MNRAS **000**, 1–8 (2020) Preprint 30 August 2020 Compiled using MNRAS LATEX style file v3.0

# Modelling stars with Gaussian Process Regression – I: Augmenting Stellar Model Grid

Tanda Li,[1]★ Guy R. Davies,[1]† Alex Lyttle,[1] Lindsey Carboneau,[1] and A. N. Others[1]
[1] *School of Physics and Astronomy, University of Birmingham, Birmingham, B15 2TT, United Kingdom*

Accepted XXX. Received YYY; in original form ZZZ

**ABSTRACT**
Grid-based modelling is widely used for estimating stellar parameters. However, theoretical model grid is sparse. This paper demonstrates an application of Gaussian Process Regression (GPR), which transfer a sparse model grid to a continuing function. We trained a GPR model with a model grid to learn the function from 5 independent fundamental inputs (Mass, fractional age, initial metallicity, initial helium fraction, and the mixing-length parameter) to 6 model outputs (age, seismic large separation, effective temperature, surface gravity, radius, and surface metallicity). The GPR function was then validated and the typical standard deviations of GPR predictions are 1% for the age, 0.5$\mu$Hz for the seismic large separation, 12K for the effective temperature, 0.004dex for the surface gravity, 0.008$R_\odot$ for the radius, and 0.008dex for the surface metallicity. This indicates that GPR can be used to augmenting stellar grids and furthered model stars. We lastly applied the GPR function for modelling the Sun-as-a-star. The GPR function gives nicer statistical estimates than grid-based modelling does.

**Key words:** keyword1 – keyword2 – keyword3

## 1 INTRODUCTION

This will be the intro. (ask Guy to write some general idea of Gaussian Process and the GPR model).

## 2 THEORETICAL MODEL GRID

We built up a model grid as the training data for GPR models. We aimed to cover stars with approximate solar mass on the main-sequence and the subgiant phases. Thus, the mass range is set up as 0.8 – 1.2$M_\odot$, and we computed each stellar evolutionary track from the Hayashi line and terminated at the base of red-giant branch where $\log g$ = 3.6 dex. The gird includes four independent model inputs: stellar mass (M), initial helium fraction ($Y_{\rm init}$), initial metallicity ([Fe/H]), and the mixing-length parameter ($\alpha_{\rm MLT}$). Details of the computations are summarised in Table 1. Input parameters of our primary training data is uniformly spaced except the input metallicity, which is double dense above 0.2 *dex*. We also computed an extra grid that fit in between the primary training data for $M > 1.05M_\odot$. Because evolutionary tracks in this mass range have the so-called 'hook' at the late main-sequence stage and relatively difficult to model. Lastly, we computed 4,880 tracks with input parameters that are randomly sampled in the grid ranges for validating GPR models. The evolution time step was mainly controlled by the set-up tolerances on changes in surface effective temperature and luminosity. We also saved structural models for computing theoretical oscillation models.

### 2.1 Stellar models and input physics

We used Modules for Experiments in Stellar Astrophysics (MESA, version 12115) to establish a grid of stellar models. MESA is an open-source stellar evolution package which is undergoing active development. Descriptions of input physics and numerical methods can be found in Paxton et al. (2011, 2013, 2015). We adopted the solar chemical mixture $[(Z/X)_\odot = 0.0181]$ provided by Asplund et al. (2009). The initial chemical composition was calculated by:

$$\log(Z_{\rm init}/X_{\rm init}) = \log(Z/X)_\odot + [{\rm Fe/H}]. \quad (1)$$

We used the MESA $\rho - T$ tables based on the 2005 update of OPAL EOS tables (Rogers & Nayfonov 2002) and OPAL opacity supplemented by low-temperature opacity (Ferguson et al. 2005). The MESA 'simple' photosphere were used as the set of boundary conditions for modelling the atmosphere. The mixing-length theory of convection was implemented, where $\alpha_{\rm MLT} = \ell_{\rm MLT}/H_p$ is the mixing-length parameter. We also applied the MESA predictive mixing scheme (Paxton et al. 2018, 2019) in the model computation. The MESA inlist used for the computation is available on `https://github.com/litanda/mesa_inlist`.

