Draft version April 9, 2024
Typeset using LaTeX **twocolumn** style in AASTeX63

Astronomy 128: Astro Data Science Lab
Lab 2: Modeling Stellar Spectra

Hien Nguyen[1, 2]

[1] *Department of Astronomy, University of California, Berkeley, CA 94720-3411, USA*
[2] *Department of Physics, University of California, Berkeley, CA 94720-7300, USA*

ABSTRACT

Explored in this lab is a *data-driven* prediction model for stellar spectra known as *The Cannon*. Following the steps outlined by Ness et al. (2015), a generative model is constructed and used with reference spectra and stellar properties from the APOGEE survey and ASPCAP pipeline, respectively, to predict normalized spectra and infer stellar properties. The model is able to reproduce ASPCAP stellar properties with values that are within the reported range. The predicted spectra exhibit features are in moderate agreement with APOGEE spectra features, with non-negligible discrepencies.

Keywords: stellar spectra, stellar properties, continuum normalization

1. INTRODUCTION

Stellar spectroscopy is the study of the properties of a star through its spectrum: the observed intensity or flux as a function of wavelength or frequency. To the zeroth order, stellar spectra can be approximated with Planck's function

$$B_\lambda\left(T\right) = \frac{2hc^2}{\lambda^5}\frac{1}{e^{hc/\lambda kT}-1}\left[\frac{\mathrm{erg}}{\mathrm{s\,Å\,str\,cm^2}}\right]$$

also known as a *blackbody* curve. However, stars are not perfect blackbodies and their spectra exhibit features called emission and absorption lines that are caused by interactions between photons and different species of elements that exist within the stars. There are additional physical effects that contribute to the deviation of stellar spectra away from a perfect blackbody. Modeling of stellar spectra can thus be an involved and complex process.

This lab's purpose is to construct a generative prediction model for stellar spectra based on a set of stellar properties, referred to as *stellar labels* or simply *labels*. These properties include, but are not limited to, temperature, surface gravity, and elemental abundances. The model's primary objective is to fit stellar spectra and infer the properties of stars. To this end, the process utilizes Red Giants spectra from the Apache Point Observatory Galactic Evolution Experiment (APOGEE) survey (Majewski et al. (2017)). The spectra model is *data-driven*, meaning it starts with a *training set* of spectra with known labels and analyze them to determine how the spectrum varies with each label.

Construction of the spectra model primarily references Ness et al. (2015) and uses *The Cannon*, a data-driven approach for determining stellar labels from spectroscopic data. *The Cannon* learns how the continuum-normalized spectra depend on stellar labels from the high fidelity *known* labels derived through the Abundances, Stellar Parameters, And Spectra (ASPCAP) pipeline (Holtzman et al. (2015)), of *reference stars* by fitting a flexible model at each wavelength pixel. Then it uses the model to derive labels for the *survey stars*.

Section 2 will go over the download of APOGEE spectra and the corresponding ASPCAP lalbels, data processing of the spectra, the construction of *The Cannon* model, and some procedures for applications of the *The Cannon* to infer stellar spectrum and properties. Section 3 will present and discuss the results of the spectra model, followed by a conclusion in section 4.

2. METHODOLOGY

2.1. *APOGEE Spectra*

The APOGEE spectra used are provided by the Sloan Digital Sky Survey (SDSS), specifically Data Release 16 (DR16). The SDSS server sorts the APOGEE spectra by digit string, or *field*, that identifies the stars location in the sky. There are four randomly chosen fields used: "M15", "N6791", "K2_C4_168-21", and "060+00". The SDSS website provides a bulk data download tutorial. I searched the DR16 archive for the four randomly chosen fields and downloaded the *apStar* files from each field using the following `rsync` commands in terminal

```
rsync -aLvz -include "apStar-*.fits"
-exclude "*" -prune-empty-dirs -progress
```

Corresponding author: Hien Nguyen
hien24199@berkeley.edu

Table 1. Contents of *apStar* files from García Pérez et al. (2016).

HDU	Contents
0	main header with target, RV, version information
1	resampled spectra
2	uncertainties in spectra
3	pixel mask for spectra

```
rsync://data.sdss.org/dr16/apogee/spectro/
    redux/r12/stars/apo25m/Field
```

with “Field” at the end of the `rsync` url replaced with one of the four field strings. Note that to use `rsync`, the `http` URL has to be converted into an `rsync` URL by

- Replacing `//` with `rsync://`
- Remove `/sas/` from the `http` URL

Additionally, the directory `apo25m` are for stars that are observed at the Apache Point Observatory in the North, while `lco25m` contains stars observed at Las Campanas Observatory in the South. More than 3036 spectra were downloaded using terminal and the above commands. The field “060+00” is listed as “060%2B00” in the URL of the SDSS archive, which is incorrect and will confuse `rsync` into not downloading any of the *apStar* files. The solution is simple: just modify the URL when running the `rsync` URL so that it reads “060+00”.

The *apStar* files follow the naming convention “`apStar-rYY-2MX+X.fits`”. `YY` denotes the reduction pipeline version used to process the spectrum (see Holtzman et al. (2015)). `2MX+X` references the Two Micron All-Sky Survey (2MASS), and it is the 2MASS ID for the star. Each *apStar* file contains either an individual visit spectrum or a coadded multi-visit spectra for one star. Multiple-visit spectra are the combined spectra of stars that were observed multiple times to increase the signal-to-noise (S/N) ratio of the data set. The coadded spectra are Doppler shifted to the barycentric frame. In this context, the barycentric frame is a frame of reference around the barycenter, or center-of-mass, of the solar system. Barycentric correction subtracts the Earth’s relative radial velocity from a observed star’s radial velocity. The Doppler shift correction is necessary because the Earth’s orbit around the Sun changes the observed radial velocity of a star. Note that a star’s radial velocity can also be complicated by it being in a dynamic system, such as a binary system.

The *apStar* files are opened using the `astropy.io.fits` package. The data of interests in each file are the first three from the header data unit (HDU): flux, uncertainty in the flux, and the bad pixel flags (see table 1). Information from the HDU show that the units for flux and the uncertainty in flux are recorded as 10^{-17}erg/s/cm^2/Å, which is just the amount of energy rate per area and wavelength interval. An example of multi-visit spectra is shown in figure 1.

For *apStar* files contain multi-visit spectra, the first spectrum, or the combined spectrum (see García Pérez et al. (2016)), is extracted:

```
hdu = fits.pen(file_path)
spectra = hdu[1].data
# multi-visit
if len(np.shape(spectra)[0]) > 1:
    spectrum = hdu[1].data[0]
    errors = hdu[2].data[0]
    masks = hdu[3].data[0]
# individual visit
else:
    spectrum = hdu[1].data
    errors = hdu[2].data
    masks = hdu[3].data
```

Wavelength values need to be reconstructed from pixel data (see figure 1). Inspection of HDUs show that the headers `CRVAL1` is the center of a pixel and `CRDELT1` is the step size of each pixel. Each wavelength bin is also in log-scale, so the reconstructed wavelength is

$$\lambda = 10^{\mathrm{CRVAL1}\,+\,i\,\cdot\,\mathrm{CRDELT1}} \tag{1}$$

where i is the index of a pixel in each spectrum.

Random spectra from each field with reconstructed wavelengths are shown in figure 2. To summarize, the APOGEE spectra span a wavelength range from $15,000 - 17,000$ Å in 8575 bins or pixels. Observe from figure 2 that the spectra are split into three wavelength ranges due to gaps in the instrument’s detectors, see section 3.4 in Majewski et al. (2017). From Ness et al. (2015), the approximate ranges of the three wavelength *chips* are: $151500 - 15800$ Å, $15890 - 16430$ Å, and $16490 - 16900$ Å.

2.2. *Identifying Bad Pixels with apStar Bitmasks*

Certain pixels in the *apStar* spectra are flagged as bad pixels. These flags are included in the HDUs as bitmask integer values (see table 1) that can be converted to binary representation, where each binary digit identifies a type of bad pixel. The bitmask HDU being used is the `APOGEE_PIXMASK`. There are a total of 17 digits, with 8 being important: 0-7, and 12; while the others can be ignored. Table 2 contains descriptions of the relevant bits. In order to minimize the effects of these bad pixels on the training of the spectra model, the flux uncertainty is set to a large value of 10^{10} whenever any of the bit digits 0-7 and/or 12 are set to `TRUE` in the binary representation of the bitmask integers. For more details, see section A in the appendix.

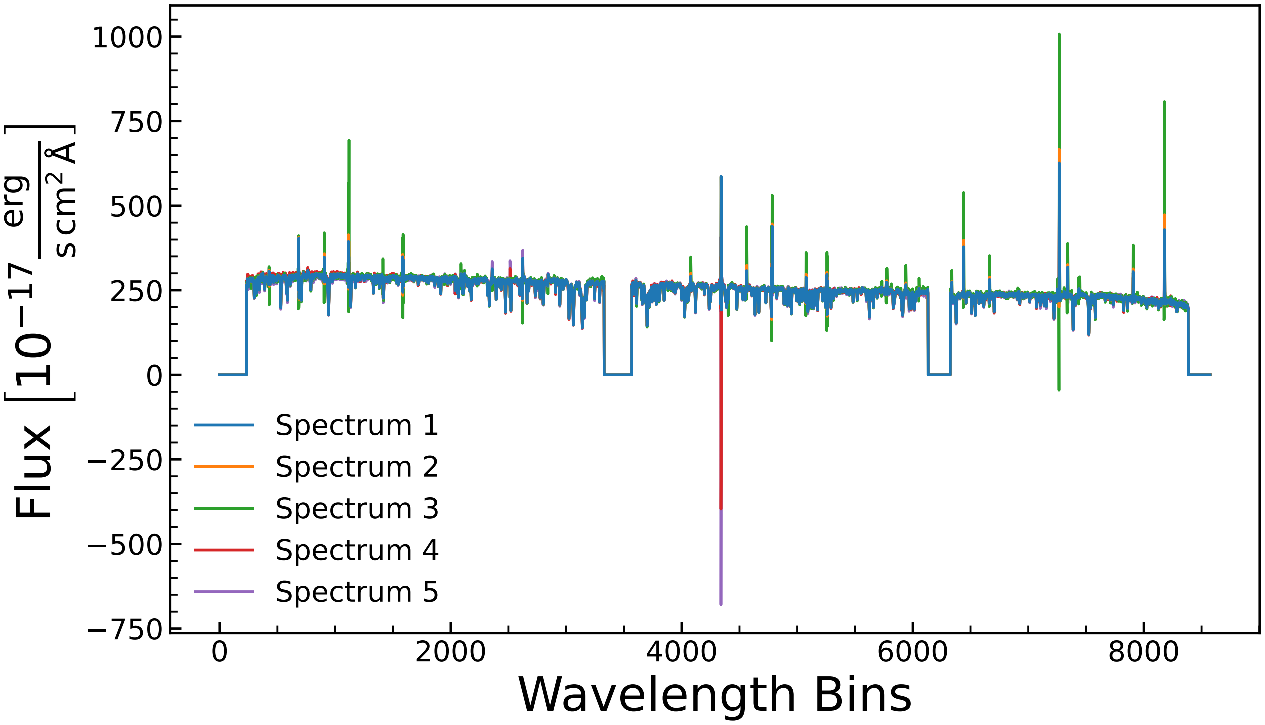

Figure 1. A flux vs wavelength bins (or pixels) plot showing multi-visit spectra for star 2M21275176+1219275. The field, or location, and the name of the star is shown in the title. On the vertical is the flux in cgs units, while the horizontal denotes the number of pixels (total 8575).

Table 2. A table of the relevant 8 out of 17 bitmasks with the bit name, bit digit, and a description of each bit. For more information see APOGEE_PIXMASK.

Bit name	Binary digit	Description
BADPIX	0	Pixel marked as BAD in bad pixel mask
CRPIX	1	Pixel marked as cosmic ray in ap3d
SATPIX	2	Pixel marked as saturated in ap3d
UNFIXABLE	3	Pixel marked as unfixable in ap3d
BADDARK	4	Pixel marked as bad as determined from dark frame
BADFLAT	5	Pixel marked as bad as determined from flat frame
BADERR	6	Pixel set to have very high error (not used)
NOSKY	7	No sky available for this pixel from sky fibers
SIG_SKYLINE	12	Pixel falls near sky line that has significant flux compared with object

2.3. *Pseudo-continuum Normalization of Spectra*

Ness et al. (2015) performed a true continuum normalization to determine pixels where the normalized flux show nearly no dependence on stellar properties. The procedure was an iterative application of *The Cannon* starting with the construction of a pseudo-continuum: a polynomial fit to the upper 90th percentile of the spectra with a running quantile across a 50 Å window. Spectra normalized with this pseudo-continuum was then passed through *The Cannon* to identify the "true" continuum pixels. The true continuum is then fitted with a second-order Chebyshev polynomial, with the fit being used to normalize all the spectra. For more details, refer to sections 2.3 and 5.3 of Ness et al. (2015).

