The Astronomical Journal, 170:72 (12pp), 2025 August https://doi.org/10.3847/1538-3881/addf51

Accelerating Stellar Photometric Distance Estimates with Neural Networks

Karlo Mrakovčić[1], Željko Ivezić[2,3], and Lovro Palaversa[3]
[1] Faculty of Physics, University of Rijeka, Radmile Matejčić 2, 51000 Rijeka, Croatia; karlo.mrakovcic@uniri.hr
[2] Department of Astronomy and the DiRAC Institute, University of Washington, 3910 15th Avenue NE, Seattle, WA 98195, USA
[3] Ruđer Bošković Institute, Bijenička cesta 54, 10000 Zagreb, Croatia
Received 2025 February 21; revised 2025 May 29; accepted 2025 May 29; published 2025 July 7

Abstract

Building on the Bayesian approach to estimating stellar distances from broadband photometry, we show that the computation can be accelerated by about an order of magnitude by using neural networks. Focusing on the case of the *ugrizy* filter complement for Rubin's Legacy Survey of Space and Time (LSST), we show that the Bayesian approach is equivalent to mapping from a 10-dimensional space of five measured colors and their uncertainties to a three-dimensional space of absolute magnitude, metallicity, and interstellar dust extinction along the line of sight. Once the neural network is trained, this mapping is faster by more than an order of magnitude compared to the Bayesian approach, for both optimized grid search and Markov chain Monte Carlo implementation methods. We have developed and tested a pipeline that achieves significant acceleration by first running the Bayesian method on 5%–10% of the sample, then using it to train a neural network, and finally processing the entire sample with the resulting neural network. This computation is done in patches of about 10 deg^2 due to the variation of Bayesian priors across the sky. We present an analysis of pipeline performance, including speed and biases as functions of input stellar parameters and signal-to-noise ratio, using TRILEGAL-based simulated LSST catalogs by P. Dal Tio et al. We intend to run this pipeline on LSST data releases and make its outputs publicly available.

Unified Astronomy Thesaurus concepts: Neural networks (1933); Distance measure (395); Distance indicators (394); Stellar distance (1595); Extinction (505); Interstellar extinction (841); Reddening law (1377)

1. Introduction

Stellar color measurements delivered by Vera C. Rubin's Legacy Survey of Space and Time (LSST; Ž. Ivezić et al. 2019) will enable photometric distance estimates for over 10 billion Milky Way stars (L. Palaversa et al. 2025). LSST-based stellar distances will reach about 10 times further than Gaia's color-based distances (C. A. L. Bailer-Jones et al. 2021) and will be transformative for studies of the Milky Way in general (e.g., M. Jurić et al. 2008; N. A. Bond et al. 2010; M. Berry et al. 2012), and of its stellar and dark matter halo in particular (e.g., S. R. Loebman et al. 2012, 2014). Distance accuracy for stars with sufficiently small LSST photometric errors will be within the 5%–10% range, or about twice as accurate as for surveys lacking the UV *u* band (which provides metallicity constraints).

L. Palaversa et al. (2025) describe a Bayesian procedure and pipeline that builds on previous work and can handle LSST-sized data sets. Their photo-D method is conceptually quite simple: "multidimensional color tracks (either empirical or model-based), parameterized by luminosity, metallicity, and extinction, are fit to observed colors and the best fit produces estimates of the three model parameters." They achieved a computation speed of about 10 ms per star on a single core for both optimized grid search and Markov chain Monte Carlo methods. Consequently, it would take about 10 days to process data for 10 billion stars on a 100-core machine. While not prohibitively expensive, it seems prudent to explore various ways to accelerate this computation.

In this paper, we explore the potential of neural networks for accelerating the Bayesian approach. The Bayesian computation is equivalent to mapping from a 10-dimensional space of five measured colors and their uncertainties to a three-dimensional model parameter space of absolute magnitude, metallicity, and interstellar dust extinction along the line of sight (discussed in detail in Section 2.2). This computation uses Bayesian priors and is done in patches of about 10 deg^2 due to the variation of Bayesian priors across the sky (see L. Palaversa et al. 2025 for details). Once the neural network is trained for such a patch, its application is faster by more than an order of magnitude compared to the Bayesian approach. We present here a pipeline that achieves significant acceleration by first running the Bayesian method on a small fraction (5%–10%) of the full sample, then using it to train a neural network, and finally processing the entire sample with the resulting trained neural network.

In Section 2, we describe the technical details of our methodology and in Section 3 we optimize and analyze its performance. Our discussion and our principal conclusions are summarized in Section 4.

2. Methods

To accelerate the Bayesian approach, we developed a custom neural network model specifically designed for this problem. The architecture of the model was decided using the hyperparameter tuning process described further below (see Section 2.5). The input to the neural network consists of a measured magnitude (chosen here in the *r* band) and colors ($u-g$, $g-r$, $r-i$, $i-z$), along with their associated observational uncertainties. Although model parameters are fundamentally constrained by measured colors, the inclusion of photometric errors is essential, as discussed further below (see Section 2.2). We use three data sets to optimize the

method and test its performance: two simulated data sets where the true values of model parameters are known, and a catalog based on the Sloan Digital Sky Survey (SDSS) for a high-extinction region in the Galactic plane, as follows.

2.1. Data

We used multiple data sets throughout this study. One simulated data set is used to perform hyperparameter tuning. Two data sets, one observed and one simulated, are used to train and evaluate the model performance.

All simulated data sets used in this study are derived from the TRILEGAL simulation (P. Dal Tio et al. 2022), which provides a realistic mock catalog of the Milky Way stars to LSST depth ($r = 27.5$) and over the entire LSST survey area. Each star in the simulated catalog is characterized by apparent magnitudes in the *ugriz* photometric bands, as well as the true values of absolute magnitude M_r, interstellar dust extinction along the line of sight A_r, and stellar metallicity [Fe/H]. For a given sky patch (with an area of $\sim$10 deg^2, which is similar to the size of the LSST Camera's field of view), the sample is binned by apparent r-band magnitude (27 bin centers from $r = 14$ to $r = 27$, and overlapping bins with 1 mag width), and prior distribution in the model parameter space is estimated for each bin.

For the hyperparameter tuning, we used a data set consisting of around 720,000 simulated stars, divided into a training set (70%), validation set (15%), and test set (15%). This allowed us to systematically evaluate different neural network architectures on a consistent data split.

For training and testing the final model, we used a separate data set consisting of around 440,000 stars. On that data set, we conducted an additional search to identify the minimal size of training set required for optimal performance. Through this process, we determined that only 10,000 stars were needed for training. These 10,000 stars were further split into a training set (75%) and a validation set (25%), with validation sets used for early stopping and learning rate scheduling. The remainder of the data set of around 430,000 stars was reserved for testing. All final results reported in this paper were evaluated on this test set.

The observed data set is based on SDSS-SEGUE catalogs assembled and discussed by[4] M. Berry et al. (2012). For testing, we selected their catalog that overlaps the Galactic plane between longitudes $l = 110°$ and $l = 130°$, with 150,000 stars.

2.2. Why Neural Networks Need Photometric Errors

Stellar model parameters are fundamentally constrained by measured colors. A restricted case of blue stars where two model parameters—effective temperature and metallicity—are constrained in a two-dimensional $g - r$ versus $u - g$ color–color space is visualized in Figure 2 from Ž. Ivezić et al. (2008). It thus may be somewhat surprising that photometric errors play a role too in estimating model parameters. While photometric errors are not too important in a regime of high signal-to-noise ratio (SNR), they are essential in one of low SNR and are needed to avoid bias in estimated model parameters.

[4] Their catalogs are publicly available for download from http://faculty.washington.edu/ivezic/sdss/catalogs/tomoIV/ws.html.

The impact of measurement errors is incorporated naturally in the Bayesian approach, both directly through the likelihood function and indirectly through the role of priors (see Section 2.1 of L. Palaversa et al. 2025). The Appendix provides a toy model and detailed analysis of the impact of photometric errors, particularly in the $u - g$ color, on the accuracy of metallicity estimates. This example highlights the necessity of incorporating the measurement uncertainties into the neural network's input: instead of a naive expectation that the input space is spanned by five measured colors, it contains their five measurement uncertainties, too. As mentioned earlier, the primary role of the r-band magnitude is the parameterization of priors (see also discussion in Section 2.1 of L. Palaversa et al. 2025).

Therefore, neural networks must learn how to incorporate the measurement uncertainties into their predictions. By incorporating them, the network can learn to adjust its predictions based on the level of uncertainty in the input. The network will be trained to minimize not only the difference between the predicted and true values of model parameters but also that between the predicted and true (input) values of their uncertainties (as computed by the Bayesian approach for the training sample).

2.3. Neural Network Model

The neural network (NN) model used here is based on a fully connected feedforward network: the first layer serves as an input layer, and is followed by several hidden layers, ending with the output layer. The model is structured to take as input both the photometric measurements (r, $u - g$, $g - r$, $r - i$, and $i - z$) and their associated errors ($r_{\rm err}$, $u - g_{\rm err}$, $g - r_{\rm err}$, $r - i_{\rm err}$, and $i - z_{\rm err}$). It is trained to predict the absolute magnitude M_r, extinction A_r, and metallicity [Fe/H], along with their uncertainties.