★ E-mail: t.li.2@bham.ac.uk
† E-mail: G.R.Davies@bham.ac.uk

**Table 1.** Stellar model computations for training and validating sets.

| Primary Grid | | |
|---|---|---|
| Input Parameter | Range | Increment |
| $M$ [$M_\odot$] | 0.80 – 1.20 | 0.01 |
| [Fe/H] [dex] | -0.5 – 0.2/0.2 – 0.5 | 0.1/0.05 |
| $Y_{\rm init}$ | 0.24 – 0.32 | 0.02 |
| $\alpha_{\rm MLT}$ | 1.7 – 2.5 | 0.2 |
| Extra Grid | | |
| Input Parameter | Range | Increment |
| $M$ [$M_\odot$] | 1.055 – 1.195 | 0.01 |
| [Fe/H] [dex] | 0.25 – 0.45 | 0.1 |
| $Y_{\rm init}$ | 0.25 – 0.31 | 0.02 |
| $\alpha_{\rm MLT}$ | 1.8– 2.4 | 0.2 |
| Off-grid Models | | |
| 4880 tracks with random input parameters | | |

### 2.2 Oscillation models and seismic $\Delta\nu$

Theoretical stellar oscillations were calculated with the GYRE code (version 5.1), which was developed by Townsend & Teitler (2013). And we computed radial modes (for $\ell = 0$) by solving the adiabatic stellar pulsation equations with the structural models generated by MESA. We computed a seismic large separation($\Delta\nu$) for each model with theoretical radial modes to avoid the systematic offset of the scaling relation. We derived $\Delta\nu$ with the approach given by White et al. (2011), which is a weighted least-squares fit to the radial frequencies as a function of $n$.

## 3 TRAINING GPR MODELS

We adopted the GPy package (GPy 2012), which is a Gaussian Process framework developed by the Sheffield machine learning group, to train GPR models for the MESA Grid.

### 3.1 GPR kernels

The analysing is mainly based on the MLP kernel because of the following points (ask Guy to write some words about the MLP kernel). We also use other couple kernels to improve the GPR model, particularly for the residuals of the MLP-kernel model. The kernels involved in this works are listed below:

- MLP: Multi Layer Perceptron kernel (also known as arc sine kernel or neural network kernel)
- RBF: Radial Basis Function kernel (also known as squared exponential kernel)
- RQ: Rational Quadratic Kernel (equivalent to adding together many RBF kernels with different lengthscales)
- EXP: Exponential Kernel
- Mat32: Matern 3/2 kernel

### 3.2 GPR Model Inputs

We aim to derive stellar parameters based on fundamental inputs of the model grid. As mentioned in Section 2, our model grid has five independent fundamental inputs, says, mass, initial metallicity, initial helium fraction, mixing-length parameter, and the age. GPR model inputs are ideally in fixed dynamic ranges (form a cube-like space), however, the age ranges significantly vary with the mass. We hence defined an age index as an input instead of the age. The age index is calculated on each evolutionary track following

$$\tau' = 10^{(\tau/\tau_{\rm max})}, \tag{2}$$

where $\tau$ is the age, $\tau_{\rm max}$ is the maximum age of an evolutaionary track. Note that $\tau_{\rm max}$ must be defined in the same way for all evolutionary tracks. In this work, we defined $\tau_{\rm max}$ as the age when a track evolves to $\log g = 3.6$. The purpose to use an exponential formula is flattening sharp changes of global parameters around the 'hook' and the turn-off point. This makes GPR models converge easily. A comparison between GPR models using $\tau/\tau_{\rm max}$ and $10^{(\tau/\tau_{\rm max})}$ is illustrated in Figure 1. As it can be seen that, the effective temperature evolution presents a hump around 0.7 of the total lifetime (the hook) and quickly drops in the last 10% lifetime. Using the age index instead of fractional age significantly smoothes the curve and make it easer to fit. We used the same kernel (MLP) and optimiser to fit both curves and compared residuals in the bottom panel. As it can be seen that, the GPR model using the age index showes better agreement with the original data especially for the hump (~6100K).

We summary our selections of GPR model inputs and outputs as below.

- GPR model inputs and their dynamic ranges:
  Mass ($M = 0.8 – 1.2 M_\odot$)
  Age index ($\tau' = 0.0 – 10$)
  Initial metallicity ($[{\rm Z/X}]_{\rm init} = 0.0015 – 0.0572$)
  Initial helium fraction ($Y_{\rm init} = 0.24 – 0.32$)
  Mixing-length parameter ($\alpha_{\rm MLT} = 1.7 – 2.5$)
- GPR model outputs:
  Effective temperature ($T_{\rm eff}$)
  Surface gravity ($\log g$)
  Radius ($R$)
  Two global seismic parameters ($\Delta\nu$ and $\nu_{\rm max}$)
  Surface metallicity ($[{\rm Z/X}]_{\rm surf}$)
  Age ($\tau$)

Thus, our GPR model can be described as

$$\text{Outputs} = f(M, \tau', ({\rm Fe/H})_{\rm init}, Y_{\rm init}, \alpha_{\rm MLT}). \tag{3}$$