A similar normalization procedure was applied to all the spectra in this lab. A pseudo-continuum normalization was carried out using a given list of wavelengths (cannon_continuum_apogee.npz) that do not

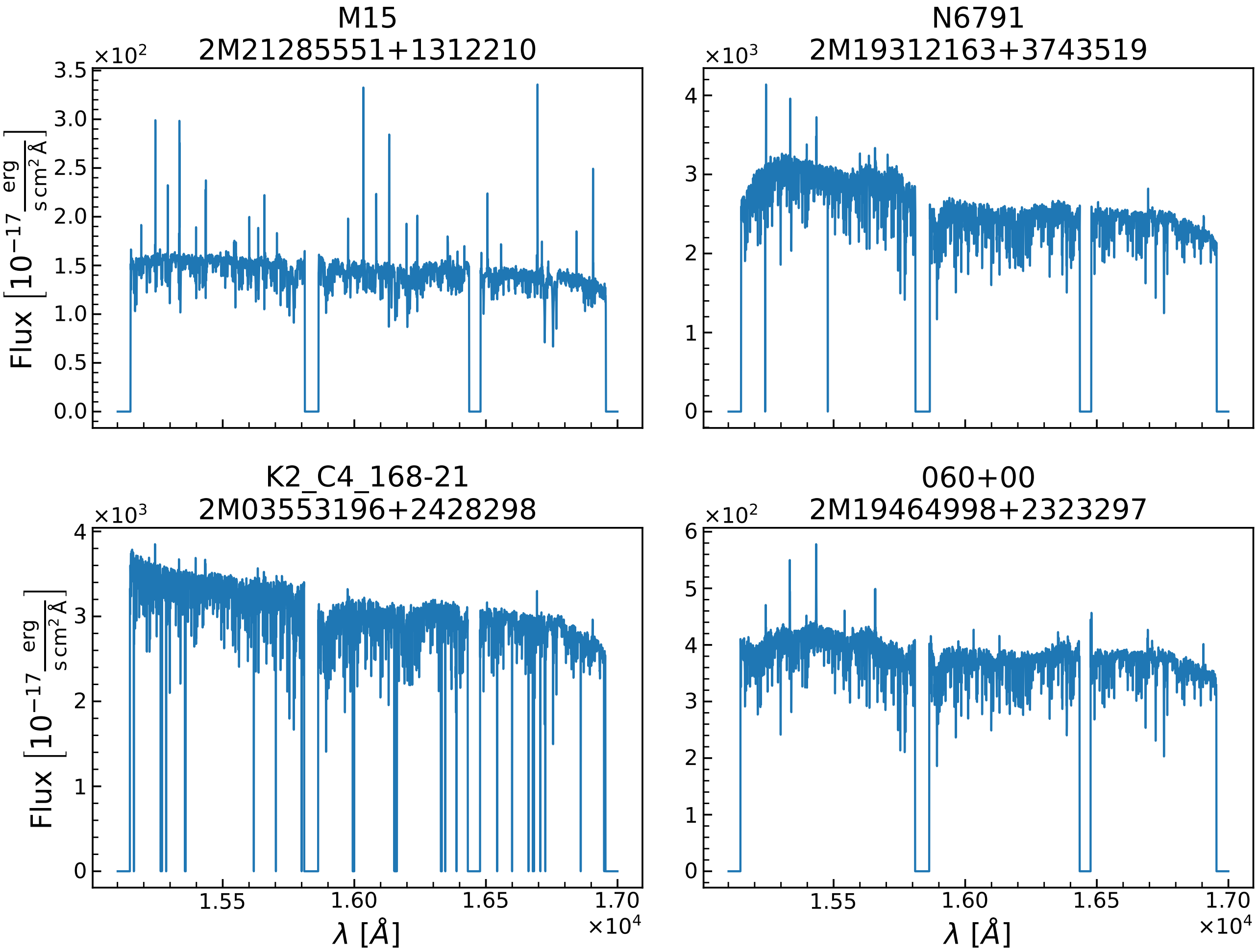

Figure 2. Plots of flux vs reconstructed wavelength for four random spectra in each of the four randomly chosen fields. The field and name of each star are shown in the title of each plot. Flux is measured in cgs units, and the reconstructed wavelengths are shown in angstroms.

have strong dependence on spectral labels, but instead only on the absolute magnitude and distance of the star. Splicing the spectra with the indices of these wavelengths essentially constructs a continuum that can be used for Chebyshev fitting.

The wavelength grid from the *apStar* files and from the given `cannon_continuum_apogee.npz` are on different wavelength scales. So before normalization, all spectra from the *apStar* files were interpolated onto a given fine wavelength grid (`apogee_wavelengths.npz`) in order to better match continuum pixels. APOGEE spectra and inflated errors (see section 2.2 and appendix A) were interpolated onto the new wavelength grid using `numpy.interp`. Continuum pixels were then identified by slicing these interpolated spectrum and inflated errors to match with wavelength pixels from the given continuum file. A second-order Chebyshev polynomial was then fitted to these continuum pixels, using `numpy.polynomial.chebyshev.chebfit` and `numpy.polynomial.chebyshev.chebval` for fitting and evaluating the continuous continuum, respectively. All spectra were then normalized with the derived pseudo-continuum. Note that due to gaps in the instrument's detectors, pseudo-normalization was carried out for three individual wavelength chips (see the end of section 2.1). For an example, see the unnormalized spectrum, pseudo-continuum fit, and pseudo-continuum normalized spectrum for the star 2M19395986+2341280 in figure 4.

2.4. *Creating Training & Cross-Validation Sets*

The training data set for the spectra model consists of spectra with known labels derived from the ASPCAP pipeline (see García Pérez et al. (2016)). These spectra are downloaded from the *allStar* catalog. In this lab, I use five ASPCAP labels (atmospheric parameters): the effective temperature $\mathrm{T_{eff}}$, the surface gravity $\log \mathrm{g}$, chemical abundances [Fe/H], [Mg/Fe], and [Si/Fe].

Spectra are then discarded according to the following conditions:

- There are no labels for T_{eff}, log g, [Fe/H], [Mg/Fe], [Si/Fe]
- There is low S/N: SNR < 50
- Spectra of dwarf stars ($\log g > 4$, or $T_{\mathrm{eff}} > 5,700\,\mathrm{K}$)
- Spectra of stars with low metallicity [Fe/H] < -1

Spectra without derived stellar properties have label values set at -9999. So to filter for stars with the five desired labels, I filter the data set for labels with values greater than -9999. To remove spectra with low S/N, I filtered for SNR values greater than 50. To discard spectra with low metallicity, I applied a [Fe/H] greater than -1 filter. Spectra for giants were selected for by filtering for $\log \mathrm{g} < 4$ and $\mathrm{T_{eff}} < 5,700\,\mathrm{K}$. There are 1854 stellar spectra that satisfy these conditions. A corner plot showing the distribution of stellar spectra for each ASPCAP labels is shown in figure 3.

The surface gravity g of a star is the gravitational acceleration at its surface at the equator. The surface gravity obeys the inverse square law: $g \propto R^{-2}$ where R is the radius of the star. For example, consider a solar-mass star. From least to greatest in magnitude, the expected surface gravity on the main sequence (MS), just before the helium flash (HeF), and during the core helium burning (HeB) phase are

$$g_{\mathrm{HeB}} \sim \frac{M_\odot}{\left(100R_\odot\right)^2} < g_{\mathrm{HeF}} \sim \frac{M_\odot}{\left(15R_\odot\right)^2} < g_{\mathrm{MS}} \sim \frac{M_\odot}{\left(R_\odot\right)^2}\,.$$

In log space, means larger stars, e.g. giants, have smaller $\log g$ values than smaller stars, e.g. dwarfs. Observe from figure 3 that there is a positive correlation betwen effective temperature and surface gravity, i.e. giants have lower surface gravity and cooler effective temperatures than dwarf stars. Therefore the filter for $\log g < 4$ and $T_{\mathrm{eff}} < 5,700\,K$ effectively filters out dwarf spectra from the *allStar* catalog.

To construct the training and cross-validation data sets for the spectra model, I first constructed a file list for all `.fits` files in the downloaded and filtered *allStar* catalog. Then I randomly split up the file list and labels into two equal-size sets using `sklearn.model_selection.train_test_split`. Spectra from *apStar* files were cross-matched with file names from the *allStar* training and cross-validation sets, and mass pseudo-normalized with the methods described in section 2.3. The dimensions of the resulting wavelength, spectrum, and error arrays for each set is 927 spectra × 7174 pixels (less than 8575 pixels because I left out gap pixels between the three wavelength chips).

2.5. *Constructing the Spectra Model*

Following Ness et al. (2015), the spectra model constructed here is a second-order polynomial in labels. The starting point is a spectra model that predicts a spectrum at each wavelength pixel as a function of the five atmospheric parameters: T_{eff}, $\log g$, [Fe/H], [Mg/Fe], and [Si/Fe]. The final spectra model is a combination of thousands of individual models, one for each wavelength pixel. For each wavelength, I also fit an intrinsic scatter term, s_λ^2. The intrinsic scatter is defined so that the variance in the observed normalized flux at each wavelength is

$$\sigma_{\mathrm{total}}^2 \equiv s_\lambda^2 + \sigma_\lambda^2\,, \tag{2}$$

where σ_λ is the uncertainty in normalized flux. The scatter term accounts for systematic and other uncertainties per pixel. The noise model is defined to be the total variance in normalized flux (equation 2).

A model relating the pseudo-normalized flux to stellar labels can be written as

$$f_{n\lambda} = \boldsymbol{\theta}_\lambda^\intercal \cdot \boldsymbol{\ell}_n + \mathrm{noise} \tag{3}$$

where for a single wavelength pixel λ and a star n: $f_{n\lambda}$ is the pseudo-normalized flux scalar, $\boldsymbol{theta}_\lambda$ is the coefficient or free parameter vector, and $\boldsymbol{\ell}_n$ is the label vector.

To make the model a second-order polynomial in labels, the label vector $\boldsymbol{\ell}_n$ is defined to be quadratic in the labels ℓ_{nk}:

$$\boldsymbol{\ell}_n \equiv \begin{bmatrix} 1,\, \ell_{n1} - \bar{\ell}_{n1},\, \cdots,\, \ell_{nK} - \bar{\ell}_{nK}, \\ \left(\ell_{n1} - \bar{\ell}_{n1}\right)^2,\, \left(\ell_{n1} - \bar{\ell}_{n1}\right)\left(\ell_{n2} - \bar{\ell}_{n2}\right),\, \cdots, \\ \left(\ell_{nK} - \bar{\ell}_{nK}\right)^2 \end{bmatrix} \tag{4}$$

where ℓ_{nk} is a single label where k = 1, 2, ..., K, and $\bar{\ell}_{nk}$ are label offsets. The offsets $\bar{\ell}_{nk}$ are chosen to be the means of the training data set in order for the spectra model to pivot around a reasonable point in label space. The element "1" allows for a linear offset in the fitting. All of the quadratic terms contain all possible label products exactly once.

Including the 1 element, the label vector $\boldsymbol{\ell}_n$ and coefficient vector $\boldsymbol{\theta}_\lambda$ both have dimensions $\frac{1}{2}\left(k+1\right)\left(k+2\right) \times 1$. For $k = 5$ unique stellar labels, there are 21 free parameters in $\boldsymbol{\theta}_\lambda$. In this case, the label vector is

$$\boldsymbol{\ell}_n = \begin{bmatrix} 1,\, T_{\mathrm{eff}},\, \log g,\, [\mathrm{Fe/H}]\,,\, [\mathrm{Mg/Fe}]\,,\, [\mathrm{Si/Fe}]\,, \\ T_{\mathrm{eff}}^2,\, T_{\mathrm{eff}} \cdot \log g,\, T_{\mathrm{eff}} \cdot [\mathrm{Fe/H}]\,,\, \cdots, \\ \left(\log g\right)^2,\, \log g \cdot [\mathrm{Fe/H}]\,,\, \cdots, \\ [\mathrm{Fe/H}]^2\,,\, [\mathrm{Fe/H}] \cdot [\mathrm{Mg/Fe}]\,,\, \cdots, \\ [\mathrm{Mg/Fe}]^2\,,\, [\mathrm{Mg/Fe}]\,[\mathrm{Si/Fe}]\,, \\ [\mathrm{Si/Fe}]^2 \end{bmatrix}.$$

Table 3. Descriptions of label columns pulled from *allStar* catalog. For details, go here or see García Pérez et al. (2016).

