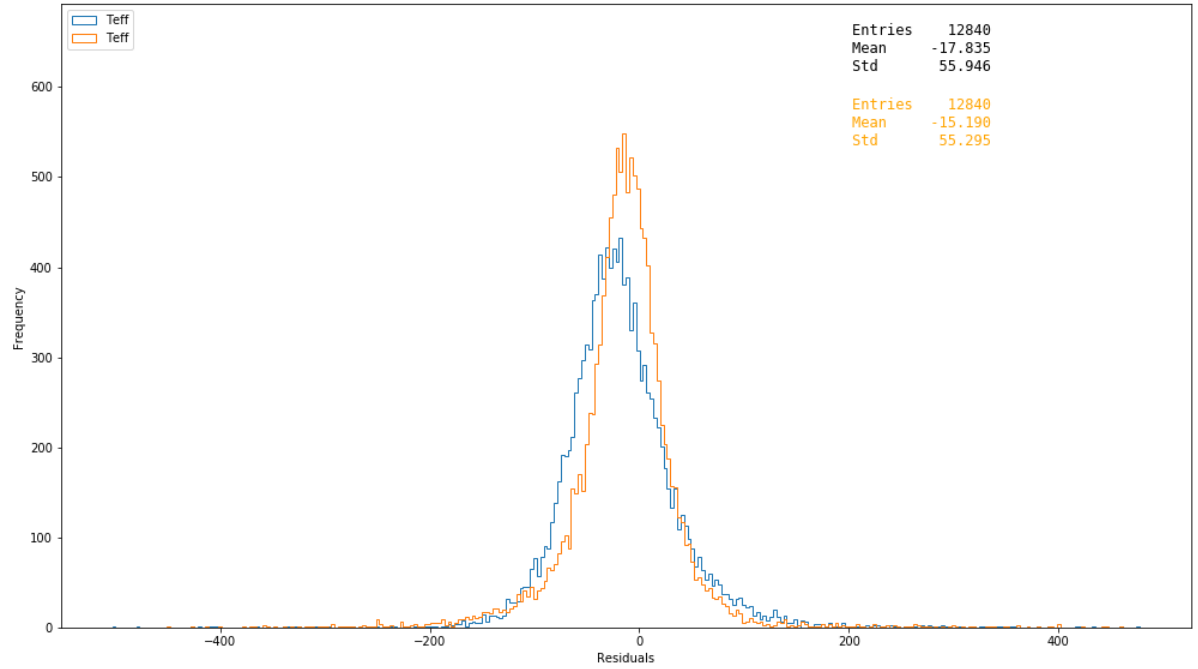
  w/ 32 pixels we where sure about detecting important features in the spectrum