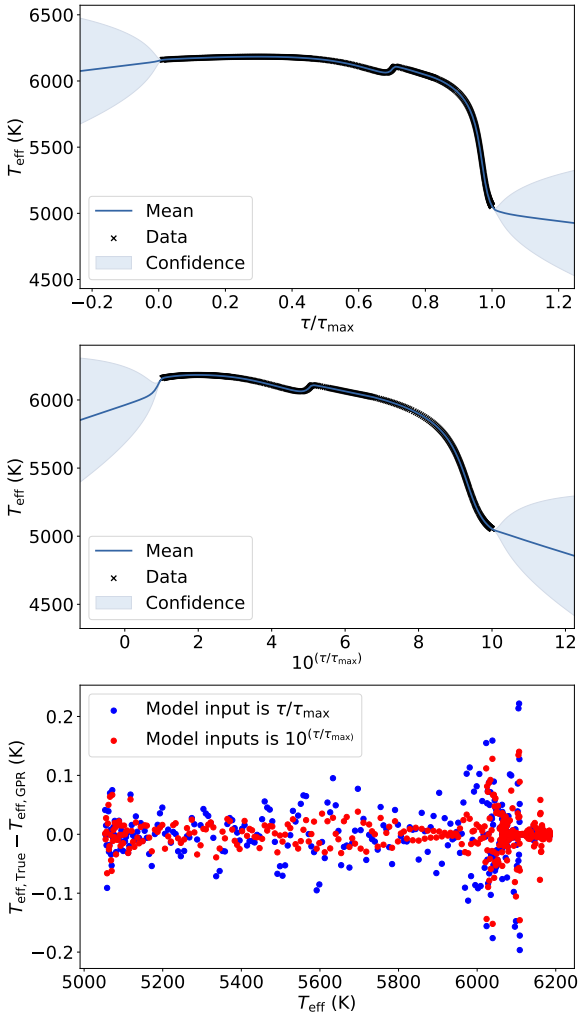


**Figure 1.** The comparison between GPR model predictions with two different input age indices ($\tau/\tau_{\rm max}$ and $10^{(\tau/\tau_{\rm max})}$) for a 1.1$M_\odot$ track. Top: the GPR model of the effective temperature as a function of $\tau/\tau_{\rm max}$. The adopted kernel is MLP. Middle: same as the top, but the GPR model input is $10^{(\tau/\tau_{\rm max})}$). Bottom: residuals between true values and GPR model predictions.

### 3.3 Training Procdure

To demonstrate our work flow, we present our training process of a 2-demissions GPR model. The GPR model contents two inputs, namely, $M$ and $\tau'$, and one output ($T_{\rm eff}$). We only adopted the stellar models with fixed [Fe/H]$_{\rm init}$ (0.0 dex), $Y_{\rm init}$ (0.28), and $\alpha_{\rm MLT}$ (2.1) from the primary grid. The total number of stellar models is 24,485. A validation data set was computed with the same model inputs but random masses. Our training procedure contents three major steps: data selection, model training, and model validation.

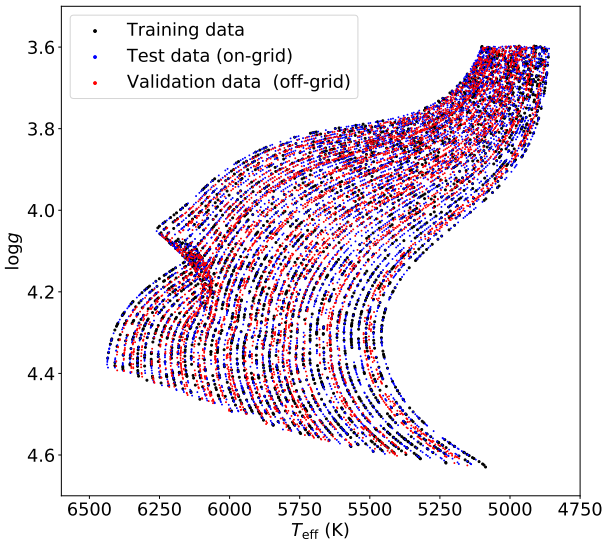


**Figure 2.** Selected training and testing data for the GPR model with 2-demission inputs on the $T_{\rm eff}$ - log $g$. Black, blue, and red dots represent training, test, and validation data.

#### 3.3.1 *Step1: Data Selection*

As discussed in Section **??**, we have three types of data for training, testing, and validating GPR models. The training data is apparently the model grid used to train GPR models. The test dataset, which was also selected from the model grid but not used in any training processes. Test data are used to examine wether a GPR model gives proper description for the model grid. Once a GPR model provides good agreement with the testing data, the training succeeds. The last type is validation data, which are the off-grid models with random input parameters. Validation data are used to validate GPR models and also to estimate their systematical uncertainty.