Column name	Units	Description
APOGEE_ID		2MASS ID
FILE		*apStar* file name
FIELD		Field name
SNR		S/N estimate
TEFF	K	$\mathrm{T_{eff}}$ from ASPCAP analysis of combined spectrum (from PARAM)
TEFF_ERR	K	$\mathrm{T_{eff}}$ uncertainty (from PARAM_COV)
LOGG	log [cm/s^2]	log g from ASPCAP analysis of combined spectrum (PARAM)
LOGG_ERR	log [cm/s^2]	log g uncertainty (PARAM_COV)
FE_H	dex	[Fe/H] from ASPCAP analysis of combined spextrum (PARAM)
FE_H_ERR	dex	[Fe/H] uncertainty (PARAM_COV)
MG_FE	ditto	ditto
MG_FE_ERR	ditto	ditto
SI_FE	ditto	ditto
SI_FE_ERR	ditto	ditto
ASPCAPFLAG		Bitmasks indicating possifle issues with individual measurements from ASPCAP fits

With N_{ref} reference objects n with known intrinsic scatter s^2_λ (so that the noise term is a constant), the model can be written as a linear matrix equation:

$$\mathbf{f}_\lambda = \mathbf{X}\boldsymbol{\theta}_\lambda \tag{5}$$

where the flux vector $\mathbf{f}_\lambda$ is now a $N_{\mathrm{ref}} \times 1$ vector containing pseudo-normalized flux of N_{ref} stars for a wavelength pixel λ. The coefficient vector $\boldsymbol{\theta}_\lambda$ is still a $\frac{1}{2}(k+1)(k+2) \times 1$ vector. Therefore the matrix $\mathbf{X}$ is a $N_{\mathrm{ref}} \times \frac{1}{2}(k+1)(k+2)$ matrix of labels:

$$\mathbf{X} \equiv \begin{pmatrix} \ell_1^T \\ \vdots \\ \ell_{N_{\mathrm{ref}}}^T \end{pmatrix} \tag{6}$$

To summarize, 1854 stellar spectra were filtered out from the *allStar* catalog and split into two equal-size (927) training and cross-validation data sets. The matrix $\mathbf{X}$ was created with label vectors from the training set. As a result, $\mathbf{X}$ has dimension 921×21. Ness et al. (2015) defines a single-pixel log-likelihood function:

$$\ln p\left(f_{n\lambda} \,|\, \boldsymbol{\theta}^T_\lambda,\, \boldsymbol{\ell}_n,\, s^2_\lambda\right) = -\frac{1}{2}\frac{\left(f_{n\lambda} - \boldsymbol{\theta}^T_\lambda \cdot \boldsymbol{\ell}_n\right)^2}{s^2_\lambda + \sigma^2_{n\lambda}} - \frac{1}{2}\ln\left(s^2_\lambda + \sigma^2_{n\lambda}\right)\,, \tag{7}$$

There are 7174 pixels, so the total likelihood function is the sum of all 7174 single pixel log likelihood functions. There are two methods to find the best values of s^2_λ and $\boldsymbol{\theta}_\lambda$: optimizing the likelihood

$$\boldsymbol{\theta}_\lambda,\, s_\lambda \leftarrow \overset{\text{argmax}}{\boldsymbol{\theta}_\lambda,\, s_\lambda} \sum_{n=1}^{7174} \ln p\left(f_{n\lambda} \,|\, \boldsymbol{\theta}^T_\lambda,\, \boldsymbol{\ell}_n,\, s^2_\lambda\right)\,, \tag{8}$$

or using Bayesian inference and Markov Chain Monte Carlo (MCMC) with defined priors.

2.5.1. *Optimizing Likelihood Function*

Optimization of the log-likelihood function starts with defining a grid of scatter s^2_λ for each wavelength pixel. The grid is defined to have 15 s^2_λ values evenly spaced within a range of $10^{-5} \to 1$. For each scatter value, a weighted-least square was performed using `numpy.linalg.lstsq` to solve for the coefficient vector $\boldsymbol{\theta}_\lambda$. A weights matrix $\mathbf{W}$ of dimension 927×927 is constructed with inverse of the square root of the total variance, as defined in equation 2, on the diagonal with σ_{total} replaced by a 927×1 vector $\boldsymbol{\sigma}_{\mathrm{total}}$:

$$\mathbf{W} \equiv \begin{pmatrix} \sigma_{1,\,\mathrm{total}} & & \\ & \ddots & \\ & & \sigma_{927,\,\mathrm{total}} \end{pmatrix}\,. \tag{9}$$

The label matrix $\mathbf{X}$ and pseudo-normalized flux vector $\mathbf{f}_\lambda$ are then weighted using $\mathbf{W}$. Weighing $\mathbf{X}$ and $\mathbf{f}_\lambda$ ensures that bad pixels do not contribute much to the model (see section 2.2). The coefficient vector $\boldsymbol{\theta}_\lambda$ and scatter term s_λ were then with equation 7 to find the log-likelihood. The $\boldsymbol{\theta}_\lambda$ and s_λ that correspond to a global maximum in the log-likelihood was determined using `numpy.argmax`. Repeating for all 7174 pixels resulted in a 1-dimensional array of best intrinsic scatter values and a 7174×21 array of coefficient vectors. That means there are a total of 150,654 free parameters in the spectra model.

2.5.2. *Spectra Model with MCMC*

The second way to solve for the best $\boldsymbol{\theta}_\lambda$ and s_λ is to perform Bayesian inference using a MCMC sampler. I use the No-U-Turn sampler from `pymc3` with uniform priors for each label. The ranges for the uniform priors are: $0 \leq T_{\rm eff} \leq 10^4$, $0 \leq \log g \leq 10$, $-2 \leq$ [Fe/H] ≤ 2, $-2 \leq$ [Mg/Fe] ≤ 2, and $-2 \leq$ [Si/Fe] ≤ 2. The ranges were defined based on physical ranges of the stellar labels. The total log-likelihood function, or the sum of equation 7 for all pixels, was then explicitly enrolled into the `pymc3` model.

2.6. *Applying The Cannon Model*

2.6.1. *Constructing the Gradient Spectra*

For each of the five labels $\ell_{n\lambda}$, the gradient spectrum $\partial f_\lambda / \partial \ell_{n\lambda}$ was created with numerical differentiation:

$$\frac{\partial f_\lambda}{\partial \ell_{n\lambda}} \approx \frac{f_{n\lambda}\left(\ell_{n\lambda} + \Delta\ell_{n\lambda}\right)}{\Delta\ell_{n\lambda}}.$$

This was implemented by modifying the model to add a variation to one of the five labels, predicting a pseudo-continuum normalized spectrum with this modified label, take the difference between this varied normalized spectrum and the original model normalized spectrum without variation to the label, and divide the difference by the variation in the label. The process is then repeated for the remaining labels.

2.6.2. *Cross-validation Fitting with a Non-linear Optimizer*

Since the label vector is quadratic in the labels (see equation 4), a non-linear optimizer is required to fit the labels of spectra in the cross-validation set. The *Trust Region Reflective* (trf) method from `scipy.optimize.curve_fit` was used to determine a point in label-space for which the predicted model spectrum is in good agreement with the observed pseudo-continuum normalized spectrum. The means of labels from the training data set was used as the initial guess for the fitting parameters, with the uncertainty set to the total uncertainty in equation 2. The maximum number of iterations in the `curve_fit` function was increased to 5000 to ensure that the optimizer can fully run. The result is discussed in section 3.5.

3. RESULTS & DISCUSSIONS

3.1. *ASPCAP Stellar Labels*

Figure 3 shows the distribution of stellar spectra for each ASPCAP-derived stellar labels. Note the positive correlation between the log of surface gravity and effective temperature, supporting the constraint for large values of $\log g$ and $T_{\rm eff}$ when filtering out dwarf stars from the *allStar* catalog. The mean and uncertainties for each stellar labels are shown in table 4. These values will be useful for comparison with the spectra model labels in later sections.

Table 4. Tabulated values for the bias (mean) and scatter (standard deviation) for ASPCAP derived stellar labels. Refer to the histograms in figure 3.

ASPCAP Labels	Bias	Scatter
$T_{\rm eff}$	4682.93	$^{+295.27}_{-522.72}$
$\log g$	2.44	$^{+0.62}_{-0.83}$
[Fe/H]	-0.07	$^{+0.23}_{-0.33}$
[Mg/Fe]	0.05	$^{+0.17}_{-0.07}$
[Si/Fe]	0.02	$^{+0.10}_{-0.04}$

Table 5. Tabulated values for second-order Chebyshev polynomial coefficients used to fit the continuum pixels of star 2M19395986+2341280, see figure 4. There are three sets of coefficients since the Chebyshev fit was performed for three individual wavelength chips.

Wavelength chips [Å]	Chebyshev coefficients
15150-15800	6610, -0.767, $1.16 \cdot 10^{-5}$
15890-16430	31300, -3.83, $5.92 \cdot 10^{-5}$
16490-16950	-67500, 8.16, $-1.23 \cdot 10^{-4}$

3.2. *Pseudo-Continuum Normalization of Star 2M19395986+2341280*

Figure 4 shows unnormalized, second-order Chebyshev polynomial fit, and pseudo-continuum normalized spectra of the star 2M19395986+2341280. In the background of the unnormalized spectrum is the raw APOGEE spectrum data. The foreground spectrum is interpolated to a fine wavelength grid in order to closely match the flux pixels with continuum pixels, see 2.1. Wavelength pixels corresponding to a continuum, i.e. pixels that should not have strong dependence on stellar labels, are overplotted on the spectra as a scatter. The second panel of the figure shows a second-order Chebyshev polynomial fit to the continuum pixels of the spectrum. The fit was performed individually for the three wavelength chips, and the Chebyshev coefficients are tabulated in table 5 from low to high (see `chebfit` documentation).

For all three wavelength chips, the Chebyshev fit to the continuum spectrum follows closely with the top of the spectrum data. However, polynomial fits do not behave well for the edge cases, as observed by the polynomial having a positive concavity for wavelegnth chip 15890 → 16430 Å and a negative concavity for chip 16490 → 16900 Å. In order for the polynomial fit to properly fit a continuum, or characterize a black-body spectrum, the curve should ideally be flat as in the first wavelength chip 151500 → 15800 Å. The discrepency