Our neural network uses the moments network approach (N. Jeffrey & B. D. Wandelt 2020), where the output for each parameter is not a single value but two numbers: the mean (μ_i) and the standard deviation (σ_i) of the predicted parameter's posterior distribution. This probabilistic output allows the model to provide an estimate of the uncertainty of its predictions, which is crucial for estimating standard deviation.

2.4. Loss Function

The loss function used during training is designed to minimize the squared error between the predicted values and the true values while taking into account the predicted uncertainty. The form of the loss function is

$$\mathcal{L} = \frac{1}{N}\sum_{i=1}^{N}((\theta_i - \mu_i)^2 + ((\theta_i - \mu_i)^2 - \sigma_i^2)^2) \quad (1)$$

where θ_i are the true values of the parameters, and μ_i and σ_i are the predicted mean and standard deviation, respectively.

We train the network to predict all parameters (M_r, A_r, [Fe/H]) simultaneously, using a weighted sum of the loss functions described above for each parameter:

$$\mathcal{L} = \frac{1}{0.1^2}\mathcal{L}_{M_r} + \frac{1}{0.02^2}\mathcal{L}_{A_r} + \frac{1}{0.1^2}\mathcal{L}_{\rm Fe/H} \quad (2)$$

where $\mathcal{L}_{M_r}$, $\mathcal{L}_{A_r}$, and $\mathcal{L}_{\rm Fe/H}$ are the individual loss functions for each parameter. Values of 0.1 mag, 0.02 mag, and 0.1 dex are typical minimal values for Mahalanobis distance, based on

photometric accuracy of around 0.01–0.02 mag, as shown in previous work (Ž. Ivezić et al. 2008). These weights ensure that each parameter's contribution to the total loss is appropriately scaled relative to its typical uncertainty.

The predicted uncertainties (σ_i) play a critical role in ensuring that the model provides not only accurate predictions but also appropriate error estimates, which are essential for understanding the confidence in the distance estimates.

2.5. Hyperparameter Tuning

Hyperparameter tuning refers to the process of optimizing key parameters of the neural network, such as the numbers of layers, neurons per layer, and activation functions. This process is typically computationally expensive and requires significant resources, which is why it was performed on the HPC Bura at the University of Rijeka. All jobs were run on its CPU cluster, with each job utilizing two CPU cores and 4 GB of RAM. It is a one-time process to find the best-performing network architecture. Once determined, this architecture is used for all subsequent stellar parameter prediction models without the need for any future hyperparameter adjustments.

During the hyperparameter tuning, a hyperparameter data set of around 720,000 simulated stars was used, and separated into training, validation, and test sets as described in Section 2.1. The training set was used to optimize model weights for each candidate hyperparameter configuration, the validation set was used to monitor the training progress, while the test set was used to select the best-performing models according to their loss.

For this experiment, hyperparameter tuning was carried out using the Hyperband algorithm (L. Li et al. 2018), an approach that allows for efficient exploration of a vast hyperparameter space. The algorithm operates by training many networks for a small number of epochs and gradually eliminating poorer-performing models, similar to a knockout tournament. The final models are then trained for up to 1024 epochs, with the best-performing models being evaluated based on mean square error (MSE) metrics.

A total of 45 runs of the Hyperband algorithm were conducted, with the "winner of winners" being selected from the last 45 finalists. In total, 34,740 models were compared per job, with each job taking approximately 7 days to complete, running in parallel.

The hyperparameters varied in the tuning included:

1. Total number of hidden layers: minimum 1, maximum 10.
2. Number of neurons in each hidden layer: minimum 1, maximum 50.
3. Activation function for each hidden layer: a choice from ELU (D.-A. Clevert et al. 2015), GELU (D. Hendrycks & K. Gimpel 2016), Hard Sigmoid, Linear, ReLU (R. H. R. Hahnloser et al. 2000; V. Nair & G. E. Hinton 2010), SELU (G. Klambauer et al. 2017), Sigmoid, Softmax (C. Dugas et al. 2000), Softplus (V. Nair & G. E. Hinton 2010), Softsign (D. Macêdo et al. 2019), Swish (P. Ramachandran et al. 2017), and Tanh.

Each model was initialized with a learning rate $\alpha = 0.01$. The learning rate was reduced by a factor of 0.9 after the loss metric stopped improving, following a "Reduce learning rate on Plateau" strategy (D. Wilson & T. Martinez 2001). This iterative approach allowed the algorithm to search as large a portion of the parameter space as possible (around 1.5×10^6 searched out of 8×10^{24} possible combinations), ultimately identifying the best-performing architecture.

The best-performing model, selected based on the validation set, was then used as a standard model architecture for further training and evaluation on the final data set.

2.6. Bayesian Error Estimation Using Neural Networks

In addition to predicting stellar parameters, we train a secondary neural network to estimate the uncertainties of the Bayesian predictions. This network uses the same architecture as the primary network to ensure consistency and simplicity in design.

This secondary neural network takes the input photometric features (r, $u-g$, $g-r$, $r-i$, $i-z$) and photometric errors ($r_{\rm err}$, $u-g_{\rm err}$, $g-r_{\rm err}$, $r-i_{\rm err}$, $i-z_{\rm err}$) and is trained to predict the Bayesian uncertainties for each stellar parameter ($\sigma_{\rm B}$). These uncertainties represent the error in the Bayesian estimates for a given stellar parameter and are combined with the uncertainties from the primary network to give the total uncertainty of the predictions.

The loss function used for training this secondary network is a weighted sum of the MSE for each prediction of stellar parameter uncertainty, with the same weights as those used in the primary loss function. This allows the secondary network to focus on estimating the Bayesian errors for all parameters simultaneously, scaled by their respective uncertainties.

The need for this additional network arises because the primary neural network predicts uncertainties relative to the Bayesian estimates ($\sigma_{\rm NN}$), but does not account for the inherent uncertainty in the Bayesian estimates themselves. The total uncertainty is then calculated by combining the two error sources as follows:

$$\sigma_{\rm total} = \sqrt{\sigma_{\rm NN}^2 + \sigma_{\rm B}^2} \tag{3}$$

where $\sigma_{\rm NN}$ is the uncertainty in the neural network's prediction relative to the Bayesian estimate, and $\sigma_{\rm B}$ is the Bayesian error.

This formula applies under the assumption that the correlation between the uncertainties predicted by the neural network and those of the Bayesian model is negligible. To verify this, we calculated the correlation between the errors for each stellar parameter. The correlations are around 0.1 for M_r, around 0.05 for A_r, and around 0.2 for [Fe/H]. Since these values are close to zero, we can confidently use the expression in Equation (3) without including a correlation. This approach ensures that the final predictions account for both the uncertainty in the Bayesian model and the uncertainty in the neural network's predictions.

3. Results

The main argument for using neural networks against the Bayesian approach is that the NN algorithm is much faster without any significant compromise in accuracy. Therefore, the errors added by the NN approach should be much less than the uncertainties already introduced by the Bayesian algorithm. By performing the analysis on NN predictions, Bayesian predictions, and ground truth we can examine the NN performance and the loss in accuracy due to the speed-up.

This section presents the optimized size of training set (number of stars), the best-performing NN architecture after hyperparameter tuning, and a comparison between the neural network and Bayesian approaches for estimating stellar parameters (M_r, A_r, and [Fe/H]). We also include comparisons

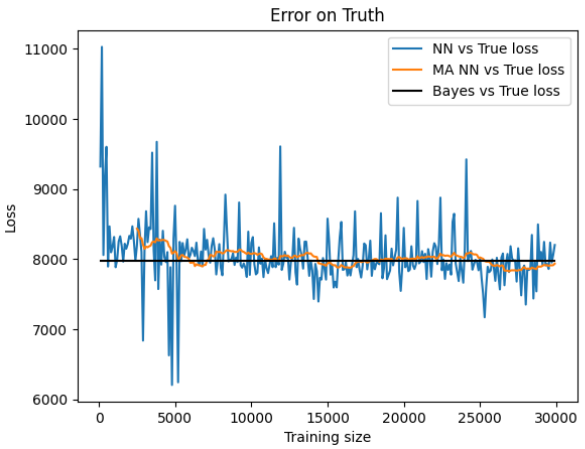

Figure 1. Search for the optimal size of the training set. The figure shows the magnitude and a moving average of the magnitude of a weighted loss function (Equation (2)) for the neural network predictions and true values for different sizes of training data sets. We found that the best training size is around 10,000 examples. Both curves display fluctuations due to stochastic training effects, but beyond 10,000 examples typical loss values plateau and show no significant improvement. The Bayesian loss remains flat because its predictions are fixed and independent of training size.

with ground truth from simulations and real-world data. The figures are divided into four categories: simulation results comparing Bayesian method versus ground truth, NN and Bayesian methods, NN versus ground truth results, and Bayesian versus NN results on real data.