Because the computational and memory complexity exponentially increase with the number of data involved in the Gaussian Process. A practical limit of the training data is about 10,000. The testing and validation data could be massive but we set a limit of 50,000 for the efficiency. This is to say, only a small subset of the model grid can be used. The sampling method is hence critical. The training data should uniformly cover the output ranges to avoid data gaps but also highly weight where quick changes occurs. Due to the MESA step-control strategy, stellar models are dense at the main-sequence and the red-giant phases but quite sparse on subgiant stage. Random sampling is hence not appropriate. We tested a few methods and found that using the displacement of each evolving step on the HR diagram as the weight gave the best sampling. The displacement is defined as $\delta d = (\delta T_{\rm eff}^2 + \delta L^2)^{1/2}$, where $\delta T_{\rm eff}$ and $\delta L^2$ are the differences between two consecutive models on the evolutionary track. We sampled on each evolutionary track separately and then combined the models as datasets. The sampling method gave a relatively uniform data distribution on the HR diagram and also highly weighted the regions around the hook and turn-off point. We demonstrated our selection for data to train the 2-demission GPR model ($T_{\rm eff} = f(M, t')$) in Figure 2. Here we sampled 3,000 and 5,000 on-grid models as the training and test data, and 5,000 off-grid models as validation data.

### 3.3.2 Step 2: Model Training

In the model training step, we applied a data-residual process, which is described as follow. A GPR model (M0 here after) is primarily trained with the training data and tested with the test data. If residuals of test data are significant large or present any local structures, an extra GPR model (M1 hereafter) will be trained to describe the first-order residuals. With M0 + M1, the second-order residual could be calculated and used to train another extra GPR model (M2 hereafter). The training process goes to $n$-order residuals (M$n$) until GPR predictions do not significantly improve. Thus, the process provides a series of GPR models as a combination to describe stellar evolutions. Here we illustrated the details the process by training the 2-demission GPR model ($T_{\rm eff} = f(M, t')$).

We firstly train the primary GPR model (M0) with the training dataset shown in Figure 2. We used the MLP kernel and optimise M0 with random initial hyperparameters. The result of M0 is demonstrated in Figure 3. It can be seen that the GPR model has transferred the model grid into a continued function (indicated by contour lines). For models below $\sim 1.05 M_{\odot}$, the $T_{\rm eff}$ contour lines are smooth and the GPR model give good description for the data. However, the structure becomes complex at $t' \sim 5$ when $M$ is larger than 1.05 because of the appearance of the 'hook' in stellar evolution. And the GPR model predictions does not very convinced in this area. This can be seen in the left bottom panel in Figure 4, which shows the offsets between M0 predictions and the test data (testing errors hereafter). The large offset in the local area indicate M0 does not well describe the whole grid. This is because we only used a small subset for training and did not pass enough details for the GPR model to learn. It is a common issue with the Gaussian process when training big data set because of the limitation of the size of training data. To mange this issue, we developed the data-residual process aiming to use multiple GPR models to describe the grid. This process is illustrated in Figure 4. After training and testing M0, we trained an extra GPR model (M1) to describe the 1st-order residuals. We randomly selected 20,000 on-grid models, used M0 to predict their effective temperature, and calculated 1st-order residuals. We then sampled 3,000 training data from these 20,000 models weighting by their absolute residual values, saying that models with large offsets were highly-weighted. The selected training data for M1 is presented in the top middle graph. As it can be seen that the area where M0 does not well describe are highlighted with more dense data points than other regions. The training of residuals are operated with a kernel competition because we can predict which kernel would work. We trained 5 GPR models with MLP, RBF, RQ, EXP, and Mat32 kernels, combined each of them with M0, and computed testing errors. Because M1 mainly focuses on improving the poor prediction in small regions (a few percent of the dataset). We quantified absolute testing errors with two cumulative values at 95%, and 99.8%. The 99.8% cumulative value was firstly compared to decide the best kernel. If there was no obvious differences (< 5%), the 95% value would be used. In the presented example, the MLP kernel won and the corresponding model was selected as M1. We presented the testing errors for M0+M1 in the bottom middle panel. Those large offsets significantly reduce and the error distribution turns to be random. With the same approach for M1, we trained M2 for the 2nd-order residuals and tested the model combination of M0+M1+M2. However, the improvement of testing errors (bottom right) was very small (<5% for both 95% and 99.8% cumulative values), and we hence terminated our training process. Here we obtained a GPR model combination to describe $T_{\rm eff}$ as a function of $M$ and $\tau$;.