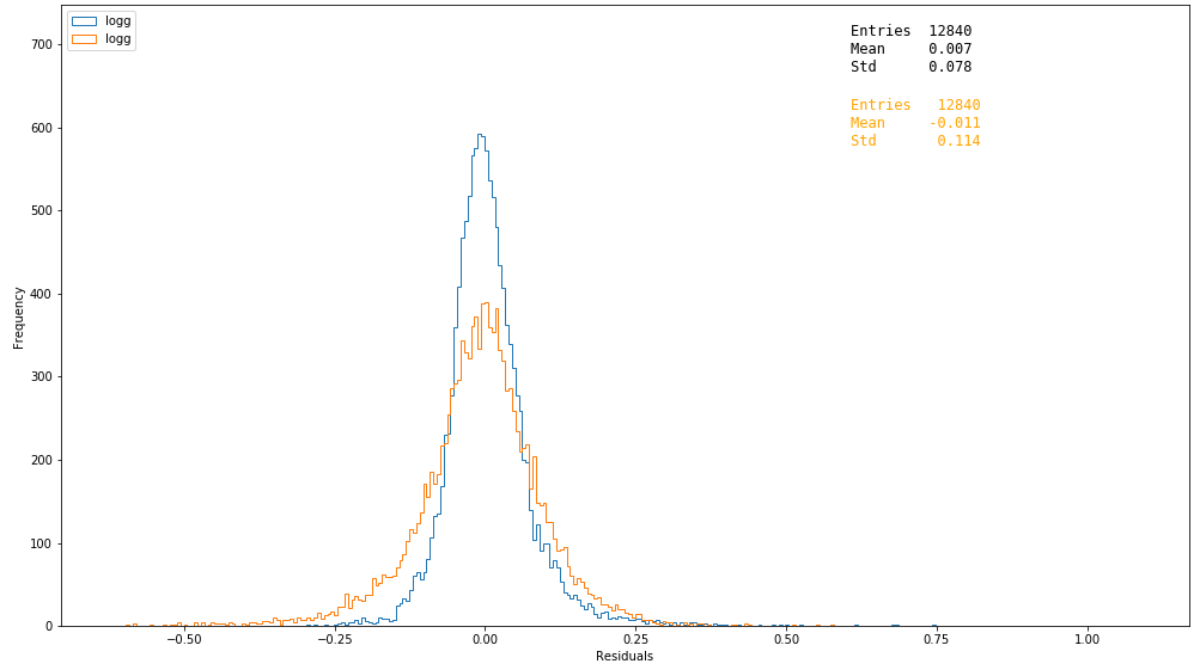
- Activation = ReLu
- Gradient Descent Optimizer = ADAM

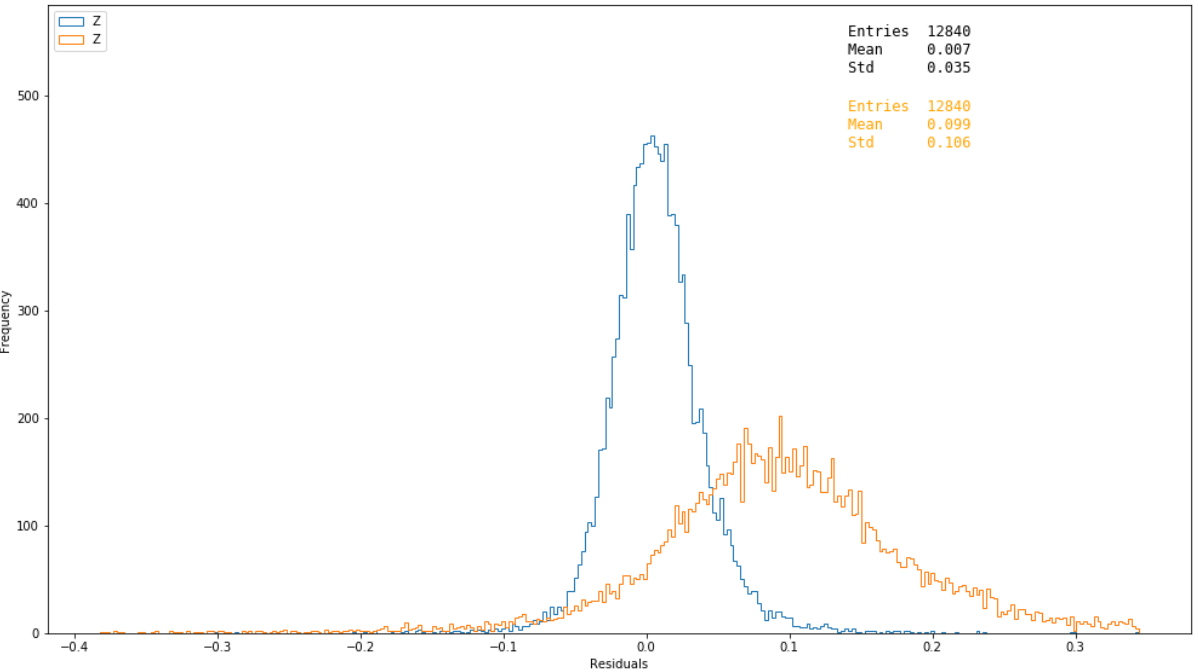
- Maxpool window length (the number of inputs which are compared against each other to find the max value) = 4
- Learning rate (of the gradient descent optimizer) = 0.0007
- Batch size (number of spectra fed into the model at once) = 64