3.1. Training Size

To determine the appropriate training size for our neural network, we tested various sample sizes. We found that 10,000 stars were the minimal training size in which the performance of the neural network was comparable to the ground truth in terms of both the loss and the MSE metrics. The loss function used here is the same as the one described in Section 2, which accounts for both the predicted values and their uncertainties. As shown in Figure 1, the loss values between the neural network and the true values stabilize at 10,000 examples. Although the curve exhibits noise due to the stochastic nature of training, the moving average stabilizes around 10,000 training examples. Beyond this point, the typical loss values plateau and show no significant improvement, suggesting that this is a reasonable balance point between performance and size of training set. Additionally, in both the neural network versus Bayesian estimates and the neural network versus true values, the error becomes negligible compared to the Bayesian method's own comparison with the true values.

All subsequent analyses in this work are performed on an independently selected test set and consistently demonstrate strong results. This reinforces the conclusion that using a training set of 10,000 stars is sufficient, and a further increase in training size does not justify the increase in the computational cost of generating training labels with the Bayesian algorithm in the full pipeline as described in Section 4.1.

3.2. Neural Network Architecture

The neural network architecture was chosen after performing an extensive hyperparameter tuning, and it was used in two separate neural networks: the primary network for predicting the stellar parameters and a secondary network for estimating uncertainties in the Bayesian photo-D algorithm. The details of the architecture are as follows.

1. *Input layer*. The input to the network consists of 10 neurons, with a linear activation function.
2. *First hidden layer*. The first hidden layer contains 45 neurons with the Gaussian error linear unit (GELU) activation function.
3. *Second hidden layer*. This layer consists of 45 neurons, using the Softsign activation function.
4. *Third hidden layer*. The third layer contains 42 neurons with the Softsign activation function.
5. *Fourth hidden layer*. The fourth layer consists of 33 neurons, using the Sigmoid activation function.
6. *Fifth hidden layer*. The fifth layer contains 36 neurons, using the GELU activation function.
7. *Output layer*. For the primary network the output layer contains six neurons while for the secondary it contains three. Both networks use a linear activation function.

Training details. The primary network was trained using the Adam optimizer (D. P. Kingma & J. Ba 2014), with a learning rate of 0.001. The training process was allowed to run for a maximum of 1024 epochs, with a batch size of 1024. If there was no improvement on the validation set for 100 epochs, the run was stopped early. The training included two complete runs, and the best-performing run was selected based on the performance of the validation set.

The secondary network used the same architecture as the primary network with one modification: the output layer consisted of only three neurons, corresponding to the uncertainties predicted for M_r, A_r, and [Fe/H]. This secondary network was trained for a maximum of 256 epochs, with only one complete training run, and the same early stopping criterion as the primary network was applied.

3.3. Comparison between Bayesian Estimates and Ground Truth (Simulation)

To establish the basis of comparison, we first evaluated the predictions of the Bayesian algorithm against the ground truth derived from simulations. This comparison highlights the inherent errors in the Bayesian estimates relative to the true values. While this analysis was done extensively by L. Palaversa et al. (2025), it is helpful to repeat it with the same data and in the same way as with NN prediction. These comparisons were performed on the test set of approximately 430,000 stars, which were not used during training as described in Section 2.1.

Figure 2 compares Bayesian estimates to actual parameter values, with a dashed red line indicating a perfect 1:1 correlation.

Residuals between Bayesian estimates and true values, as functions of $u - g$ color and its error, are shown in Figures 3, 4, and 5, mirroring the panels of L. Palaversa et al. (2025). No major bias is evident and results are as expected.

Figure 6 illustrates how Bayes-predicted residuals vary with differences in u magnitude, M_r, and [Fe/H]. The strong correlation between M_r and [Fe/H] residuals is due to the dependence of the M_r versus color relation on metallicity, which is essentially a shift in M_r as a function of [Fe/H] (see the bottom left panel of Figure 20 and Equation (A2) of Ž. Ivezić et al. 2008).

Residual histograms for M_r, A_r, and [Fe/H] (normalized by estimated standard deviations) are shown in Figure 7. A

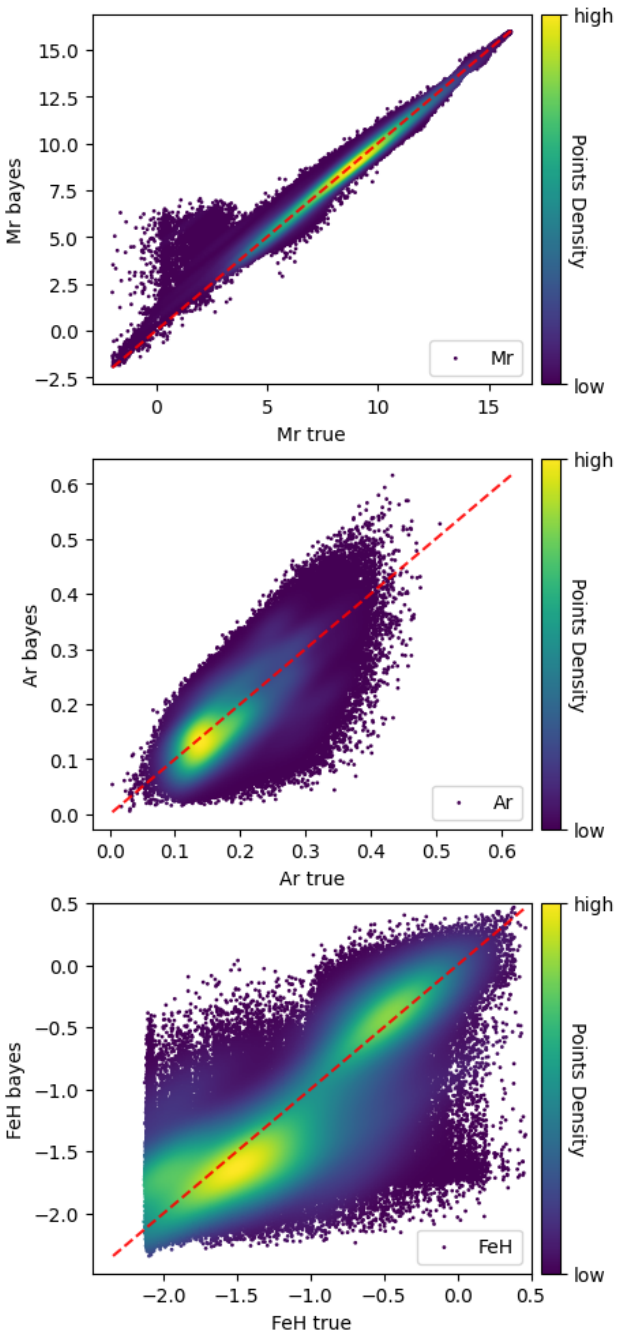

Figure 2. Comparison between Bayesian estimates and ground truth for M_r, A_r, and [Fe/H] using simulated data. The dashed red line indicates the 1:1 correlation. The data set includes 434,865 stars with a minimum magnitude SNR of 5 in all bands. The color scale represents the density of points.

Gaussian distribution reference ($N(0, 1)$) is marked by a red line to evaluate prediction uncertainties. All of these results are consistent with L. Palaversa et al. (2025).

3.4. Comparison between NN and Bayesian Estimates (Simulation)

In this subsection, we compare the performance of the neural network model to the Bayesian method using a simulated data set. This comparison was conducted on the same test set of around 430,000 stars described in Section 2.1.

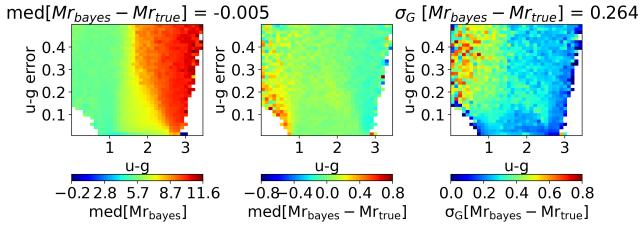

Figure 3. Performance of the Bayesian algorithm for M_r. The color code in the left panel shows the median of M_r in each pixel, that in the middle panel shows the median residuals ΔM_r (Bayesian – ground truth), and that in the right panel shows the standard deviation of the residuals.

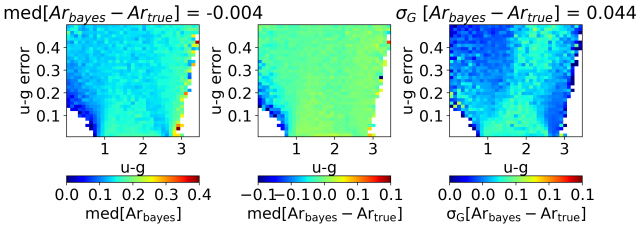

Figure 4. Performance of the Bayesian algorithm for A_r. The color code in the left panel shows the median of A_r in each pixel, that in the middle panel shows the median residuals ΔA_r (Bayesian – ground truth), and that in the right panel shows the standard deviation of the residuals.