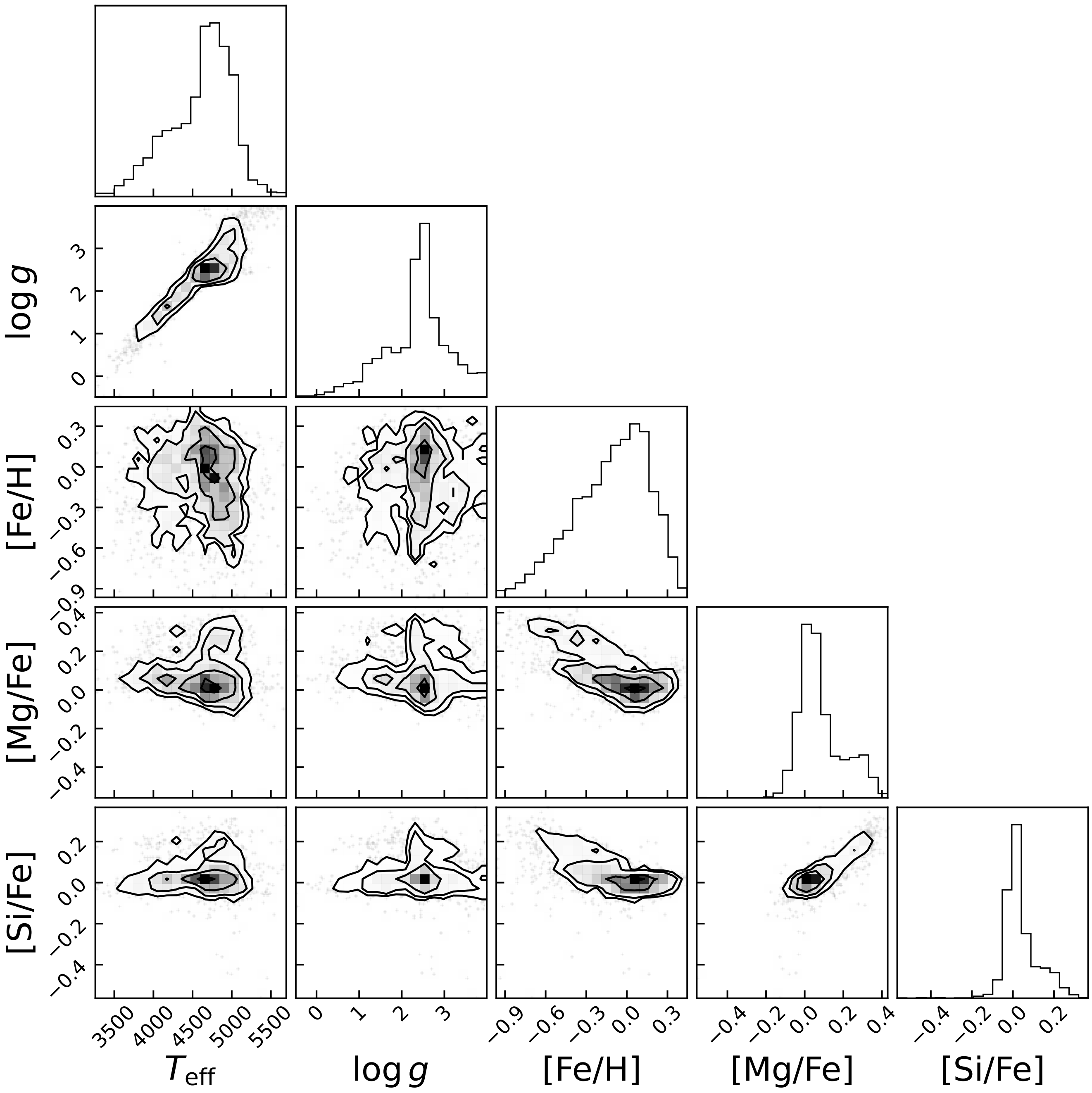

Figure 3. A corner plot showing the distribution of stars from the *allStar* catalog in stellar label space. The histograms show the distribution of each ASPCAP derived label, with the mean and standard deviation shown in table 4.

can be chalked up to the Chebyshev being poorly constrained at the boundaries. However, it is also a possibility that the continuum pixels obtained by matching the spectrum pixel to the given trusted wavelengths still do not closely resemble a true continuum. Most of the continuum pixels closely follow the top of the spectra, which is consistent with their independence from stellar labels. There remains outlier scatter pixels that deviate away from the continuum, indicating that these pixels still do not match closely with a true continuum.

3.3. *Prediction of Normalized Spectrum from ASPCAP Labels*

A prediction for the normalized spectrum of star 2M03533659+2512012 is shown in figure 5. 2M03533659+2512012 is a star from the training data set, and its normalized spectrum was predicted using

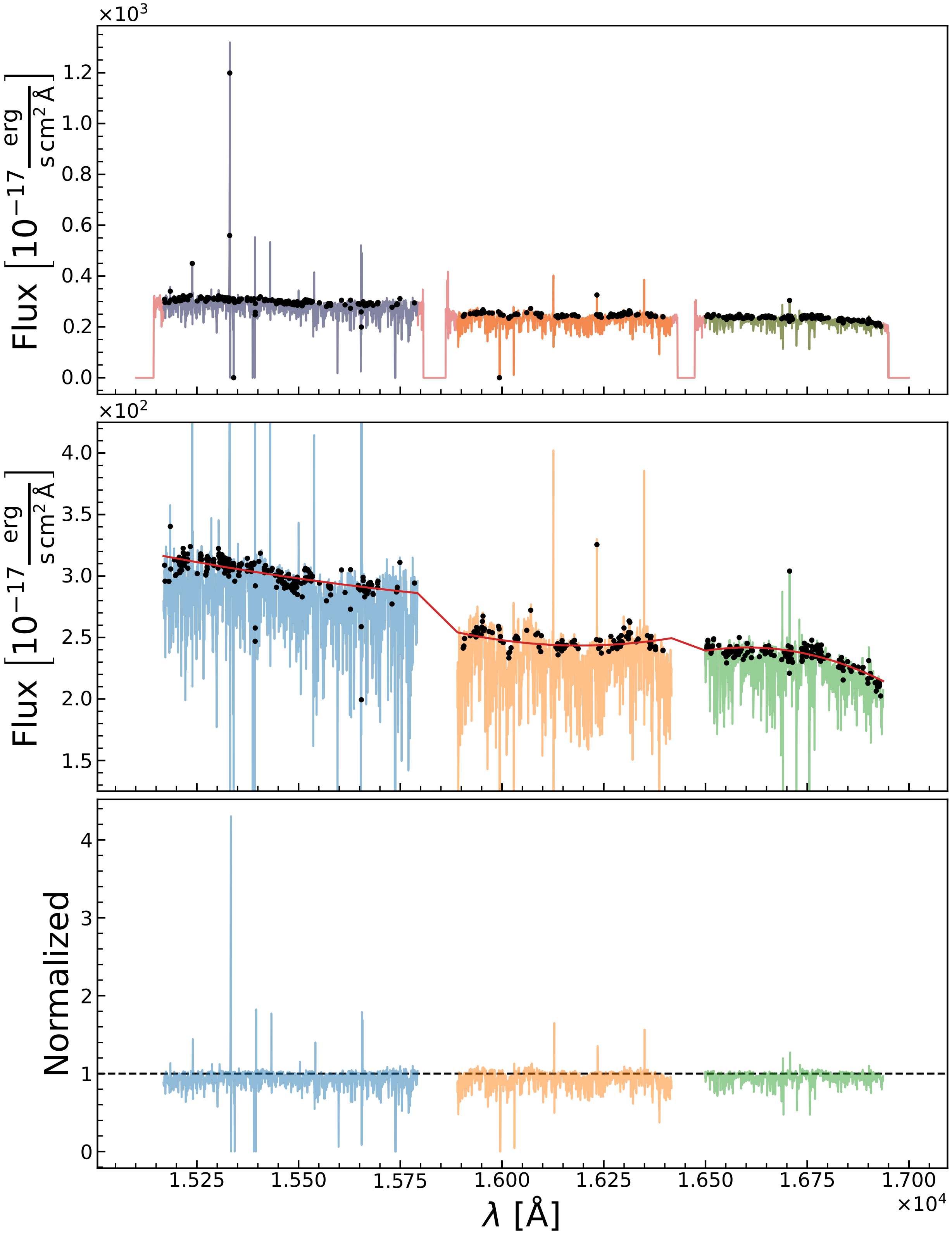

Figure 4. Flux vs wavelength spectra plots for the star 2M19395986+2341280. The spectra are split into three wavelength chips: 151500 → 15800 Å, 15890 → 16430 Å, and 16490 → 16900 Å. The top panel shows the raw spectrum in the background splitting of wavelength chips and interpolation. In the foreground is a plot of the interpolated spectrum along with a scatter plot of continuum pixels. The middle panel includes the second-order Chebyshev polynomial fit to the continuum pixels for each wavelength chips. The Chebyshev polynomial coefficients are shown in table 5. The bottom panel is the continuum normalized spectrum.

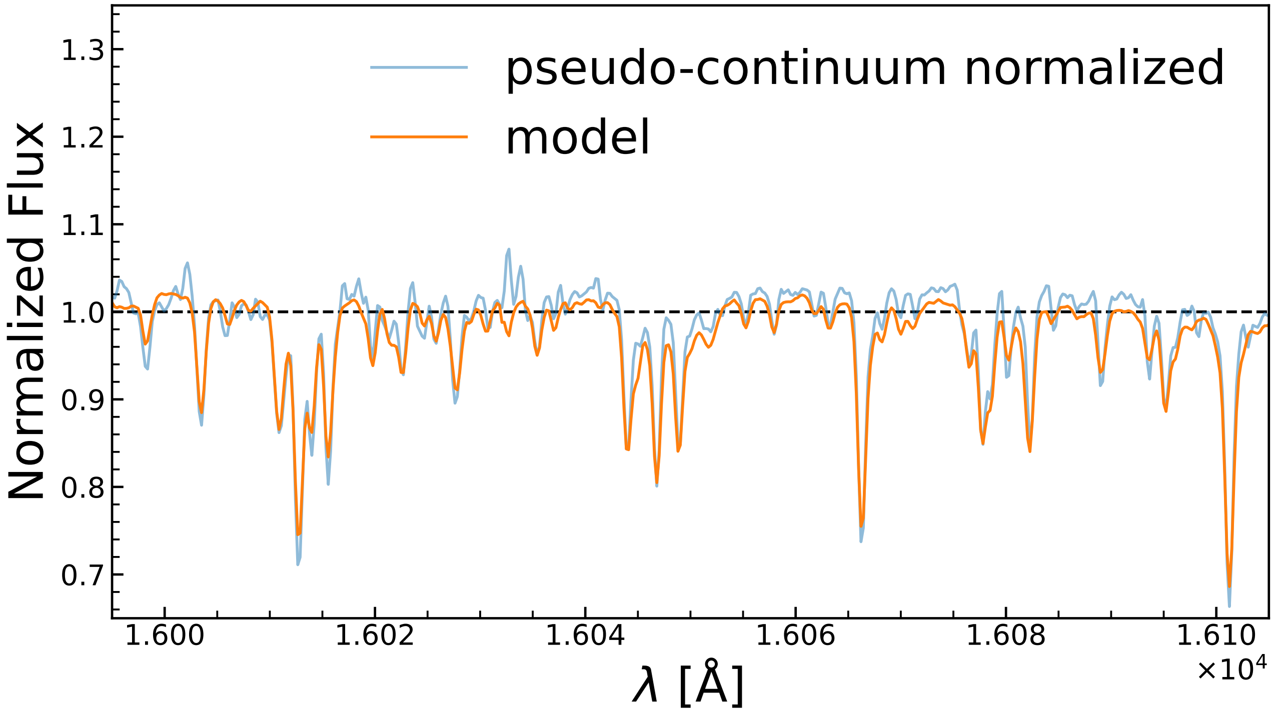

Figure 5. Comparison between model and data pseudo-continuum normalized spectra for star 2M03533659+2512012. The blue spectrum was interpolated and pseudo-continuum normalized as discussed in section 2.3. The orange spectrum is the predicted normalized spectrum using a spectra model constructed with optimization of a log-likelihood function. The wavelength range 16000 Å → 161000 Å was chosen for better visualization of the spectra.

the spectra model and the star's ASPCAP-derived labels. 2M03533659+2512012's spectrum was normalized with a pseudo-continuum, as discussed in section 2.3. The model captured the overall shape of the spectrum, but the model also consistently underestimated the normalized flux magnitude. Ideally, the data and model spectra should closely resemble each other. The presence of deviations indicate that the model is not robust. It is possible that the fine grid of intrinsic scatter chosen in section 2.5.1 is not letting the optimization process to find a global maximum for the log likelihood function, and thus yielding non-optimal values for intrinsic scatter s^2_λ and coefficient vectors $\boldsymbol{\theta}_\lambda$. Additionally, the pseudo-continuum normalization process described in section 2.3 still contain artifacts, which likely contributed to the underestimation by the spectra model.

3.4. *Comparison of Absorption Lines to Label Gradients and Intrinsic Scatter*

Gradient spectra with respect to labels allows the identification of wavelengths that are most sensitive to a particular label. Figure 6 show the gradient spectra for the effective temperature and surface gravity. The "infinitesimal" variation in effective temperature was set to $\Delta T_{\rm eff} = 0.01$. The variation in $\log g$ was set to $\Delta \log g = 0.001$.