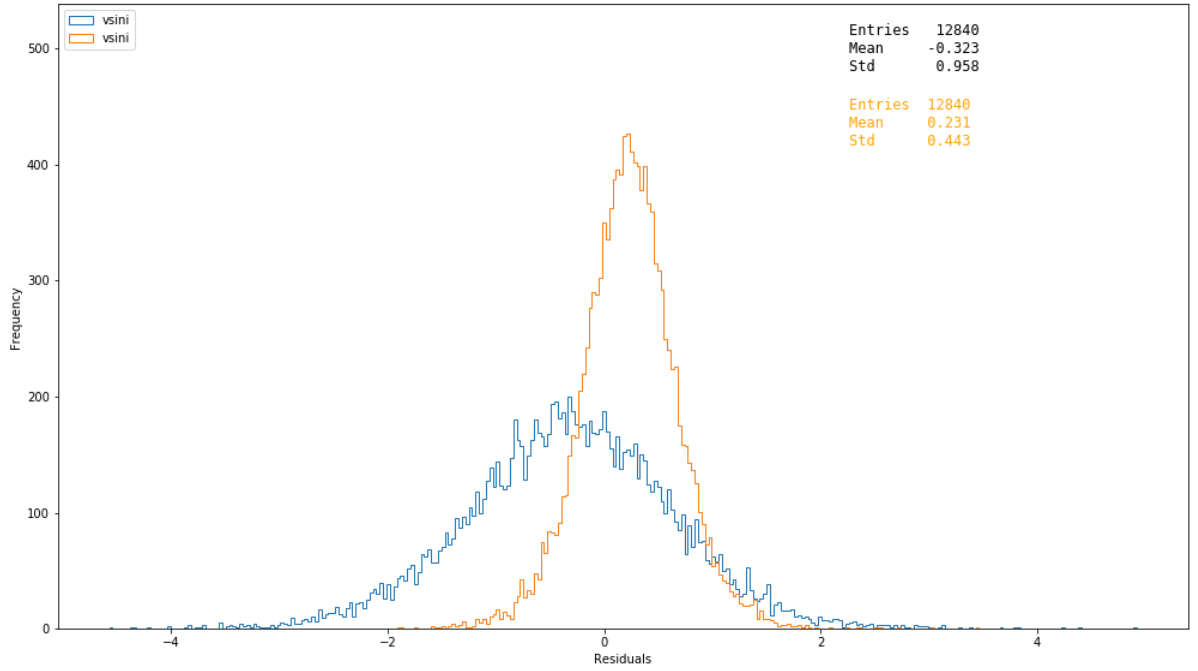
- Loss function = Mean Squared Error

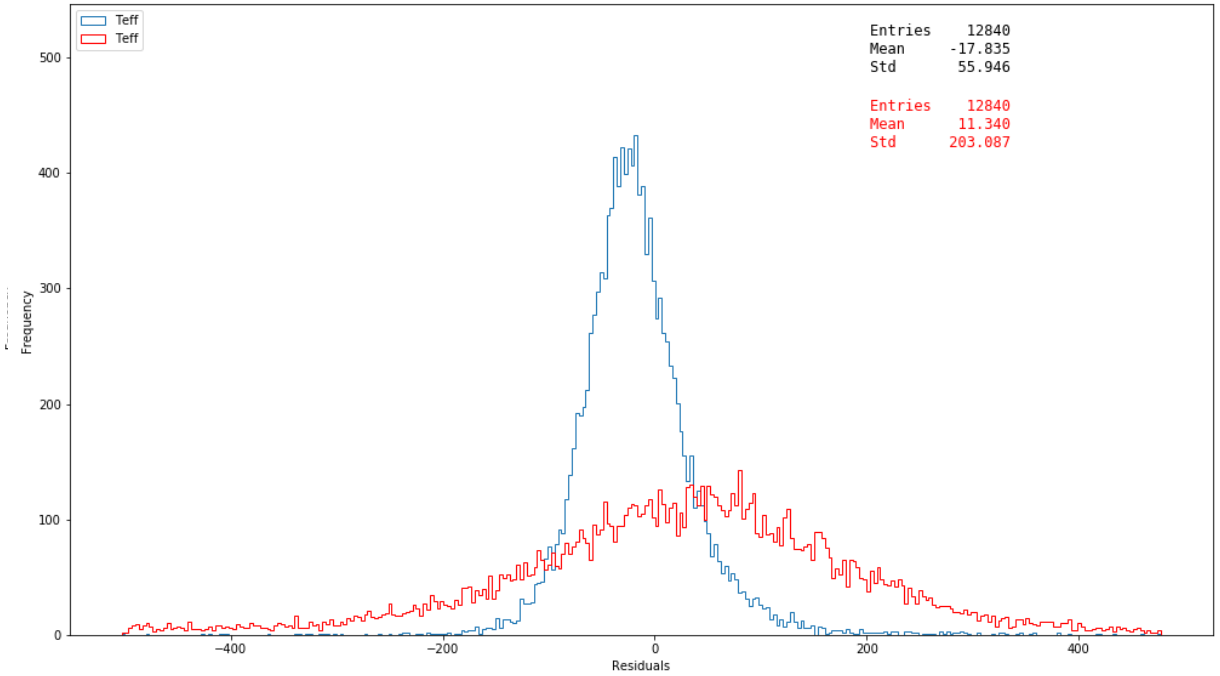
Results