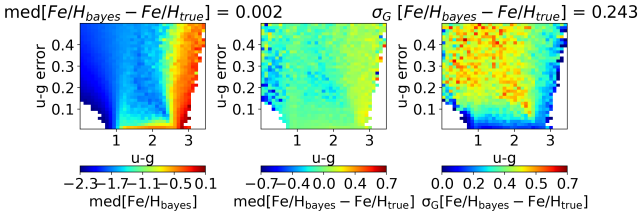

Figure 5. Performance of the Bayesian algorithm for [Fe/H]. The color code in the left panel shows the median of [Fe/H] in each pixel, that in the middle panel shows the median residuals Δ[Fe/H] (Bayesian – ground truth), and that in the right panel shows the standard deviation of the residuals.

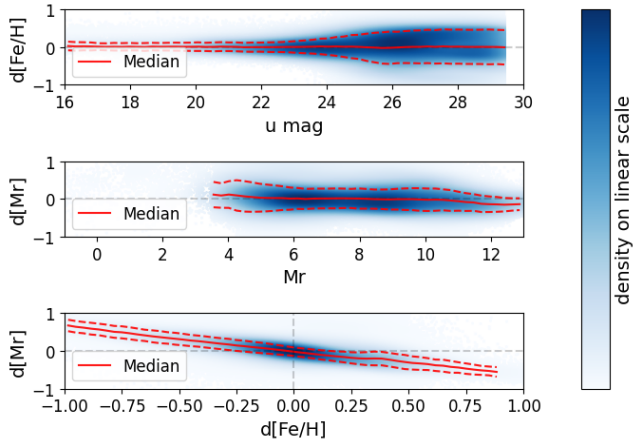

Figure 6. Residuals between Bayesian predictions and true values plotted against u magnitude (top), M_r (middle), and differences in [Fe/H] (bottom). The full red line represents the median residuals, the dashed red line represents the standard deviation of the residuals, and the density of residuals is shown on a color scale.

Figure 8 shows the comparisons between the NN and Bayesian estimations. The NN model demonstrates a strong agreement with the Bayesian approach across all three

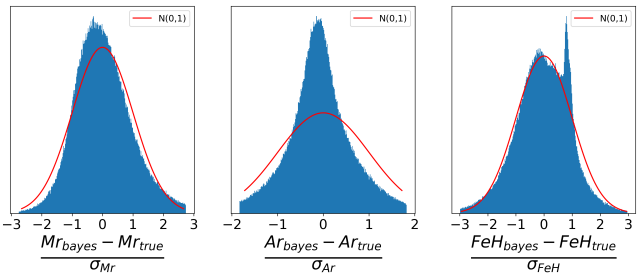

Figure 7. Histograms of residuals between Bayesian estimates and ground truth for M_r (left), A_r (middle), and [Fe/H] (right). The red curves show a normal distribution ($N(0, 1)$) for comparison.

parameters. Although some scatter is observed, particularly for A_r, the overall trends follow the expected behavior. This scatter is much less for M_r and [Fe/H] than in Figure 2, showing that the dominant source of error is the Bayesian algorithm. A_r scatter, while higher, is still comparable to the Bayesian algorithm as seen in Table 1.

Residuals between the NN and Bayesian estimates, as functions of $u - g$ color and its error, are presented in Figures 9, 10, and 11. Figures show the bias dependence of the neurons associated with the $u - g$ color. Results show consistent residuals across most regions, with slightly higher residuals of [Fe/H] in some regions, likely due to its sensitivity to photometric errors. Compared to Figures 3, 4, and 5, biases are less pronounced, which is also seen in Table 1.

Figure 12 shows that the residuals between the NN-predicted values and the Bayes estimates are smaller in areas where the data set has the most examples. The median [Fe/H] residual is stable across the range of u magnitude. M_r residuals are tightly clustered around zero with small deviations for brighter stars where the representation of the stars is scarce. The metallicity residuals do slightly affect the M_r estimates, which is to be expected from the dependence on metallicity of the absolute magnitude versus color relation (for details see L. Palaversa et al. 2025).

Figure 13 shows that M_r, A_r, and [Fe/H] residuals follow a Gaussian distribution, with only minor deviations in the tails. M_r and [Fe/H] show a slight overestimation of the standard deviation, while A_r shows a slight underestimation. Despite this, the overall distribution remains centered around zero, suggesting that overall the NN performs without any significant bias.

In summary, the results demonstrate that the neural network model provides stellar parameter estimates that are highly consistent with those from the Bayesian method. Both the residuals and error distributions show that the NN maintains a high level of accuracy across all parameters. This is also supported by the residual histograms shown in Figure 13, as well as the summary statistics presented in Table 1, which quantify both the bias and scatter for each parameter across the methods. While there are slight deviations from the Bayes predictions, those are all negligible compared to the Bayes uncertainties.

3.5. Comparison between NN and Ground Truth (Simulation)

To fully measure the performance of the NN algorithm, we also compare the performance of the neural network model against the ground truth using simulated data. This evaluation

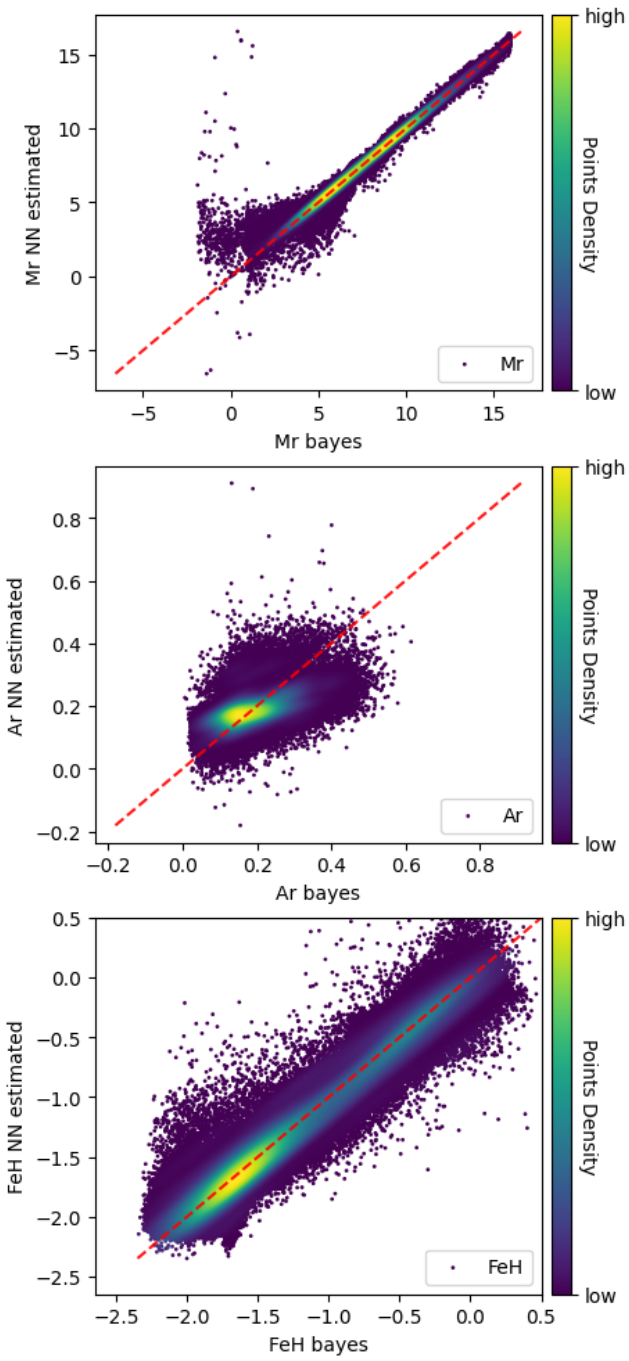

Figure 8. Comparison between NN and Bayesian estimates for M_r, A_r, and [Fe/H] using simulated data. The dashed red line indicates the 1:1 correlation. The data set includes 434,865 stars with a minimum magnitude SNR of 5 in all bands. The color scale represents the density of points.

was performed on the test set of approximately 430,000 simulated stars, which were not seen during training.

Figure 14 compares the NN estimates and the ground truth. The NN shows a close alignment with the ground truth, especially for M_r. Most of the features in this figure are also present in Figure 2. A larger amount of scatter is visible than in Figure 8, due to challenges in accurately estimating those parameters with the Bayesian approach.

Figures 15, 16, and 17 show the predicted values of the stellar parameters as well as residuals in the $u - g$ and $u - g$

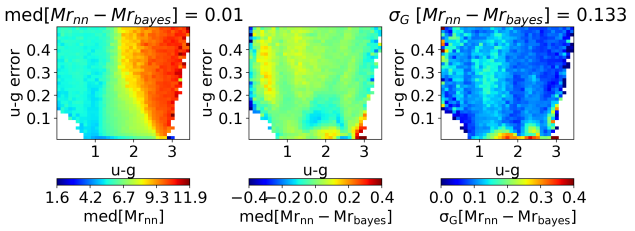

Figure 9. Performance of the NN algorithm for M_r. Analogous to Figure 3 except the residuals are defined as (NN – Bayesian).