The gradient spectra for the three chemical abundances are shown in figure 7. Overplotted in each panel of the figure are known APOGEE absorption lines for Fe, MgI, and SiII. The wavelength values listed in the linked GitHub are measured wavelengths in air, so an index of refraction solution derived by Nikolai Piskunov was used to convert the values to wavelengths in vacuum:

$$\begin{aligned} n =& 1 + 8.336624212083 \cdot 10^{-5} \\ &+ \frac{0.02408926869968}{130.1065924522 - s^2} \\ &+ \frac{1.599740894897 \cdot 10^{-4}}{38.92568793293 - s^2} \end{aligned} \tag{10}$$

where $s = 10^4/\lambda_{\rm air}$ and the conversion is:

$$\lambda_{\rm vac} = n\lambda_{\rm air} \tag{11}$$

Table 6 contains converted wavelengths for the known absorption lines of FeI, MgI, and SiII. The wavelength range of 15350 → 15460 Å was the only range where a spectrum feature nearly line up with a known FeI absorption line at 15399.925 Å. However, it is difficult to distinguish whether or not there is an observable absorption line around that wavelength due to the noise in this range. On the other hand, the second and third panels show agreement between spectrum features for MgI absorption lines at 15745.017 Å, 15753.2 Å, and 15770.1 Å; and for SiII at 16064.397 Å.

Figure 8 is a plot of the intrinsic scatter term s^2_λ, with all of the known absorption lines for the three chemical abundances overplotted. The shaed area from 15600 → 15800 Å is shown in the right panel, where larger than average scatter values appear to correspond to MgI absorption lines at 15753.2 Å and 15770.1 Å.

3.5. *Fitting Labels with the Spectra Model*

3.5.1. *Comparison of The Cannon Label Outputs and ASPCAP Label Inputs*

The spectra model was used to fit for labels in the cross-validation data set using labels from the training data set. As mentioned before, since the label vector $\boldsymbol{\ell}_\lambda$ is quadratic in the labels $\ell_{n\lambda}$, a non-linear optimizer is needed for fitting; see section 2.6.2. Table 7 contains labels and uncertainties for star 2M212345439+1141389 from both the spectra model and the ASPCAP pipeline. In general, the label uncertainties from the model are much lower than the ASPCAP label uncertainties, indicating that the spectra model is overfitting the labels. Aside from the uncertainties, the labels predicted by the model are in good agreement with the ASPCAP values.

Figure 9 contain comparison plots between the ASPCAP-derived labels and *The Cannon*-predicted labels. The one-to-one reference line in each plots are simply the ASPCAP-derived labels plotted against themselves. The scatter of labels predicted by *The Cannon* follow the one-to-one reference line well, with the labels [Mg/Fe] and [Si/Fe] having the most outliers. The bias (mean) and scatter (standard deviation) for each label over the full cross-validation data set is shown in each label plot. The histograms on the right panel plot the residual in each label. The bias and scatter in each labels are small, indicating a good performance for the model. For example, the scatter in effective temperature is 81 K, 2 order of magnitude smaller than the typical effective temperature. Ideally, a scatter of 30 K in $T_{\rm eff}$ and 0.02 dex in [Fe/H] can be achieved with further modifications to the model.

Figure 4 in Ness et al. (2015) is a similar plot comparing their *The Cannon* predicted label outputs to ASPCAP-derived label inputs. Ness' model only considered three labels: $T_{\rm eff}$, $\log g$, and [Fe/H]. Overall, the bias in labels from Ness is larger than the bias shown in figure 9, while the scatter values are in good agreement. It is possible that the discrepency in the bias is due to Ness' model only having three labels, while here the *The Cannon* took in five labels. Ness' model is more robust in general, especially the continuum construction procedure. Hence, the bias obtained in figure 9 are likely less precise than the values obtained in Ness.

3.5.2. *Kiel Diagram of Red Giant Spectra*

Figure 10 shows a Kiel diagram ($\log g$ vs $T_{\rm eff}$) for 922 stars in the cross-validation set. The labels used to plot the scatter of spectrum pixels are the labels obtained through fitting with a non-linear optimizer (see section 2.6.2 and 3.5). Two MIST isochrones, one for [Fe/H] = 0

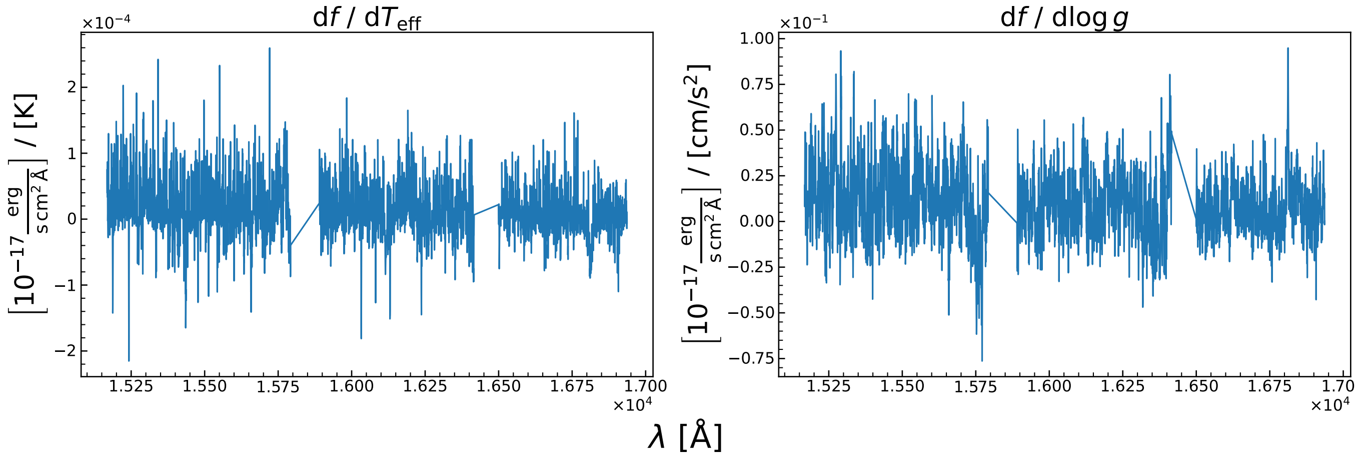

Figure 6. Gradient spectra with respect to the effective temperature and surface gravity for star 2M03533659+2512012.

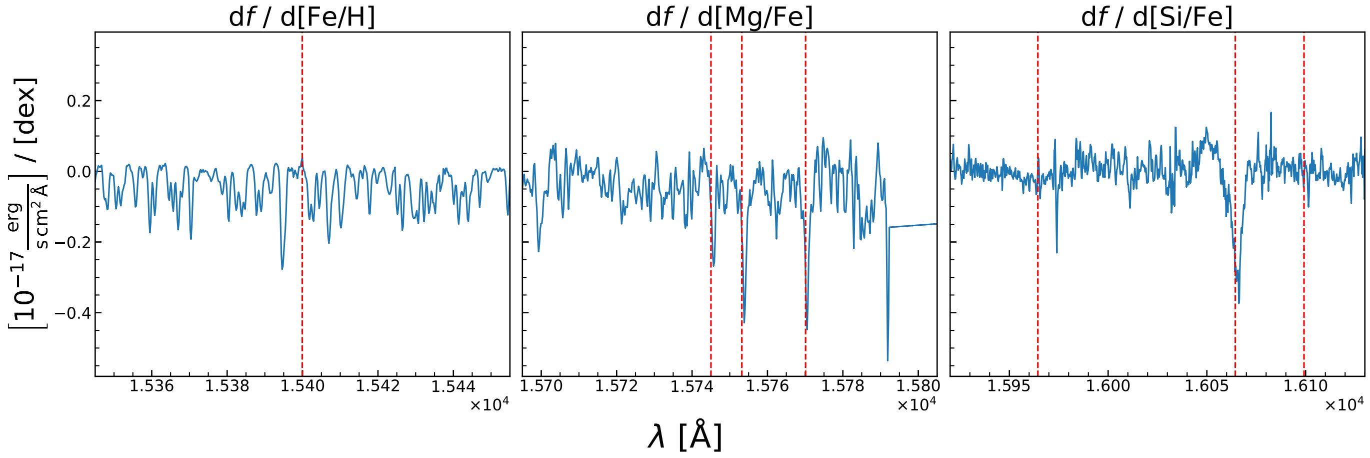

Figure 7. Three gradient spectra for the star 2M03533659+2512012. The gradients were numerically differentiated with respect to chemical abundances [Fe/H], [Mg/Fe], and [Si/Fe]. The wavelength ranges for each gradient spectra was chosen to show spectra features that nearly align with known absorption lines for each element, namely: 15745.017 Å, 15753.2 Å, and 15770.1 Å for Mg/Fe; 16064.397 Å for Si/Fe. There are no spectrum features that match up with known Fe/H absorption lines.

Table 6. A table of known APOGEE absorption lines for Fe, MgI, and SiII. The given wavelengths in air was converted to wavelengths in vacuum using the an index of refraction solution derived by Nikolai Piskunov.

Absorption Lines	Wavelength in air	Wavelength in vacuum
FeI	15194.492, 15207.526, 15395.718, 15490.339, 15648.510, 15964.867, 16040.657, 16153.247, 16165.032	15198.645, 15211.682, 15399.925, 15494.572, 15652.786, 15969.229, 16045.040, 16157.660, 16169.449
MgI	15740.716, 15748.9, 15765.8, 15879.5, 15886.2, 15954.477	15745.017, 15753.2, 15770.1, 15883.8, 15890.5, 15958.836
SiII	15361.161, 15376.831, 15833.602, 15960.063, 16060.009, 16094.787, 16215.670, 16680.770, 16828.159	15365.359, 15381.033, 15837.928, 15964.424, 16064.397, 16099.184, 16220.100, 16685.327, 16832.756

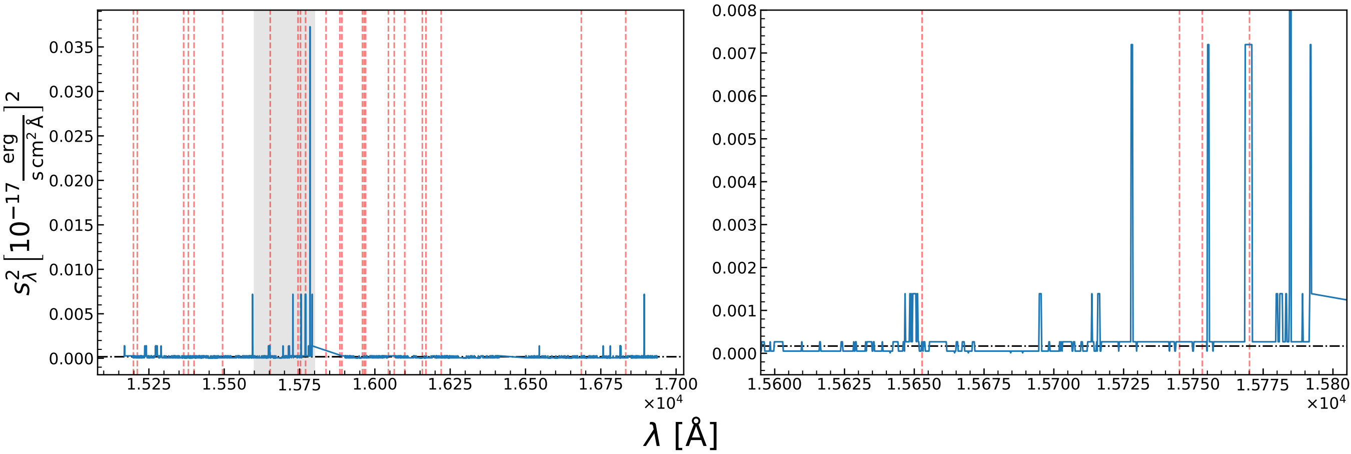

Figure 8. A plot of the intrinsic scatter uncertainty vs wavelengths. Overplotted are known absorption lines of Fe/H, Mg/Fe, and Si/Fe. The right panel zooms in on the shaded area in the left panel for the range 15600 → 15800 Å. The larger than average s_λ^2 in this range appear to line up with known absorption lines Mg/Fe at 15753.2 Å and 15770.1 Å. These wavelengths are in vacuum, for measured values in air, refer to table 6.

Table 7. A table of label and uncertainties optained from spectra model fitting with a non-linear optimizer compared with ASPCAP labels and uncertainties for the star 2M212345439+1141389, see section 2.6.2.