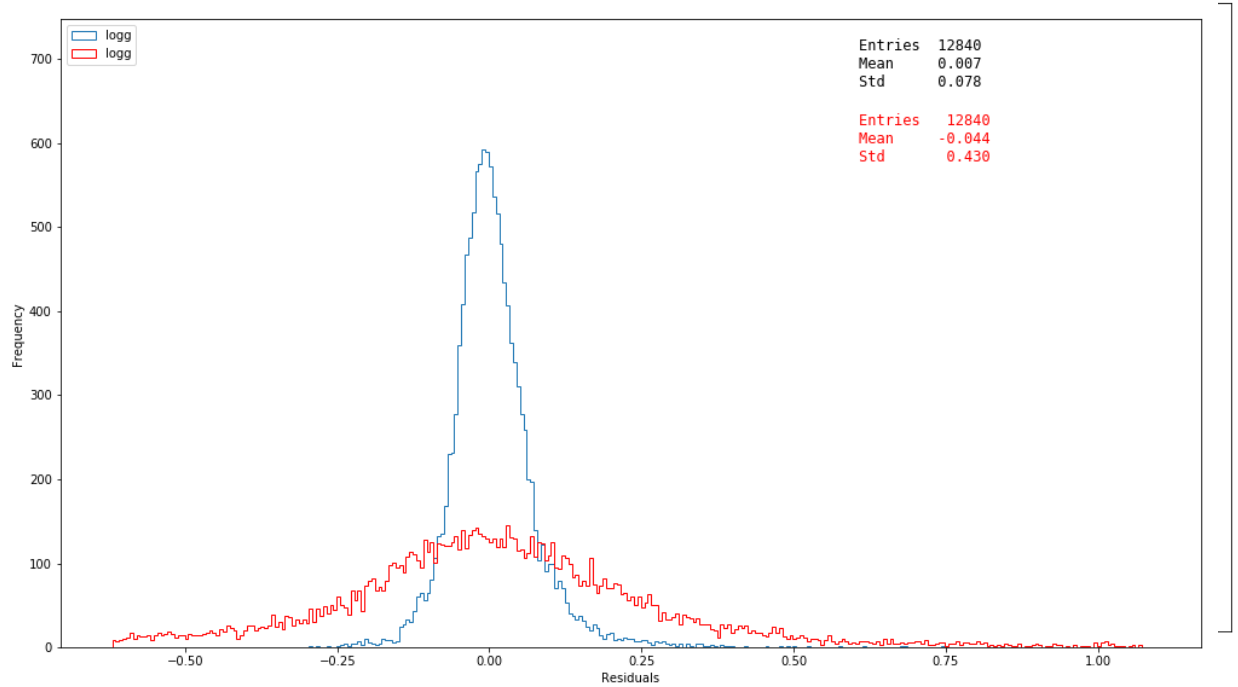

Teff
Teff
Entries 12840
Mean -17.835
Std 55.946
Entries 12840
Mean -15.190
Std 55.295
Frequency
Residuals