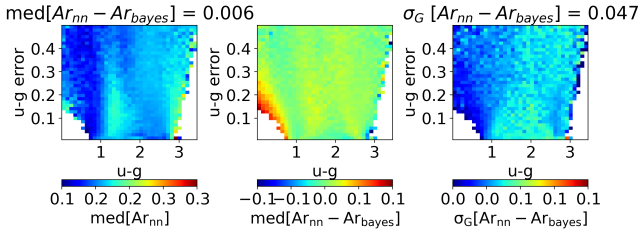

Figure 10. Performance of the NN algorithm for A_r. Analogous to Figure 4 except the residuals are defined as (NN – Bayesian).

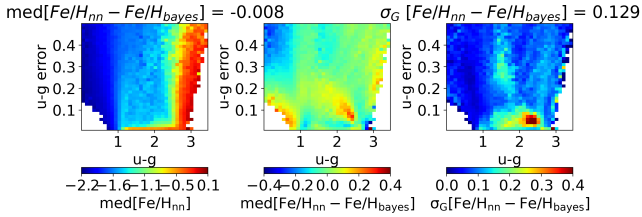

Figure 11. Performance of the NN algorithm for [Fe/H]. Analogous to Figure 5 except the residuals are defined as (NN – Bayesian).

Table 1
Median and Robust Standard Deviation of Residuals for Each Stellar Parameter

Comp.	Parameter	Median	Robust Std. Dev.
Bayes versus truth	M_r	−0.010	0.282
	A_r	−0.009	0.041
	[Fe/H]	−0.006	0.315
NN versus Bayes	M_r	0.005	0.123
	A_r	0.007	0.048
	[Fe/H]	−0.003	0.110
NN versus truth	M_r	0.005	0.324
	A_r	−0.004	0.055
	[Fe/H]	−0.014	0.352

Note. The data set consists of 434,865 simulated stars with SNR > 5 in all bands.

error space. All the regions are very similar to the performance of the Bayes algorithm (Figures 3, 4, and 5).

[Fe/H] and M_r residuals on Figure 18 remain close to zero across most of the magnitude range, while the scatter increases for the faintest stars, where photometric errors are larger. Residuals for M_r are centered around zero for the whole M_r range, and we can see a clear correlation between residuals in M_r and [Fe/H], which is expected. The main scatter comes from Bayes estimation, which is evident when comparing Figures 12 and 6 with Figure 18.

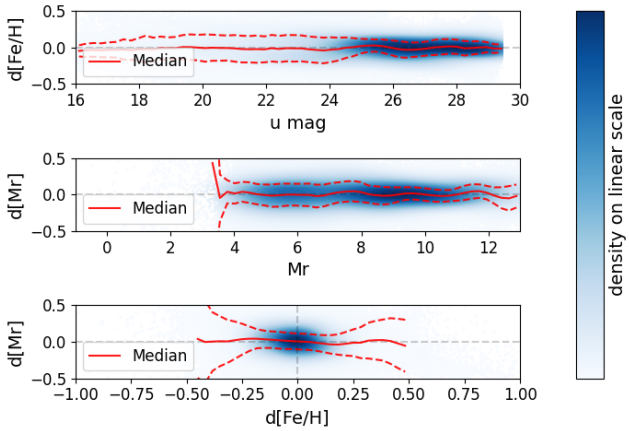

Figure 12. Residuals between NN predictions and Bayes estimates plotted against u magnitude (top), M_r (middle), and differences in [Fe/H] (bottom). The full red line represents the median residuals, the dashed red line represents the standard deviation of the residuals, and the density of residuals is shown on a color scale. Limits of the y-axis are halved when compared to Figure 6.

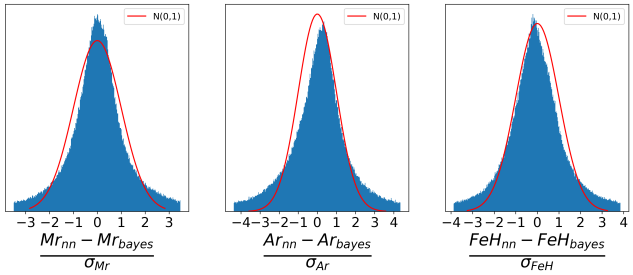

Figure 13. Histograms of residuals between NN and Bayesian estimates for M_r (left), A_r (middle), and [Fe/H] (right). The red curves show a normal distribution ($N(0, 1)$) for comparison.

Finally, Figure 19 shows that the normalized residuals for M_r align closely with a normal distribution, indicating that the NN provides a reliable prediction for absolute magnitude. A_r shows greater deviation, suggesting that the errors are slightly overestimated. Still, this overestimation is overall smaller than with the Bayes algorithm. [Fe/H] shows a very slight bias, which is also present in Bayes estimations.

Table 1 summarizes the residual statistics for each of the three stellar parameters. For each comparison, we report the median and robust standard deviation of the residuals.

These values quantitatively support our claim that the dominant source of error arises from the Bayesian algorithm. The residuals between the NN and Bayesian predictions are significantly smaller in both bias and scatter than the residuals between Bayesian estimates and the true values. Moreover, the NN residuals relative to the ground truth remain comparable in scale to the Bayesian residuals, confirming that the speed-up introduced by the NN does not come at a significant cost in accuracy.

Overall, the main source of errors is the Bayes estimation of truth, while the NN estimation of the Bayes contributes only a small amount of the total error.

3.6. Comparison with Observations

In addition to simulated data, we applied the NN model to observational data and compared the estimates with those from the Bayesian method. Figure 20 displays the comparison of the

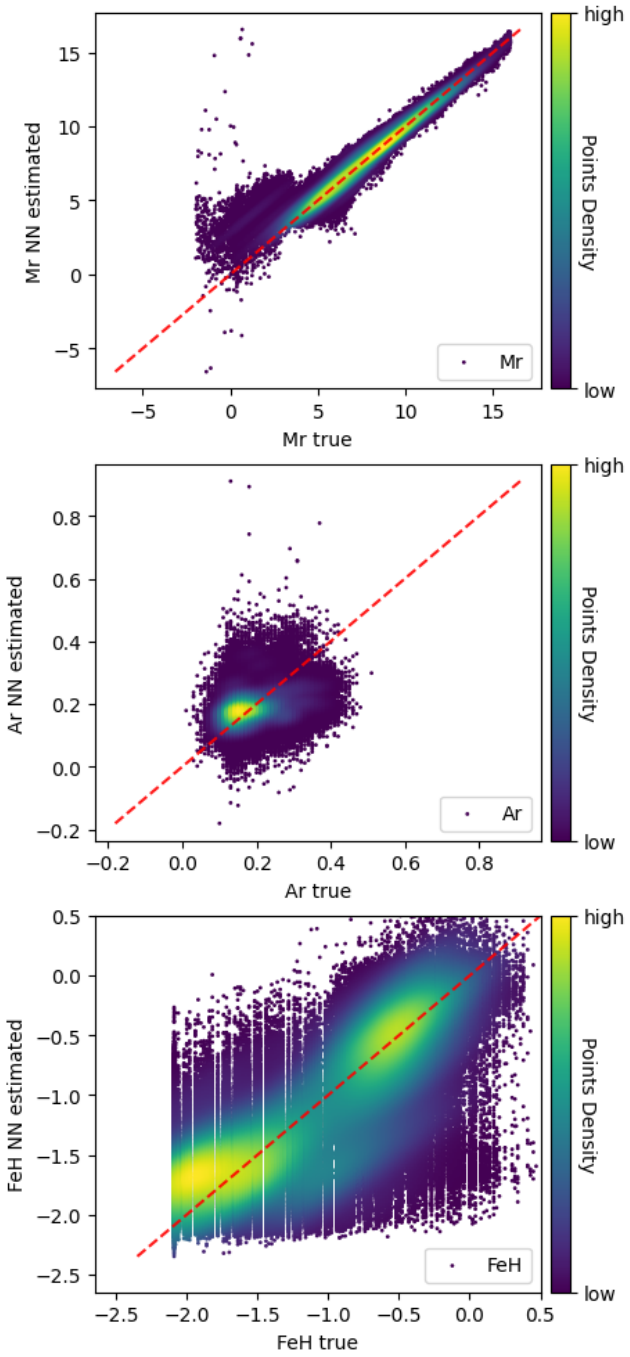

Figure 14. Comparison between NN and ground truth estimates for M_r, A_r, and [Fe/H] using simulated data. The dashed red line indicates the ideal 1:1 correlation. The data set includes 434,865 stars with a minimum magnitude's SNR of 5 in all bands.

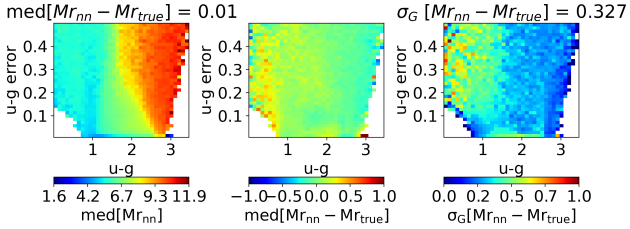

Figure 15. Performance of the NN algorithm for M_r. Analogous to Figure 3 except the residuals are defined as (NN – ground truth).