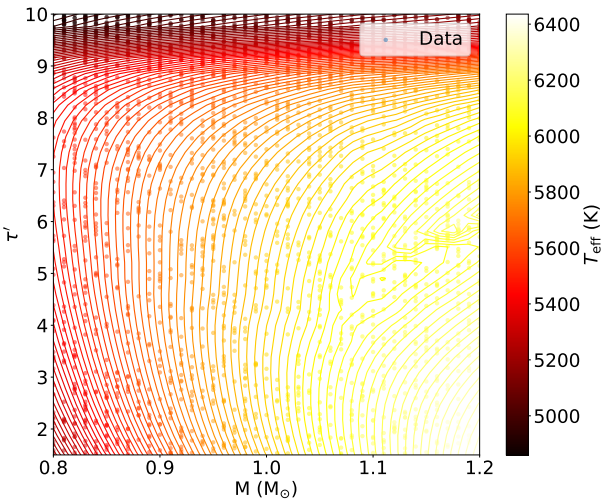


**Figure 3.** Top: The GPR model (using the MLP kernel) for the effective temperature ($T_{\rm eff}$) as a function of mass ($M$) and age index ($t'$). Dots represent the training data and counters indicate the GPR predictions.

As demonstrated above, the data-residual process is able to break through the limitation of training data by separating data features into the general and the local levels. In this example, although we set up a limitation of 3,000, the actual data points involved in the training process was around 7,000 (considering the overlapping of three GPR models). The other advantage of this process is offering flexibilities to manage different evolving features of outputs. For instance, quickly changes appear around the hook for the effective temperature but at the subgiant phase for the surface metallicity. The data-residual process allows the training to focus on different local areas in the parameter space for different outputs when dealing with residuals.

### 3.3.3 Step 3: Model Validation

We lastly validated the GPR models by using the offsets between GPR predictions and the validation data (validation errors here after). We quantified absolute validation errors with three cumulative values at 50%, 95%, and 99.8%. The first two are used to validate GPR models in general level. The last value is used to validate those evolutionary phases where GPR models do not work well. Because a large deviation between the 95% and 99.8% cumulative values often means a long tail of probability distribution, which indicates that GPR models poorly describe some local areas. In Figure 5, we demonstrated the validation errors for the model combination obtained above. As It can be seen that, the GPR models well predict $T_{\rm eff}$ in most area, however, not very well match those around M$\sim 1.1 M_{\odot}$ and $\tau' \sim 5$ (the main-sequence the hook). Comparing with the testing errors in Figure 4, the validation errors indicate that although the GPR models were well trained to describe the model grid in this area, they still do not learn the feature between the grid. The reason is that $\sim 1.1 M_{\odot}$ the boundary between tracks with and without the 'hook'. The local structure is hence more complex than any other areas (as it can be seen in Figure 3) and it leads to relatively poor predictions. Because the distribution of validation errors in the input space is not flat, it is difficult to measure a global sys-