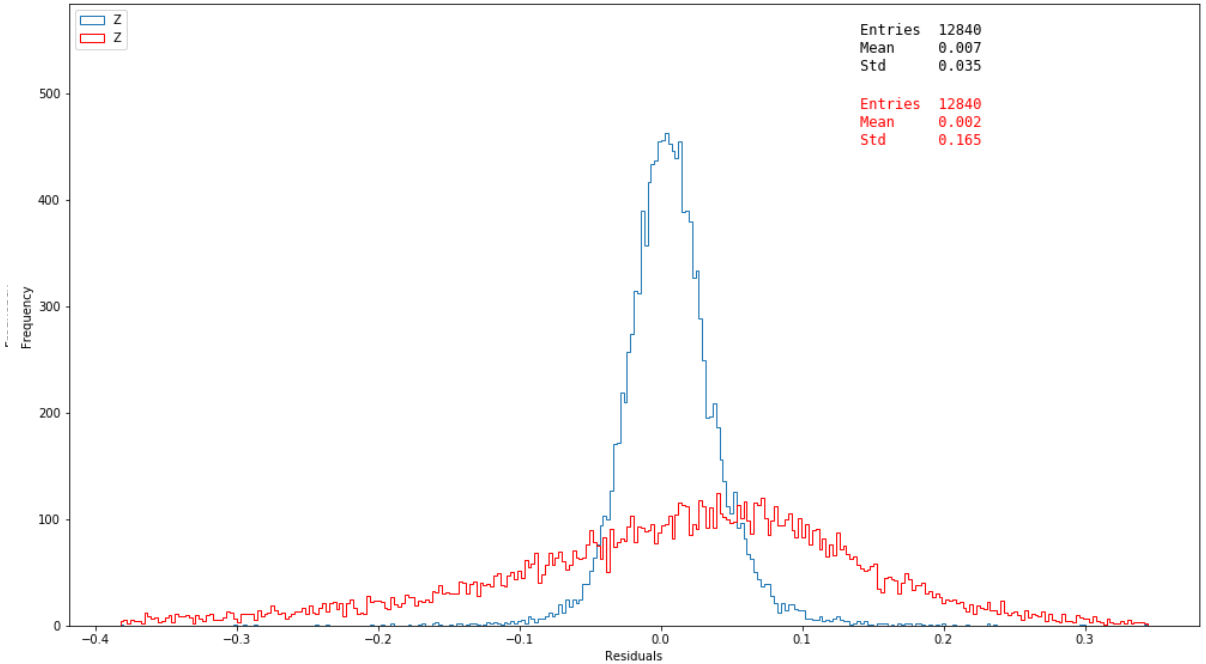

logg
logg
Entries 12840
Mean 0.007
Std 0.078
Entries 12840
Mean -0.011
Std 0.114
Frequency
Residuals