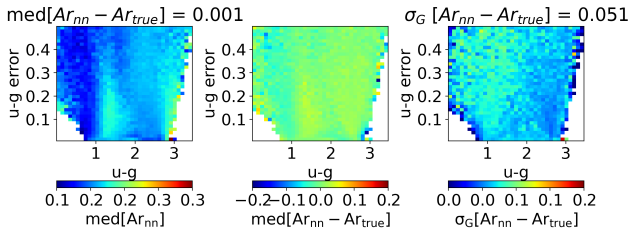

Figure 16. Performance of the NN algorithm for A_r. Analogous to Figure 4 except the residuals are defined as (NN – ground truth).

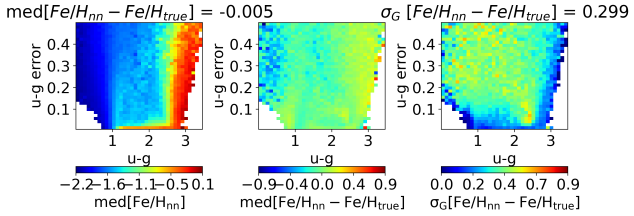

Figure 17. Performance of the NN algorithm for [Fe/H]. Analogous to Figure 5 except the residuals are defined as (NN – ground truth).

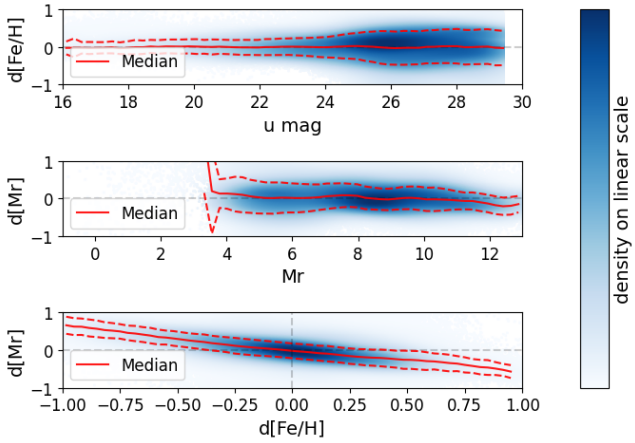

Figure 18. Residuals between NN predictions and true values plotted against u magnitude (top), M_r (middle), and differences in [Fe/H] (bottom). The full red line represents the median residuals, the dashed red line represents the standard deviation of the residuals, and the density of residuals is shown on a color scale.

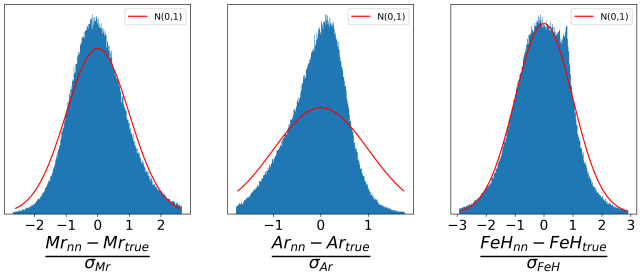

Figure 19. Histograms of residuals between NN and ground truth estimates for M_r (left), A_r (middle), and [Fe/H] (right), normalized by predicted uncertainties. The red curves show a normal distribution ($N(0, 1)$) for comparison.

NN predictions and Bayesian estimates for M_r, A_r, and [Fe/H] on real data.

As seen in the simulations, the NN model's predictions closely follow the Bayesian estimates across all three parameters, with a

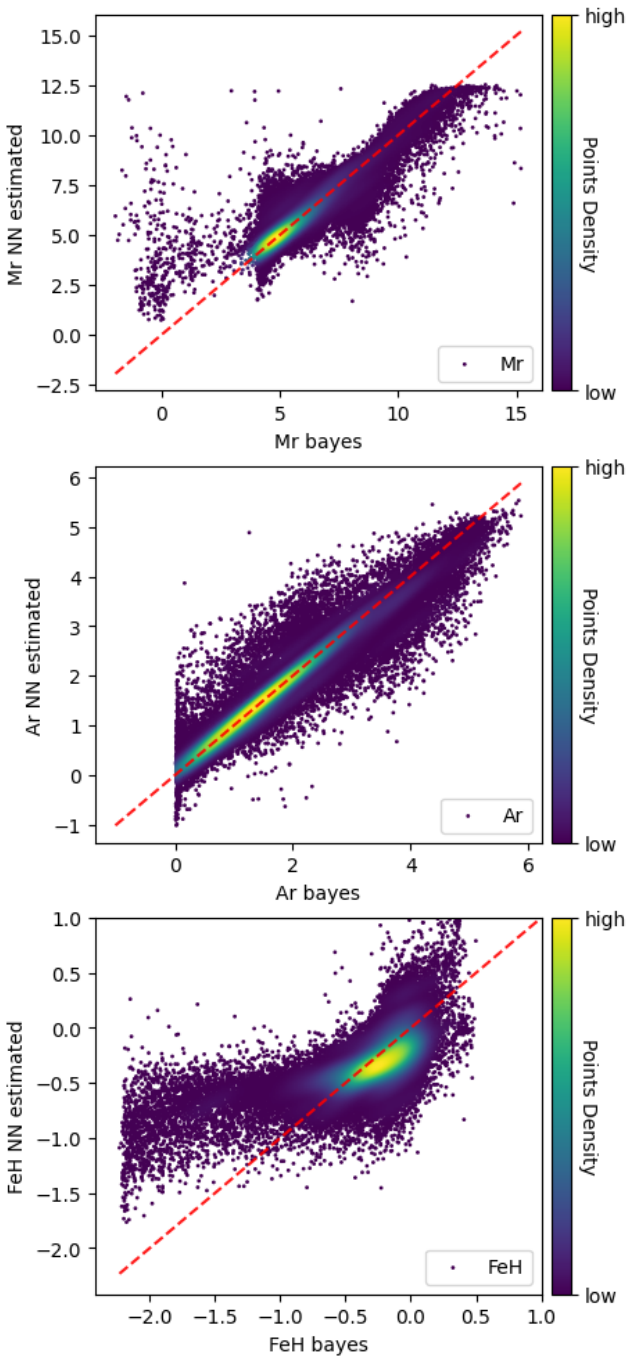

Figure 20. Comparison between NN and Bayesian estimates for M_r, A_r, and [Fe/H] using real observational data. The dashed red line indicates the ideal 1:1 correlation. The data set includes 147,064 stars with a minimum magnitude's SNR of 5 in all bands.

near 1:1 correlation line, as shown by the dashed red lines in Figure 20. However, some scatter is observed, particularly in the metallicity ([Fe/H]) estimates. We observe that the neural network estimates of [Fe/H] tend to saturate around −0.5 for a subset of stars where the Bayesian method predicts significantly lower metallicities (below −0.7). We find that approximately 5% of the data set falls into this regime. The likely explanation is that low-metallicity stars are underrepresented in the training set, which limits the neural network's ability to accurately learn the mapping in this region. Additionally, the abrupt drop in the

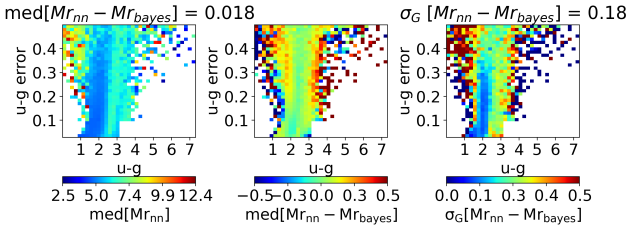

Figure 21. Performance of the NN algorithm for M_r using real observational data. Analogous to Figure 3 except the residuals are defined as (NN – Bayesian).

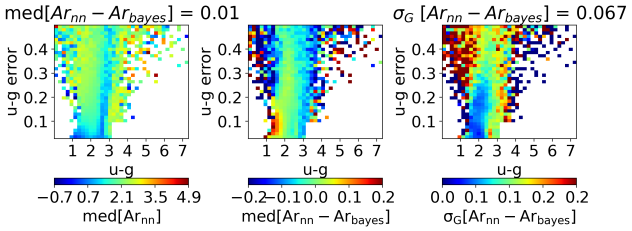

Figure 22. Performance of the NN algorithm for A_r using real observational data. Analogous to Figure 4 except the residuals are defined as (NN – Bayesian).

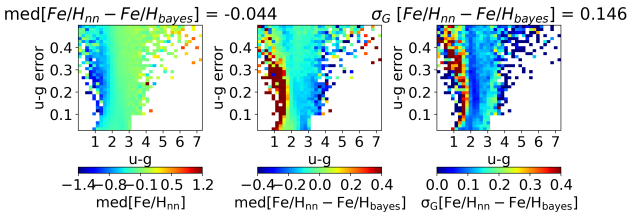

Figure 23. Performance of the NN algorithm for [Fe/H] using real observational data. Analogous to Figure 5 except the residuals are defined as (NN – Bayesian).

density of stars with Bayes $M_r < 4$ is due to the main-sequence turn-off, which reflects the age of the stellar population.