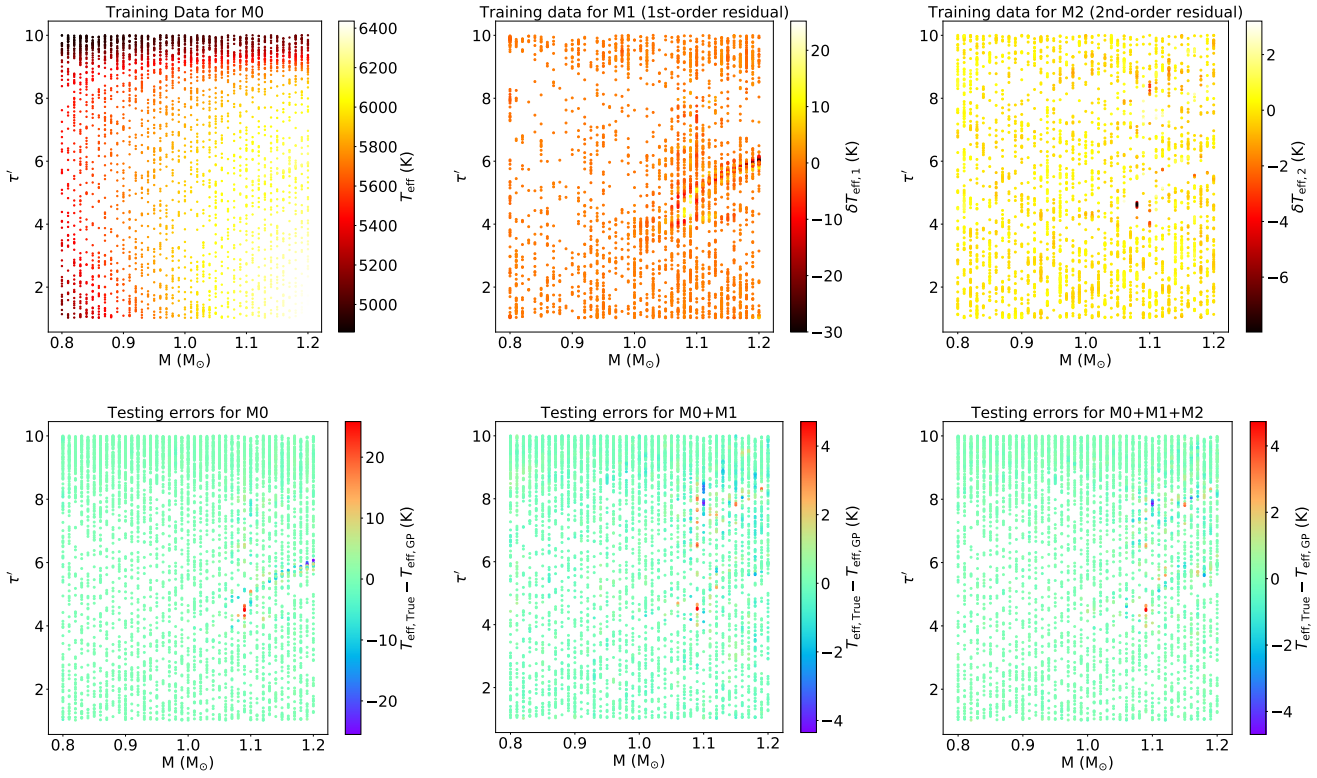


**Figure 4.** The data-residual process for training the 2-demission GPR models for the effective temperature. The top row (from left to right) shows the training data used to train the data (M0), the first-order residuals(M1), and the second-order residuals(M2). The bottom row (from left to right) demonstrates the reduction of testing errors along the training process.

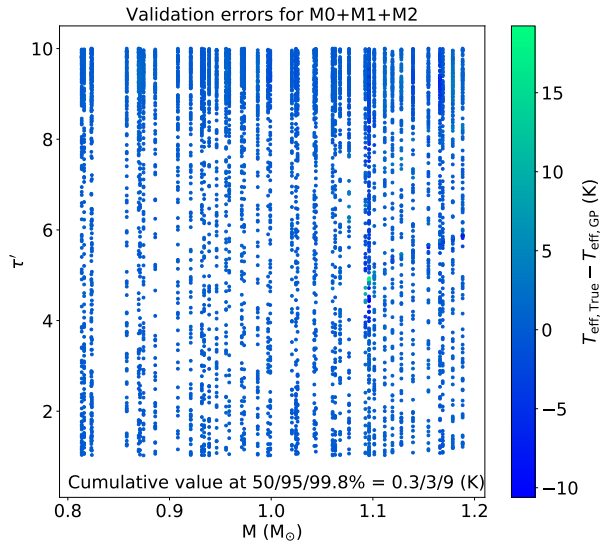


**Figure 5.** Validation errors of $T_{\rm eff}$ on the $M-\tau'$ diagram for the 2-demission GPR models ($T_{\rm eff} = f\,(M, \tau')$).

tematical uncertainty. We will discuss how we applied systematical uncertainty when fitting stars in Section 4.

### 3.3.4 *Summary of the Training Procedure*

Here we summarised our training procedure. We firstly selected three types of data for training, testing, and validating GPR models. The sampling method of training data are different for the outputs depending on their evolving features. The test and validation data were sampled uniformly on the HR diagram to equally cover all evolutionary stages. We then trained GPR models with given inputs and outputs following a data-residual process. A primary model is trained to describe the general feature of the grid, and some extra models are then trained to reveal local structures. A combination of these GPR models is used to describe the MESA grid. Model validation is lastly carried out to validate GPR models and also to estimate systematical uncertainties. With the training process, we trained 2-demission GPR models for $T_{\rm eff}$ and $\log g$, and used them to make a non-sparse $T_{\rm eff}$ - $\log g$ diagram as shown in Figure 6.

## 4 GPR MODELS FOR THE 5-DEMISSION MESA GRID

In this section, we demonstrated our training process for the 5-demission MESA grid.

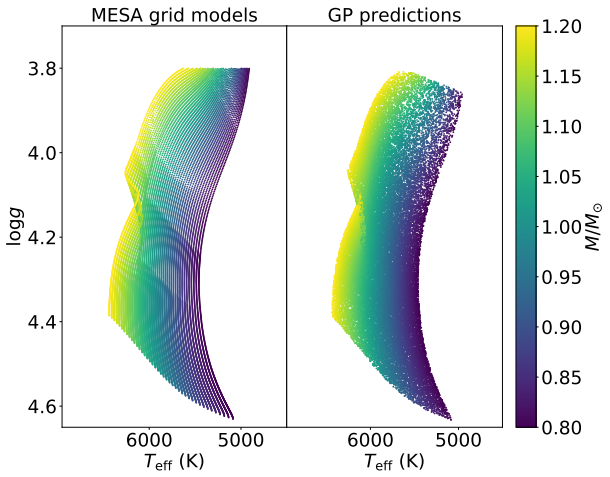


**Figure 6.** MESA grid models (sparse) and GP predictions (non-sparse) on the $T_{\rm eff}$ - log $g$ diagram. Note that we only predict models with log >= 3.8 to avoid the edge effect.