Label	Fit value	Fit uncertainty	ASPCAP value	ASPCAP uncertainty
T_{eff}	4698.78	24.56	4706.96	92.49
$\log g$	2.45	$1.67 \cdot 10^{-4}$	2.43	$6.70 \cdot 10^{-2}$
[Fe/H]	-0.52	$1.52 \cdot 10^{-5}$	-0.54	$1.02 \cdot 10^{-2}$
[Mg/Fe]	0.30	$1.21 \cdot 10^{-5}$	0.23	$1.46 \cdot 10^{-2}$
[Si/Fe]	0.24	$8.71 \cdot 10^{-6}$	0.15	$1.35 \cdot 10^{-2}$

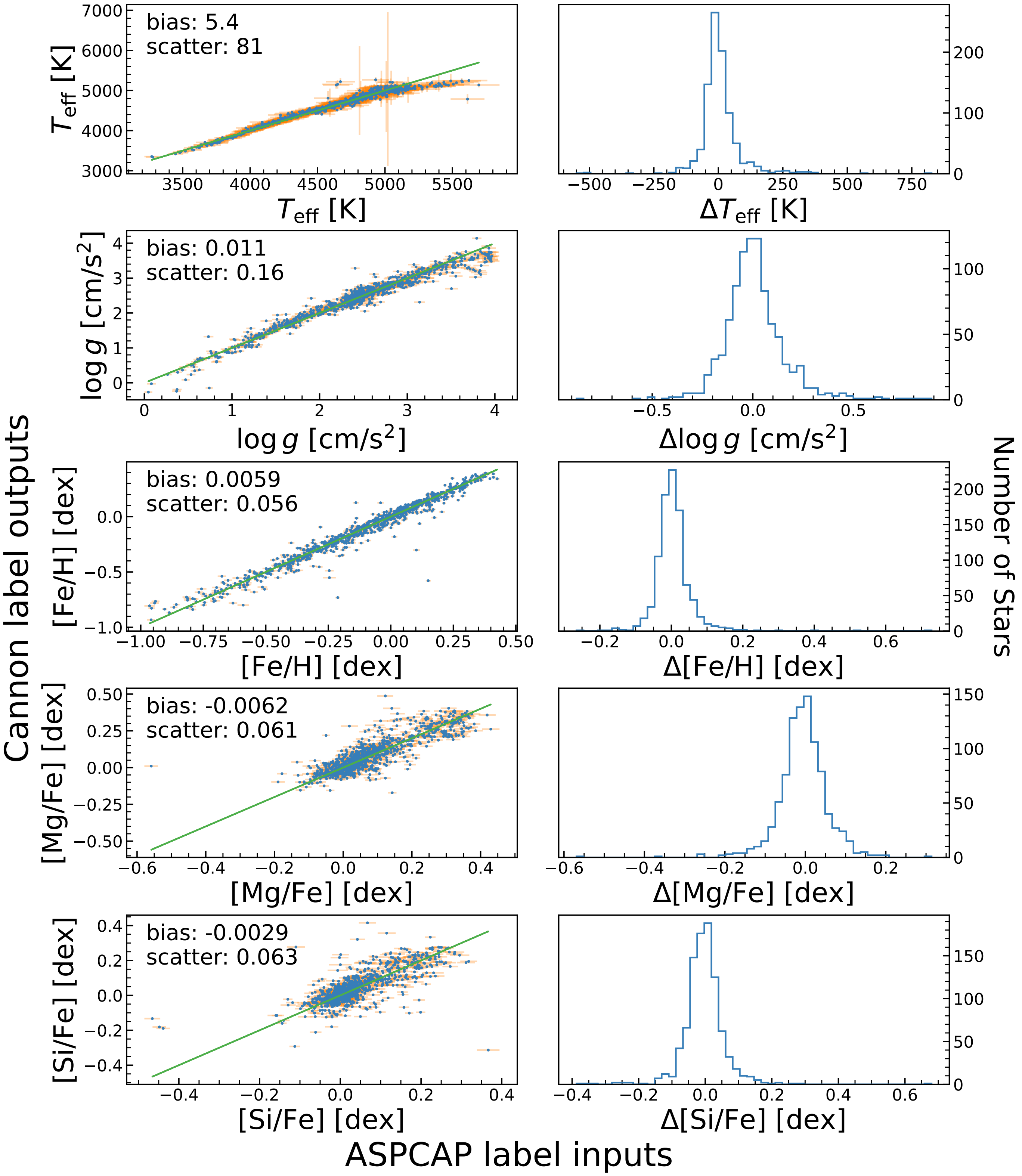

Figure 9. Comparison plots for *The Cannon* model fit of labels for the cross-validation spectra data vs their ASPCAP-derived labels. For each label, a one-to-one reference line was constructed by plotting ASPCAP vs ASPCAP labels. The right panel shows histograms with scatter of each labels and the corresponding number of stars in each label bin. The total number of label bins was set to 50. The bias (mean) and scatter (standard deviation) of labels are shown in each plot. Horizontal error bars are ASPCAP fit uncertainties in the labels (see table 3, while the vertical error bars are the uncertainties from the non-linear optimizer. Both uncertainties were pulled from the diagonal of the fitting parameter covariant matrix.

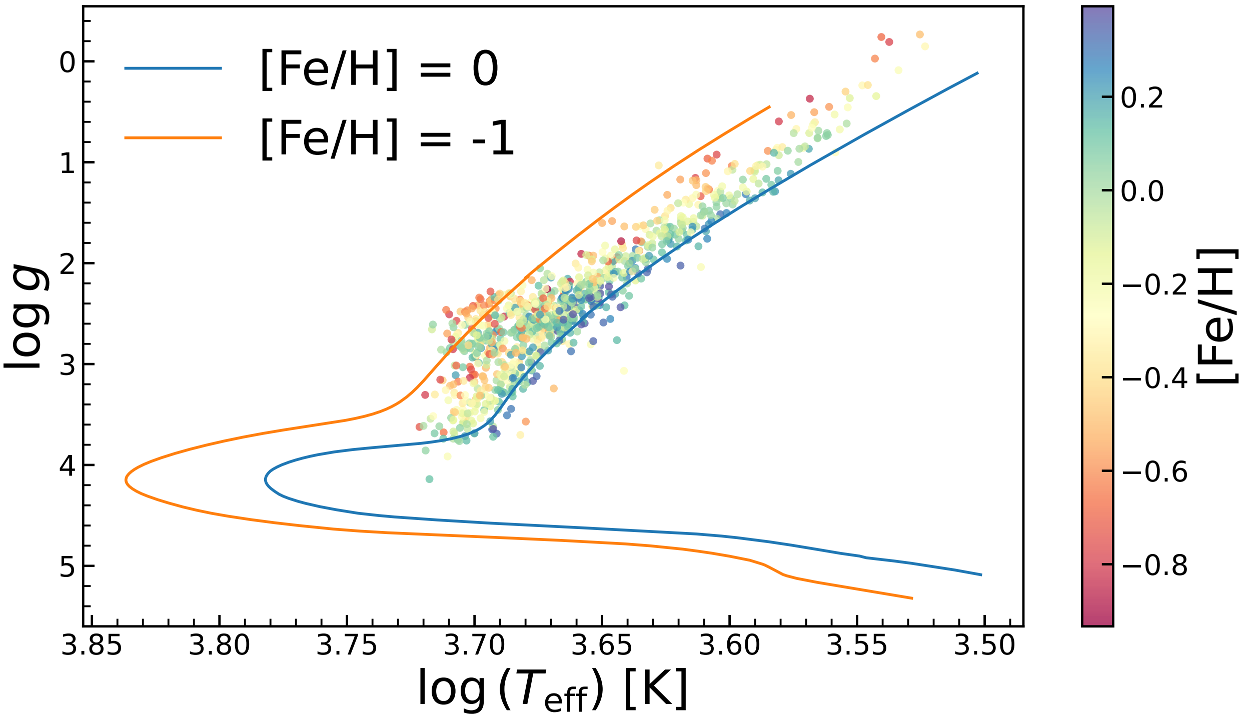

Figure 10. A Kiel diagram (surface gravity vs effective temperature) for 922 stars in the cross-validation data set (see section 2.4). The diagram is overplotted with two MIST isochrones: one with solar metallicity [Fe/H] = 0, and the other with [Fe/H] = −1. The MIST isochrones were filtered to only include the Main-Sequence and Red Giant Branch. The surface gravity and effective temperature axes were inverted to follow the convention of a Hertsprung-Russell diagram. Each spectrum pixel is colored by their metallicity [Fe/H]. A color bar on the right shows the distribution of metallicity.

or solar metallicity and one for [Fe/H] $= -1$, are overplotted onto the diagram. The theoretical isochrones were filtered for the Main-Sequence and Red Giant Branch. The color of the isochrones were chosen to be consistent with the color distribtution for the metallicity of spectrum pixels. Consequently, it is evident that the labels obtained from fitting line up well with the theoretical isochrones. Blue points with solar metallicity follow the RGB of the [Fe/H] $= 0$ isochrone, while orange points with comparatively lower metallicity are near the RGB of the [Fe/H] $= -1$ isochrone. Spectrum pixels with metallicities between 0 and -1 fall between the two isochrones, as expected.

3.6. *NUTS Spectra Model*

The spectra model was implemented into a `pymc3` model to utilize a Markov Chain Monte Carlo (MCMC) sampler, specifically the No-U-Turn Hamiltonian Monte Carlo sampler (NUTS), see section 2.5.2 for details. The `pymc3` model was then used to fit the labels of a given mystery spectrum. The mystery spectrum was then normalized with a pseudo-continuum, as discussed previously in section 2.3. The normalized spectrum is shown in figure 11.

A corner plot of the label sample from the posterior distribution is shown in figure 12, and the parameter uncertainties are collected in table 8. The uncertainty values of the `pymc3` sample are significantly smaller in comparison to the ASPCAP label uncertainty values in table 4. In comparison to the average scatter from the non-linear optimizer model, the `pymc3` uncertainties are still much smaller. The traceplot in figure **??** indicate that the posterior distribution converged properly. The issue might be in the prior distribution. If the prior distribution for the parameters is too narrow then the label estimates would be more constrained, leading to small uncertainties. Another explanation for the disagreement in uncertainty values can be due to the labels being on different scales. The effective temperature is on the order of 10^3, while the surface gravity is on the order of unity, and chemical abundances are on the order of 10^{-1}. Ideally, all the parameters should be rescaled to order unity, which can resolve the issue of smaller than expected uncertainties.

3.7. *Visualizing Spectra Changes with Offset and Color*

Two color offset spectra shown in figure 14 demonstrate how the spectrum of a star can change as the metallicity varies or as it evolves along the RGB. For the model spectra in the left panel, the mean values of labels in the training data set was used as the inputs for the spectra model. The exception being the metallicity [Fe/H], which was set to vary from -1.0 to 0.5. For the model spectra in the right panel, only the surface gravity and effective temperature were used. The range of the surface gravity was set to $0.5 < \log g < 3.5$ and effective temperature was set to $10^{3.5} < T_{\rm eff} < 10^{3.7}$. The ranges were set according to the Kiel diagram in figure 10 in order to inspect spectra of a hypothetical Red Giant.

Table 8. Tabulated values for the bias (mean) and scatter (standard deviation) for labels predicted by *The Cannon*, with its log-likelihood function (sum of equation 7) wrapped in `pymc3`'s NUT MCMC sampler (see section 2.5.2). For a visualization, refer to the corner plots in figure 12.

Labels predicted by `pymc3` model	Bias	Scatter
$T_{\rm eff}$	4493.89	$^{+3.36}_{-3.64}$
$\log g$	1.94	±0.01
[Fe/H]	-0.62	±0.004
[Mg/Fe]	0.20	±0.004
[Si/Fe]	0.17	±0.004

The expected result is that for higher metallicity (positive values having higher abundances), the absorption features in the spectra would become more prominent as more heavy elements are present to absorb photons at specific wavelengths that would have otherwise been transmitted to the observer. However, this expected behavior is not observed from the model spectra in the left panel of figure 14.

For a star with fixed metallicity, say solar metallicity [Fe/H] $= 0$, it is expected that the absorption lines in the spectra become less prominent as the star ascends the RGB. As it evolves along the RGB, the star expands while its mass and chemical abundances remain fixed. This corresponds to a decrease in the star's surface gravity, and thus an increase in the value of $\log g$. Heavy elements in the star that can contribute to absorption features in the star's spectrum become more spread out, meaning the mean free path of photons within the star becomes longer and it is more likely that photons can escape. All this translate broadened and diminished absorption lines with increasing $\log g$, which can be observed in the 6 offset spectra.