As seen in Figure 21, the M_r estimates in real data show good agreement between NN and Bayesian methods. The scatter for M_r remains minimal, increasing only in the regions of low SNR. Next, in Figure 22, we show the comparison of extinction (A_r) between the NN and Bayesian predictions. The two methods align well, although some scatter is noticeable for stars with larger color errors. This is consistent with the behavior observed in simulations, where extinction estimates tend to be more challenging due to photometric uncertainties. Lastly, the comparison for metallicity ([Fe/H]) is presented in Figure 23. Here, we observe a more pronounced bias than for M_r and A_r, likely due to the complexity of accurately estimating metallicity with low-SNR photometric colors. The scatter increases for stars with lower SNRs, where uncertainties in the $u - g$ color become more significant.

We further analyze how the residuals behave across different parameter ranges by plotting them as a function of u magnitude, M_r, and Δ[Fe/H], as shown in Figure 24. The [Fe/H] residuals remain relatively small across the magnitude range, with slight deviations for fainter stars. Similarly, the M_r residuals are tightly clustered around zero and a slight correlation between [Fe/H] and M_r residuals is noticed, as observed in the simulations.

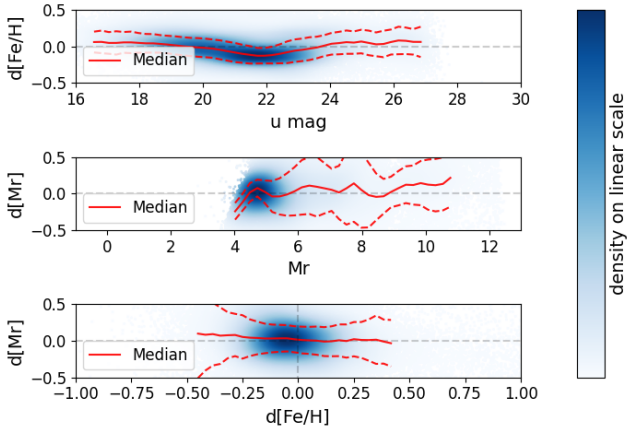

Figure 24. Residuals between NN predictions and true values plotted against u magnitude (top), M_r (middle), and differences in [Fe/H] (bottom) for real observational data. The full red line represents the median residuals, the dashed red line represents the standard deviation of the residuals, and the density of residuals is shown on a color scale.

Finally, in Figure 25 we present the histograms of the residuals between the NN and Bayesian estimates normalized by uncertainties predicted by the primary network for M_r, A_r, and [Fe/H], compared to a standard Gaussian. The distributions for M_r and A_r align closely with the Gaussian, but there is a slight deviation for [Fe/H] and a slight bias, indicating that metallicity estimates may be more affected by photometric errors.

In summary, the NN model performs well on observational data, closely matching the Bayesian estimates for M_r and A_r. However, the [Fe/H] predictions show more scatter, particularly for stars with higher photometric errors. This analysis confirms the robustness of the NN model for large-scale surveys like LSST, while also highlighting the challenges in estimating metallicity from photometry alone.

4. Discussion and Conclusion

4.1. Integration Plan for a Full-sky Data Set

To efficiently handle the massive amount of data expected from the Vera C. Rubin Observatory's Legacy Survey of Space and Time, we have developed a pipeline that balances the strengths of Bayesian methods and neural networks, optimizing both speed and accuracy. The pipeline significantly accelerates the estimation of stellar parameters while maintaining Bayesian-level precision.

The method we employ starts by applying the Bayesian approach to approximately 5%–10% of the full data set (around 10,000 examples), carefully selected to represent the diverse stellar populations across the sky. The stellar parameters (M_r, A_r, [Fe/H]) estimated from this sample are then used to train both the primary neural network and the secondary network. Once trained, the neural networks processes the remaining 90%–95% of the data, significantly speeding up the overall computation.

To account for the regional variation in stellar populations and dust extinction across the sky, we divide the sky into patches of approximately 10 deg^2. The final results from all patches are aggregated to produce a full-sky map of stellar parameters.

Overall, this pipeline is capable of processing large data sets in a fraction of the time required for a fully Bayesian approach, without compromising accuracy. The results from this pipeline will be made publicly available to benefit the wider astronomical community.

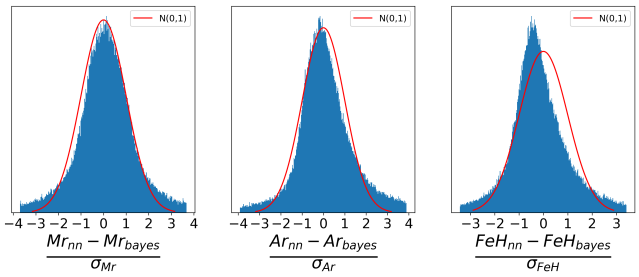

Figure 25. Histograms of residuals between NN and Bayesian estimates for M_r (left), A_r (middle), and [Fe/H] (right) using real observational data, normalized by predicted uncertainties. The red curves represent a normal distribution ($N(0, 1)$) for comparison.

4.2. Computational Efficiency

One of the key advantages of the hybrid Bayesian–neural network pipeline is its significant reduction in computation time. The Bayesian algorithm, when applied to the full data set of 440,000 stars, is expected to take approximately 4400 s. However, by applying the Bayesian method to only a small fraction of the data, we significantly reduce the computation time required for this step.

We measured the time required to process the same data set using the NN and found the average time to be 65 s with a standard deviation of approximately 20 s. Given these results, the hybrid approach would take approximately 165 s to evaluate the full data set of 440,000 stars. This represents a drastic reduction in computational time compared to the full Bayesian approach, making the pipeline feasible for processing the vast amounts of data expected from future LSST releases.

All computations were performed on CPUs. During the development and testing phases, we observed that training with GPUs led to a linear slowdown proportional to the number of GPUs used. This is likely due to the simplicity of the model, where the overheads associated with the use of GPUs outweigh the benefits. Given that the model is lightweight, the CPU-based approach provides low-cost alternatives without requiring expensive hardware and could be effectively run on most consumer-grade personal computers.

From our analysis, we observed that the uncertainties predicted by the Bayesian method are, on average, much higher than those predicted by the NN. This is consistent across most stars in the data set as seen in Figures 26 and 27. The lower uncertainties from the NN suggest that the model has learned the inherent structure in the data efficiently and without introducing significant additional uncertainty into the predictions.

This computational efficiency is a major benefit of our method, allowing for rapid processing of large data sets without sacrificing accuracy. By applying the hybrid pipeline, we can ensure the feasibility of processing billions of stars, which are expected by the upcoming sky surveys.

4.3. Conclusion

Our results demonstrate that a neural network, once trained, can provide predictions that are comparable in accuracy to those from Bayesian methods while being over an order of

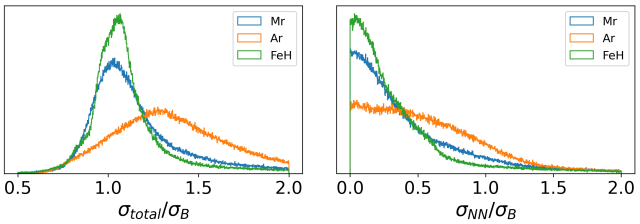

Figure 26. Comparison of uncertainties using simulated data. Left: histograms of the ratio $\sigma_{\rm total}/\sigma_{\rm B}$ for all three stellar parameters show that the total error is nearly equal to the Bayesian error. Right: histograms of the ratio $\sigma_{\rm NN}/\sigma_{\rm B}$ show that neural network-induced error is typically much smaller than the Bayesian error. This confirms that the dominant uncertainty originates from the Bayesian estimates.

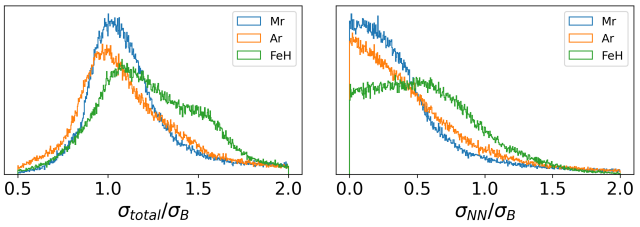

Figure 27. Same as Figure 26 but for real observational data. The distributions are broader, but the same trend holds: the total uncertainty is dominated by the Bayesian method, while the neural network adds only a small contribution.

magnitude faster. The neural network consistently produced uncertainties that were significantly lower than those predicted by the Bayesian approach, indicating that the model has successfully learned the structure of the data without introducing additional errors.

Looking forward, the application of this pipeline to real LSST data will allow for the efficient analysis of stellar populations across the entire sky, enabling new discoveries in Galactic structure, stellar evolution, and the interstellar medium. The flexibility of the pipeline allows it to be adapted for other large-scale sky surveys, making it a versatile tool for the broader astronomical community.