### 4.1 Training GPR Models

### 4.2 Validating GPR models and Estimating Systematical Uncertainties

### 4.3 Modelling with GPR Models

#### *4.3.1 Three Fake Stars*

#### *4.3.2 The Sun*

#### *4.3.3 Six* Kepler *Dwarfs and Subgiants*

- Accuracy goes down. Because the multiple-demission space size increase while the training dataset has a limitation of 10,000 points.
- We hence need to divide the grid into small chunks and train each chunk separately. (illustrate how chunks are divided)
- show training and validation results in a fancy way.

## 5 DISCUSSION

Do some disccusion.

## 6 CONCLUSIONS

Restate the main results of the paper.

## ACKNOWLEDGEMENTS

The Acknowledgements section is not numbered. Here you can thank helpful colleagues, acknowledge funding agencies, telescopes and facilities used etc. Try to keep it short.

## REFERENCES

Asplund M., Grevesse N., Sauval A. J., Scott P., 2009, Annual Review of Astronomy and Astrophysics, 47, 481
Ferguson J. W., Alexander D. R., Allard F., Barman T., Bodnarik J. G., Hauschildt P. H., Heffner-Wong A., Tamanai A., 2005, ApJ, 623, 585
GPy since 2012, GPy: A Gaussian process framework in python, `http://github.com/SheffieldML/GPy`
Paxton B., Bildsten L., Dotter A., Herwig F., Lesaffre P., Timmes F., 2011, The Astrophysical Journal Supplement Series, 192, 3
Paxton B., et al., 2013, The Astrophysical Journal Supplement Series, 208, 4
Paxton B., et al., 2015, The Astrophysical Journal Supplement Series, 220, 15
Paxton B., et al., 2018, ApJS, 234, 34
Paxton B., et al., 2019, ApJS, 243, 10
Rogers F. J., Nayfonov A., 2002, ApJ, 576, 1064
Townsend R. H. D., Teitler S. A., 2013, MNRAS, 435, 3406
White T. R., Bedding T. R., Stello D., Christensen-Dalsgaard J., Huber D., Kjeldsen H., 2011, ApJ, 743, 161

**Table 2.** Training and validating for the 5D model grid

| GPR model input | GPR model output | Model combination and kernels | Validation errors (%) |
|---|---|---|---|
| $M$, $t_{\rm frac}$, $[{\rm Fe/H}]_{\rm init}$, $Y_{\rm init}$, $\alpha_{\rm MLT}$ | $T_{\rm eff}$ (K) | M0(MLP)+M1(MLP) | 0.05/0.3/1 |
| | $\log g$ (dex) | M0(MLP)+M1(Mat32)+M2(EXP) | 0.01/0.07/0.3 |
| | $R$ ($R_\odot$) | M0(MLP)+M1(Mat32)+M2(EXP) | 0.07/0.4/2 |
| | $\Delta\nu$ ($\mu$Hz) | M0(MLP)+M1(Mat32)+M2(EXP) | 0.1/0.7/3 |
| | $(Z/X)_{\rm surf}$ | M0(MLP)+M1(Mat32)+M2(RBF) | 0.1/1/6 |
| | $\tau$ (Gyr) | M0(MLP)+M1(Mat32)+M2(RBF) | 0.1/1/14 |

## APPENDIX A: VALIDATION AND PREDICTION OF GPR MODELS

### A1 GPR model with 3-D inputs

- Training set
- validation
- GP predictions

### A2 GPR model with 4-D inputs

- Training set
- validation
- GP predictions

### A3 GPR model with 5-D inputs

- Training set
- validation
- GP predictions

This paper has been typeset from a TEX/LATEX file prepared by the author.