4. CONCLUSIONS

Overall, *The Cannon* model constructed here is not robust enough to reproduce reliable spectra and stellar parameters as derived by García Pérez et al. (2016) in their ASPCAP pipeline. The model underestimates the flux magnitude in its predicted spectra and outputs smaller than expected uncertainties for its label fits. However, the model is capable enough to produce label values that are within range of ASPCAP labels (figure 9. Its predicted spectrum capture the overall features of the normalized dataset (figure 5.

Further improvements can be made to the model. For example, instead of randomly sampling training sets, the process can include a signal-to-noise selection criteria as

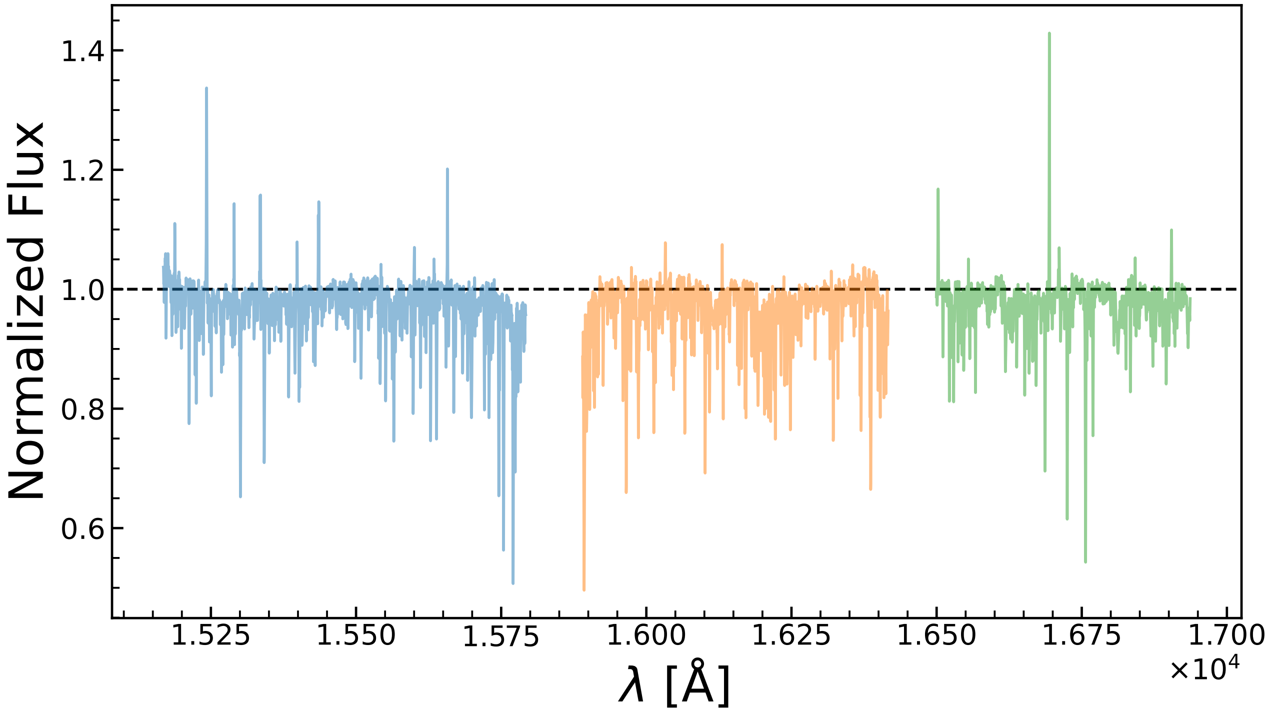

Figure 11. A pseudo-continuum normalized flux vs wavelength plot for a provided mystery spectrum. The three wavelength chips are: 151500 → 15800 Å, 15890 → 16430 Å, and 16490 → 16900 Å.

in Ness et al. (2015). The continuum normalization can be improved by running multiple iteration of *The Cannon* on the pseudo-continuum to produce pixels that more closely resemble a true continuum. Input parameters for the model can also be rescaled to unity in order to improve its performance, especially regarding the magnitude of uncertainty values.

ACKNOWLEDGEMENTS

Sincerest thanks to Professor Dan Weisz and graduate student instructors Anna Pusack and Olivia Aspegren for guidance and instruction on the conceptual and technical components throughout this lab. Thank you to Talia Saarinen, Bradley Arias, Jason Wong, and Andrew Goh for their feedback and support with programming.

APPENDIX

A. INFLATING FLUX UNCERTAINTY FOR BAD PIXELS FLAGGED BY BITMASKS

Flags for bad pixels are presented in the APOGEE HDU as integers that can be converted to binary digit representation:

$$\mathbb{Z} = 2^n + \cdots + 2^2 + 2^1 + 2^0\,. \tag{A1}$$

where each n is a binary digit. For example, the integer 512 can be written as 2^9 and in binary as 1000000000. In binary, the 9th binary digit/bit is set to 1 or `TRUE`.

Each bitmask integers in the APOGEE HDU were converted to binary representation using python's built-in `bin()` function. The indices for each binary representation were reversed and filtered for the relevant bits 0-7 and 12. Whenever any of those bits are set to 1, ie `TRUE`, the corresponding flux uncertainty was set to a large value 10^{10}. The code snippet is:

```
# initialize new errors
inflated_errors = errors.copy()

# convert bitmask integers to binary
masks_binary = [bin(n)[2:] for n in masks]

for index, mask in enumerate(masks_binary):
    # reverse index to start with 2^0
    mask_binary_reversed =
        masks_binary[index][::-1]

    # bit string longer than 1 digit
    if len(mask_binary_reversed) > 1:
```

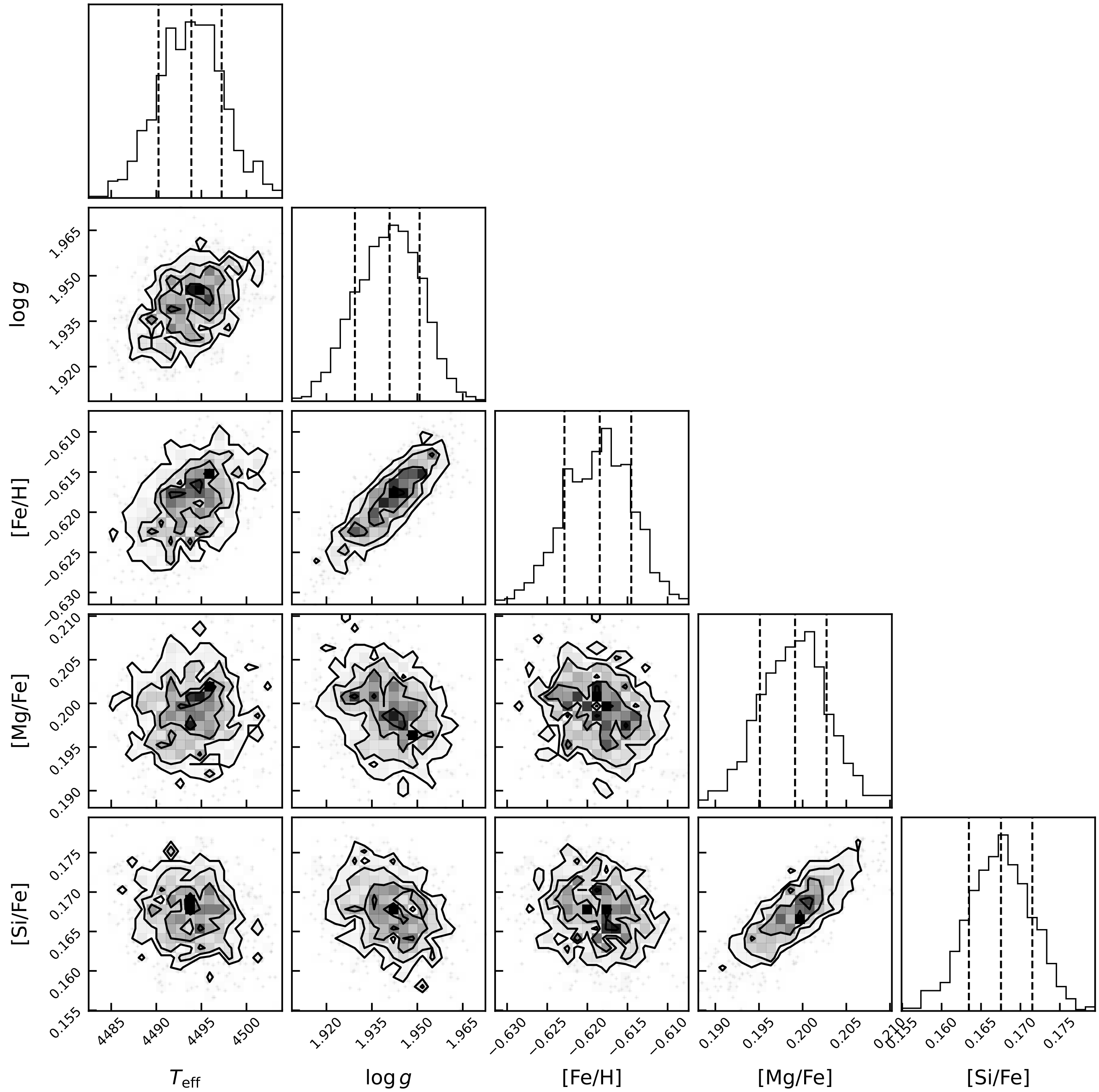

Figure 12. A corner plot showing the distribution of predicted spectrum pixels in label space. The histograms shows the label distribution and the number of pixels corresponding to each label value. The dashed lines in the label histograms indicate the lower 16th and upper 84th quantiles. The means and standard deviation in each label are shown in table 8.

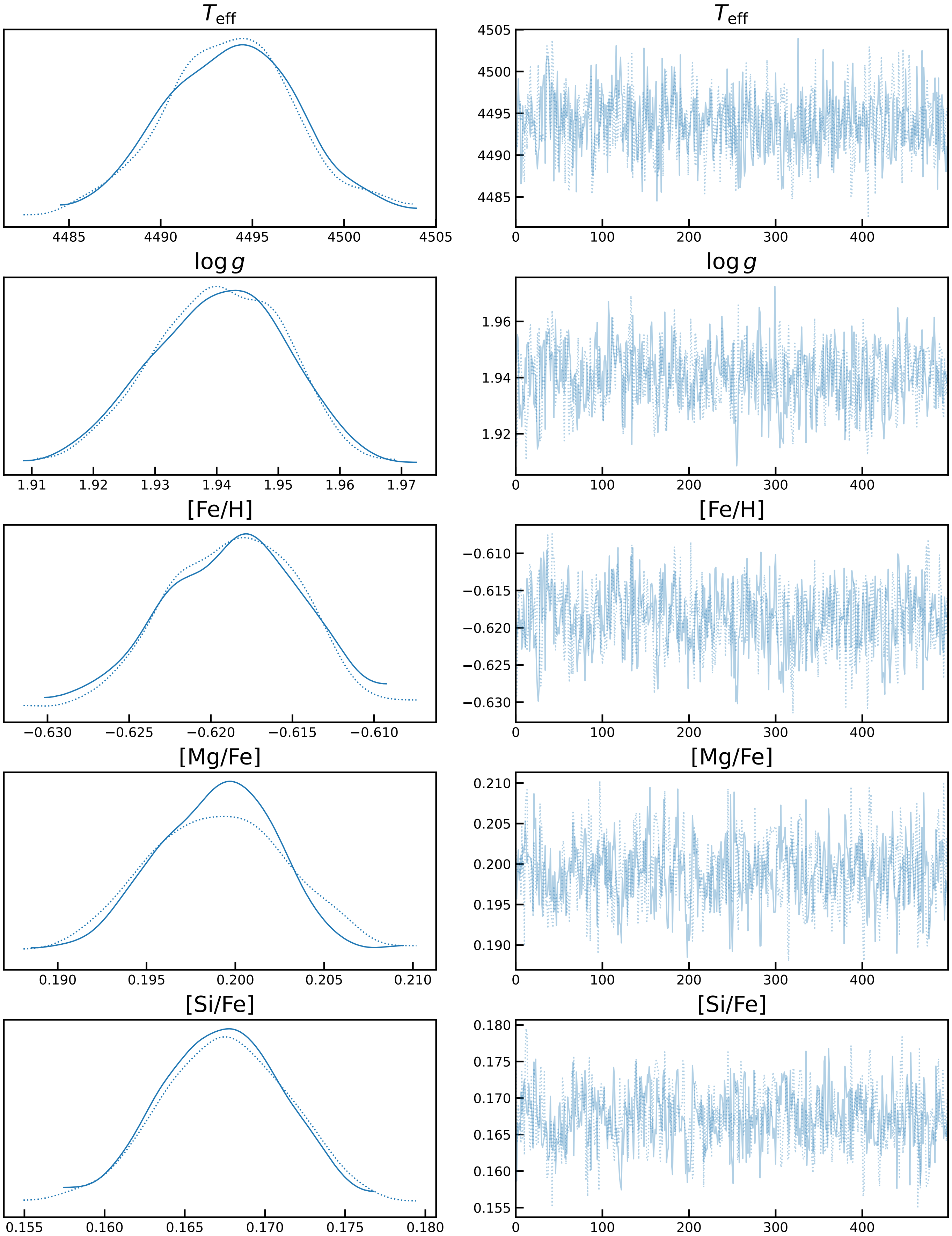

Figure 13. A traceplot for the sample of labels of the mystery spectrum, obtained from the posterior distribution. The two curves are for the two chains used in the `pymc3` model.