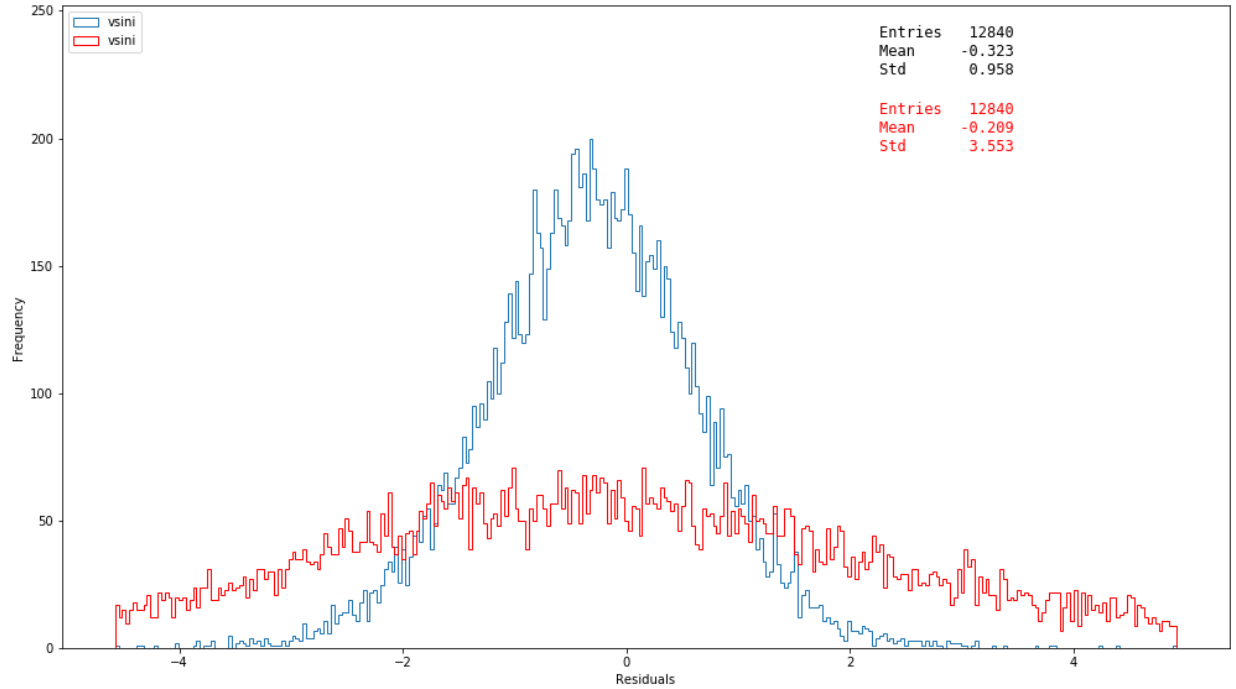

Z
Z
Entries 12840
Mean 0.007
Std 0.035
Entries 12840
Mean 0.099
Std 0.106
Frequency
Residuals


vsini
vsini
Entries 12840
Mean -0.323
Std 0.958
Entries 12840
Mean 0.231
Std 0.443
Frequency
Residuals