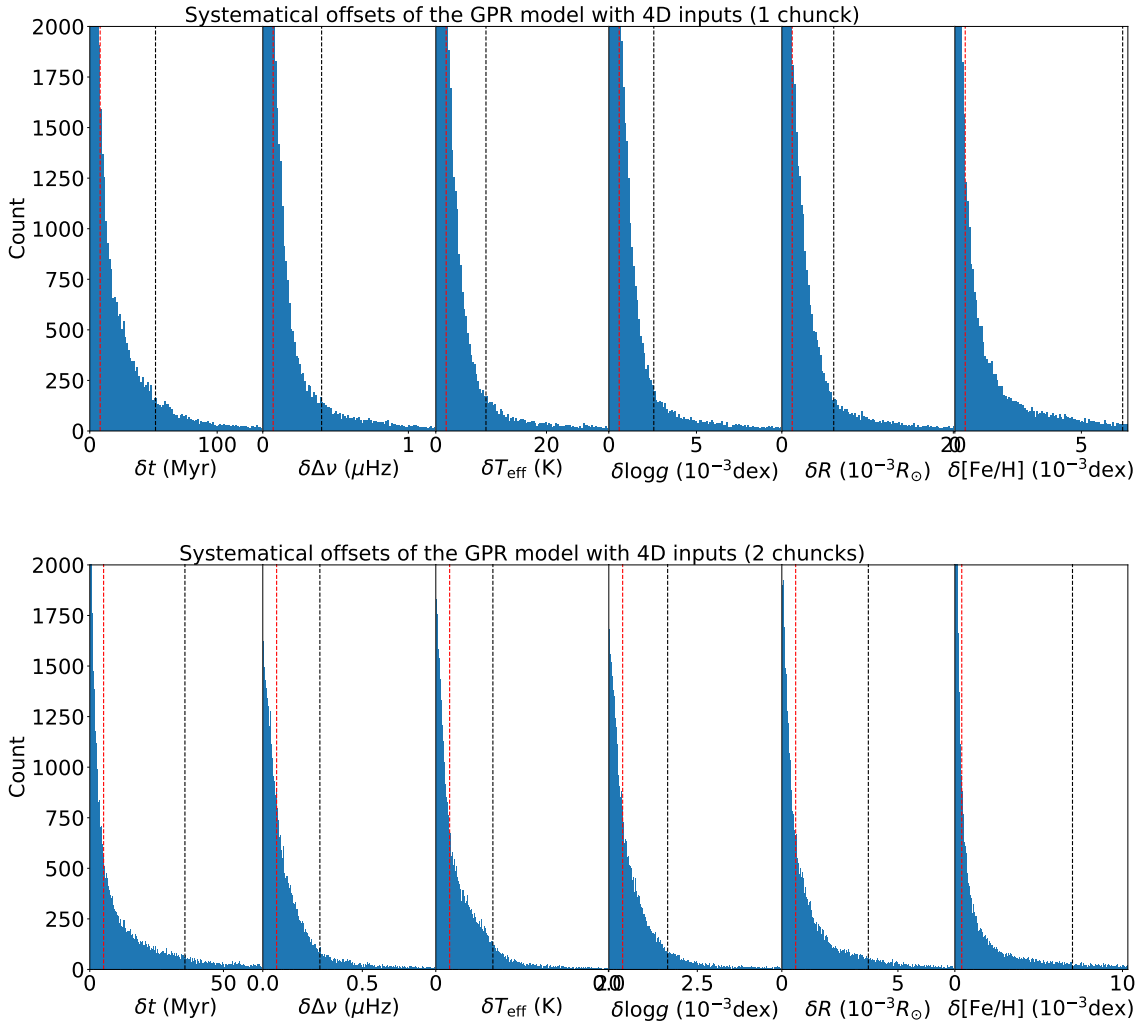


**Figure A1.** Validations for 4D inputs GPR models before and after chunking.

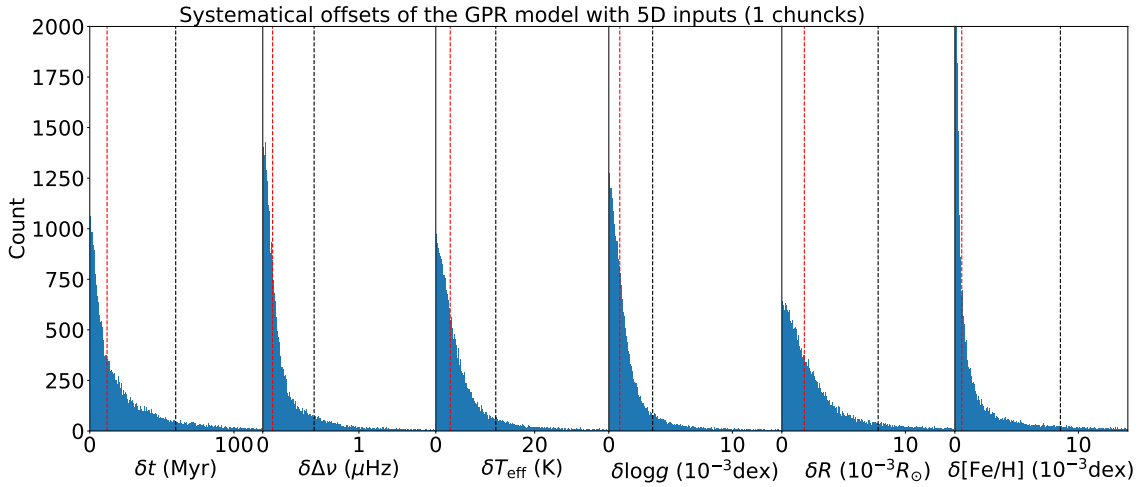


**Figure A2.** Validations for 5D inputs GPR models before and after chunking.