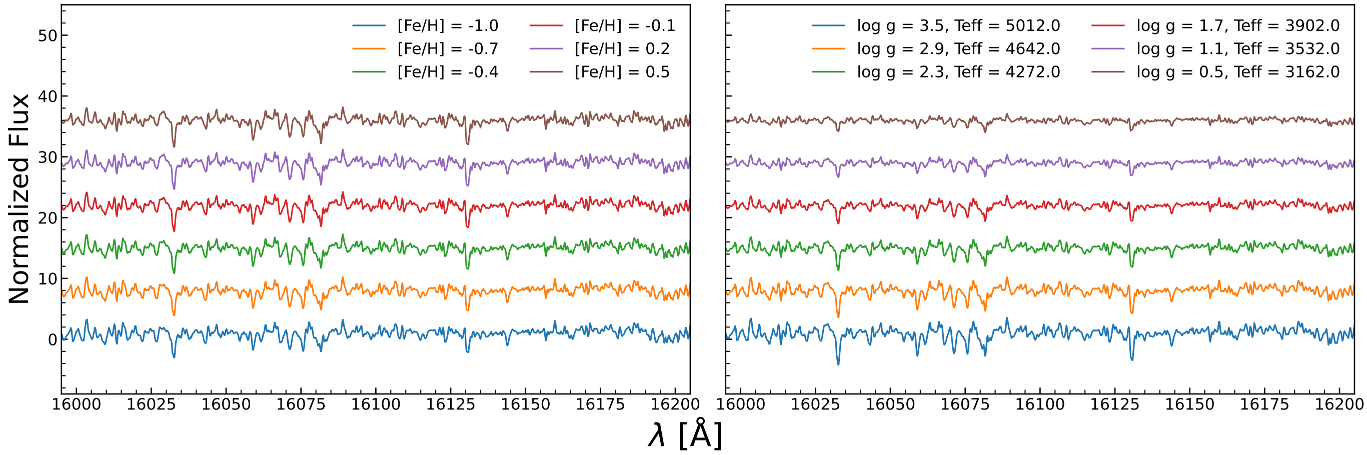

Figure 14. A color offset spectra showing how the normalized spectrum of a star changes with metallicity and as it evolves along the Red Giant Branch. The metallicity was set to vary between -1.0 and 0.5. The surface gravity $\log g$ was set to vary from 3.5 to 0.5 and the effective temperature was set to vary from $10^{3.7}$ to $10^{3.5}$, roughly corresponding to the RGB seen in the Kiel diagram from figure 10. For each color offset spectra, 6 are plotted with an offset of 7. The wavelength range was set to 16000 → 16200 Å for better visualization.

```
            # filter for bits 0-7 and 12
            mask_binary_filtered =
                mask_binary_reversed[0:8] +
                mask_binary_reversed[12:13]
            # bit(s) set to true (bad pixel)
            if ‘1’ in mask_binary_filtered:
                # inflate error
                inflated_errors[index] = 1e10
        # bit string is 1 digit
        else:
            # bit set to true (bad pixel)
            if ‘1’ in mask_binary_reversed:
                inflated_errors[index] = 1e10
```

The bits 0-7, 12, and their flags are documented in table 2. The unique bitmask integers that contain one of the bits 0-7 and/or 12 are shown in table 9. Figure 15 visualizes the distribution of bitmask integers for each wavelength bins. Figure **??** shows the flux uncertainty before and after inflating uncertainty values for bad pixels.

REFERENCES

García Pérez, A. E., Allende Prieto, C., Holtzman, J. A., et al. 2016, AJ, 151, 144, doi: 10.3847/0004-6256/151/6/144

Holtzman, J. A., Shetrone, M., Johnson, J. A., et al. 2015, AJ, 150, 148, doi: 10.1088/0004-6256/150/5/148

Majewski, S. R., Schiavon, R. P., Frinchaboy, P. M., et al. 2017, AJ, 154, 94, doi: 10.3847/1538-3881/aa784d

Ness, M., Hogg, D. W., Rix, H. W., Ho, A. Y. Q., & Zasowski, G. 2015, ApJ, 808, 16, doi: 10.1088/0004-637X/808/1/16

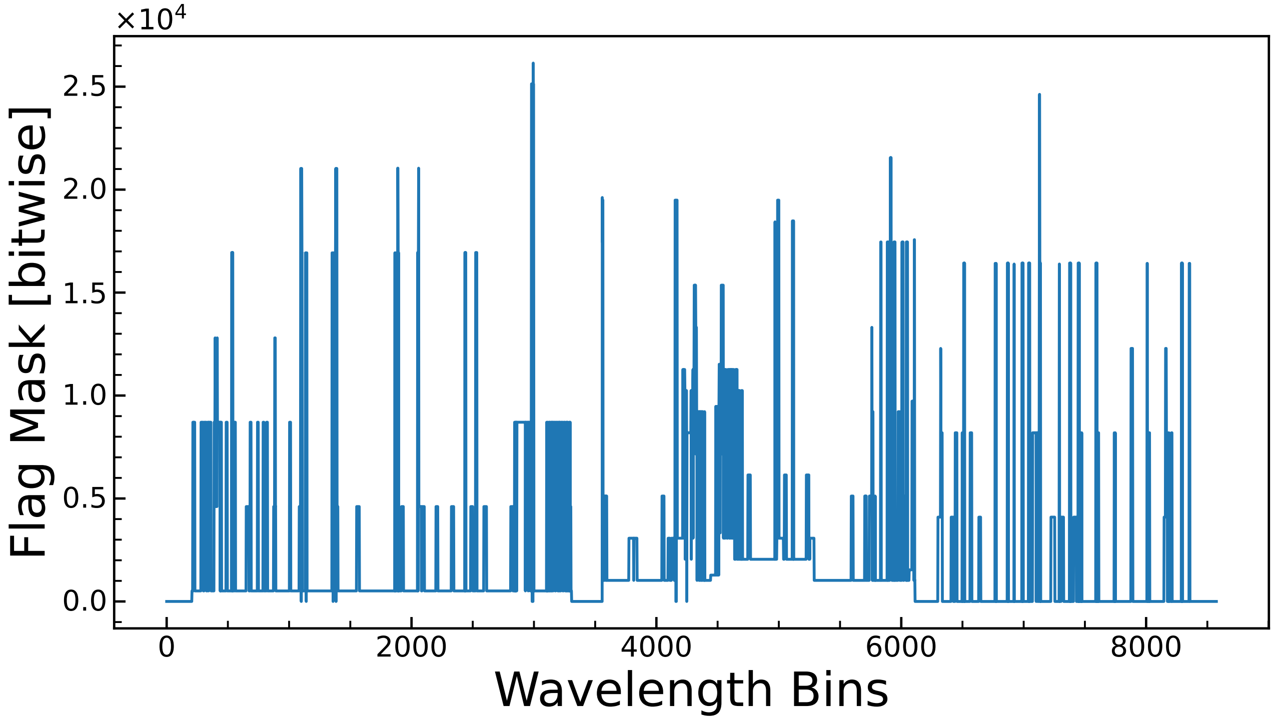

Figure 15. A plot of bitmasks (integer values) vs wavelength bins for star 2M19395986+2341280. See table 9 for details on the relevant bitmask integers.

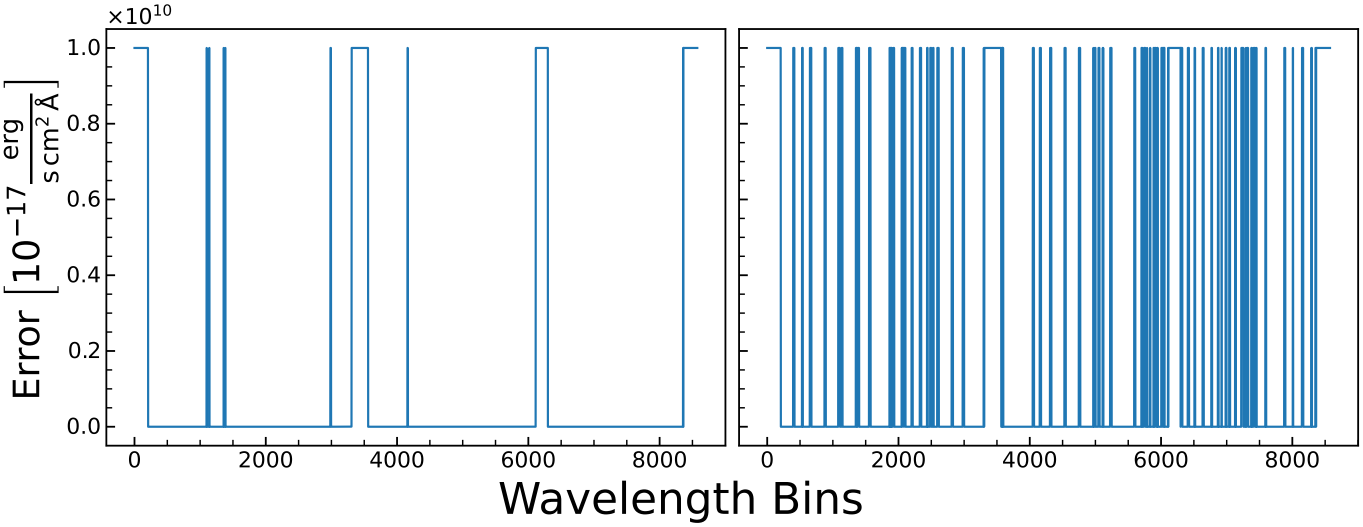

Figure 16. Plots of flux uncertainties in each wavelength bin for star 2M19395986+2341280 before (left panel) and after (right panel) setting uncertainties of bad pixels to 10^{10} erg/s/cm^2/Å.

Table 9. Bitmask integers with their binary digit representations. Presented in the tables are integers for which binary digits 0-7 and/or 12 are set to `TRUE`. These bitmask integers correspond to flagged bad pixels. Flux uncertainties corresponding to these bitmask integers are set to a large value 10^{10} so they do not contribute significantly to the likelihood function in the spectra model. Bitmask integers with binary representations containing only bits 8-11 and/or 13-16 are ignored. To see what each binary bit represent, see table 2.

Bitmask unique integers	Binary digits	Binary bit(s)
1	1	
4096	0000000000001	12
4608	0000000001001	12
5120	0000000000101	12
6144	0000000000011	12
7168	0000000000111	12
8720	00001000010001	4
12288	00000000000011	12
12800	00000000010011	12
13312	00000000001011	12
15360	00000000001111	12
16386	010000000000001	1
16417	100001000000001	0
16433	100011000000001	0, 4, 5
16435	110011000000001	0, 1, 4, 5
16929	100001000100001	0, 5
16931	110001000100001	0, 1, 5
16945	100011000100001	0, 4, 5
17441	100001000010001	0, 5
17457	100011000010001	0, 4, 5
17569	100001010010001	0, 5, 7
17953	100001000110001	0, 5
18434	010000000001001	1
18481	100011000001001	0, 4, 5
19489	100001000011001	0, 5
19617	100001010011001	0, 5, 7
21025	100001000100101	0, 5, 12
21027	110001000100101	0, 1, 5, 12
21041	100011000100101	0, 4, 5, 12
21553	100011000010101	0, 4, 5, 12
24625	100011000000011	0, 4, 5
25121	100001000100011	0, 5
25137	100011000100011	0, 4, 5
26145	100001000110011	0, 5