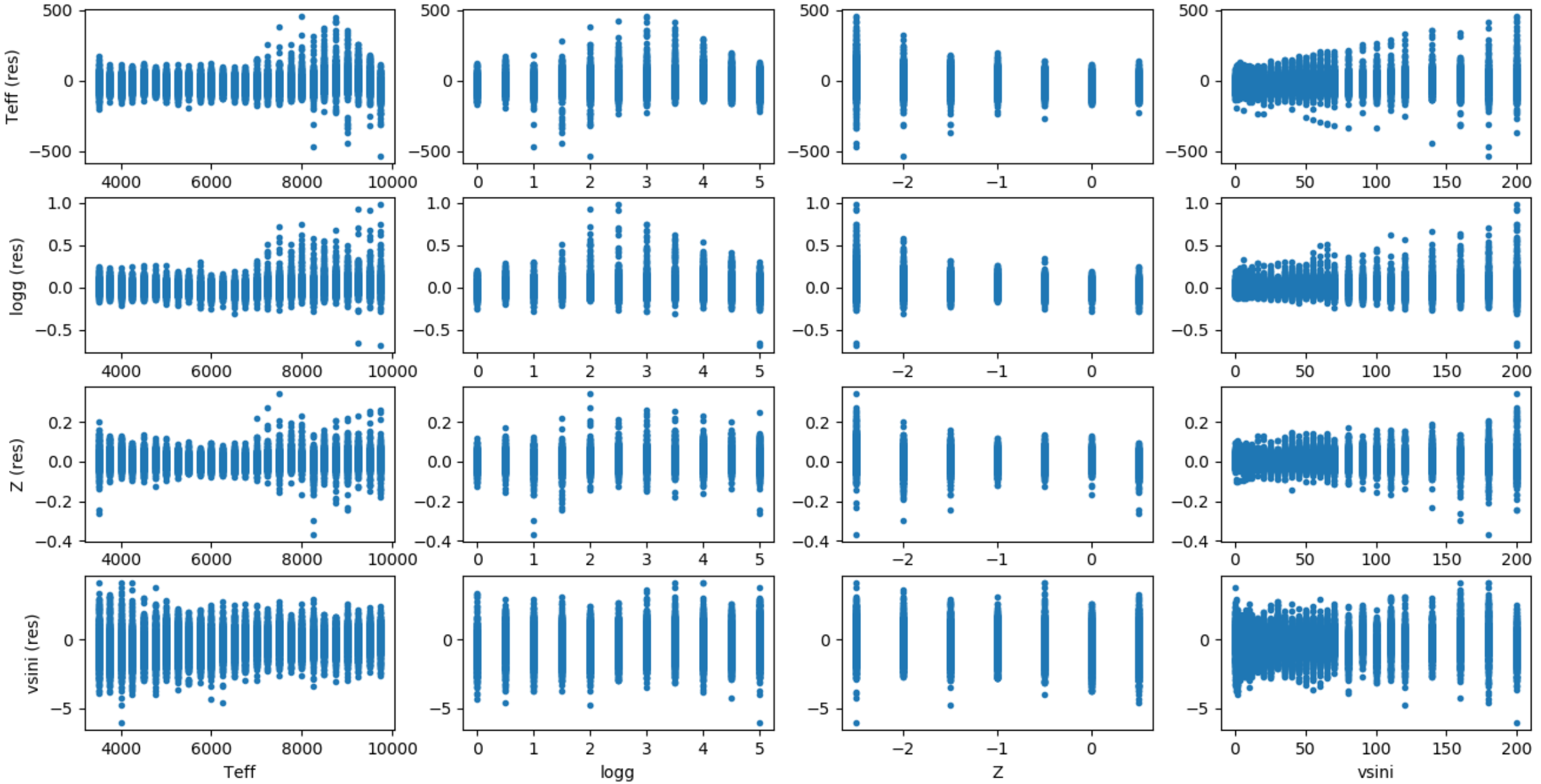

Teff
Teff
Entries 12840
Mean -17.835
Std 55.946
Entries 12840
Mean 11.340
Std 203.087
Frequency
Residuals


logg
logg
Entries 12840
Mean 0.007
Std 0.078
Entries 12840
Mean -0.044
Std 0.430
Frequency
Residuals


Z
Z
Entries 12840
Mean 0.007
Std 0.035
Entries 12840
Mean 0.002
Std 0.165
Frequency
Residuals


vsini
vsini
Entries 12840
Mean -0.323
Std 0.958
Entries 12840
Mean -0.209
Std 3.553
Frequency
Residuals

For multivariate regression approach:



Perspective & Future Work

- Make applicable to real observations
- Hyperparameter optimization using GPU cluster
- Expand to estimate alpha particle enrichment (or even indidivual abundances)
- Error estimation (Bayesian Neural Networks?)


Questions?