Overall, the hybrid Bayesian–neural network pipeline represents a significant step forward in the ability to process vast astronomical data sets efficiently, and it will play a key role in the analysis of the unprecedented data volume expected from future sky surveys.

Acknowledgments

Ž.I. acknowledges funding by the Fulbright Foundation and thanks the Ruđer Bošković Institute for hospitality. K.M. and Ž.I. acknowledge support from the DiRAC Institute in the Department of Astronomy at the University of Washington. K.M. and Ž.I. acknowledge support from the Center for Advanced Computing and Modeling, University of Rijeka, and the HPC Bura. This work is financed within the Tenure Track Pilot Programme of the Croatian Science Foundation and the Ecole Polytechnique Fédérale de Lausanne and the Project TTP-2018-07-1171 "Mining the Variable Sky," with the funds of the Croatian-Swiss Research Programme.

Facilities: Rubin:Simonyi, Gaia, Sloan.

Software: Astropy (Astropy Collaboration et al. 2013, 2018, 2022), TensorFlow (M. Abadi et al. 2015), Keras (F. Chollet et al. 2015), Matplotlib (J. D. Hunter 2007), SciPy (P. Virtanen et al. 2020).

Appendix

To demonstrate how photometric errors can affect metallicity estimates, a simplified model was employed inspired by actual observations. The model values are obtained by randomly sampling [Fe/H] from both a flat distribution spanning the range from −3 to 0 and a distribution that is made of two equal-size concatenated Gaussian distributions centered on [Fe/H] = −1.5 and [Fe/H] = −0.5, and with σ equal to 0.3 dex and 0.2 dex, respectively. The colors are calculated using the equation

$$y = 0.84 + 0.34 \times 10^{0.45x} \quad \text{(A1)}$$

where x represents [Fe/H] and y corresponds to the $u - g$ color. This particular functional form (A1) can be explained by the absorption of flux in the ultraviolet region by metallic lines according to Beer's law. The coefficients in the equation are determined by fitting SDSS data from stars with $g - r \sim 0.3$ (Ž. Ivezić et al. 2008). To simulate the scatter caused by measurement errors, random variations were introduced to both [Fe/H] and $u - g$ using Gaussian distributions with a standard deviation of $\sigma = 0.1$.

The resulting simulated sample is presented in Figures 28 and 29. This visualization demonstrates that using the median value (or any other average statistic) of spectroscopic [Fe/H] within bins of $u - g$ color as a photometric metallicity estimator fails when $u - g$ errors are not negligible, particularly in comparison to the relevant dynamic range. Specifically, the median [Fe/H] values are biased toward higher values for regions of low [Fe/H] (blue $u - g$) and biased toward lower values for regions of high [Fe/H] (red $u - g$).

The magnitude of this bias depends on the errors in $u - g$ color. In Figure 30, the bias is depicted as a function of assumed $u - g$ errors for three different true $u - g$ values. Note that the metallicity bias can be positive or negative. As

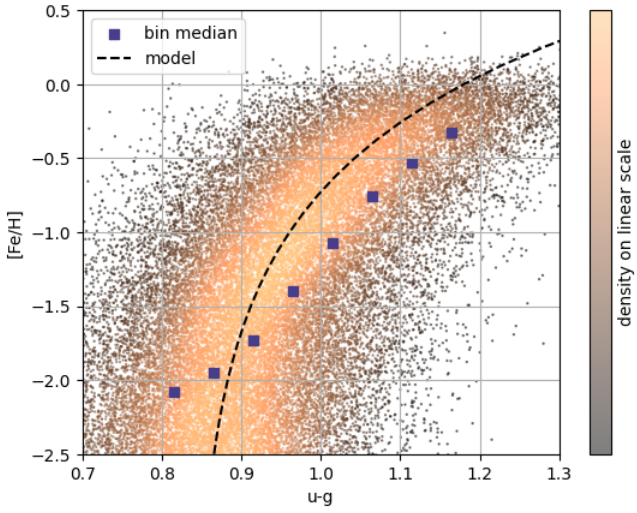

Figure 28. Uniform toy model illustrating the bias in photometric metallicity estimation caused by significant errors in $u - g$ color. The panel shows approximately 40,000 small dots, generated according to the process outlined in the text. The dashed line represents Equation (A1). The large symbols represent the median [Fe/H] values within each bin of $u - g$ color. The figure is generated by sampling [Fe/H] from a uniform distribution before calculating Equation (A1).

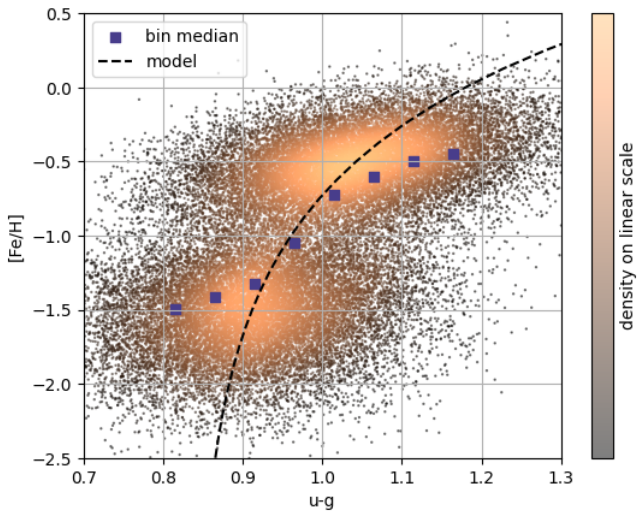

Figure 29. Binormal toy model illustrating the bias in photometric metallicity estimation caused by significant errors in $u - g$ color. Analogous to Figure 28 except it uses a distribution made of two equal-size concatenated samples from the Gaussian distributions centered on [Fe/H] $= -1.5$ and [Fe/H] $= -0.5$, and with σ equal to 0.3 dex and 0.2 dex, respectively.

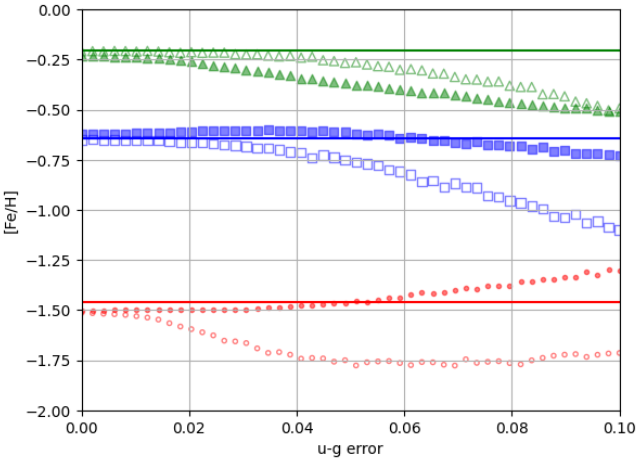

Figure 30. Demonstration of the median [Fe/H] as a function of assumed $u - g$ errors, using Gaussian distributions of $u - g$, for three different $u - g$ colors: 0.89 (circles), 0.99 (squares), and 1.09 (triangles). Lines represent the true values of [Fe/H], open symbols correspond to medians of a uniform [Fe/H] distribution (as shown in Figure 28, while closed symbols represent the case where [Fe/H] is composed of two concatenated Gaussian distributions of equal size, centered at [Fe/H] $= -1.5$ and [Fe/H] $= -0.5$, with standard deviations of 0.3 dex and 0.2 dex, respectively, as shown in Figure 29.

expected, no bias is observed when errors are negligible. The bias is also influenced by the underlying true [Fe/H] distribution; as seen in Figure 30, the difference in bias between the uniform underlying true [Fe/H] distribution and a binormal distribution is significant. Additionally, the bias is particularly pronounced when the "measured" $u - g$ values are bluer than the bluest possible $u - g$ color in the absence of noise ($u - g < 0.84$, as stated in Equation (A1)).

ORCID iDs

Karlo Mrakovčić https://orcid.org/0009-0009-8154-3827
Željko Ivezić https://orcid.org/0000-0001-5250-2633
Lovro Palaversa https://orcid.org/0000-0003-3710-0331

References

Abadi, M., Agarwal, A., Barham, P., et al. 2015, Tensorflow, Zenodo, doi:10.5281/zenodo.4724125
Astropy Collaboration, Price-Whelan, A. M., Lim, P. L., et al. 2022, ApJ, 935, 167
Astropy Collaboration, Price-Whelan, A. M., Sipőcz, B. M., et al. 2018, AJ, 156, 123
Astropy Collaboration, Robitaille, T. P., Tollerud, E. J., et al. 2013, A&A, 558, A33
Bailer-Jones, C. A. L., Rybizki, J., Fouesneau, M., Demleitner, M., & Andrae, R. 2021, AJ, 161, 147
Berry, M., Ivezić, Ž., Sesar, B., et al. 2012, ApJ, 757, 166
Bond, N. A., Ivezić, Ž., Sesar, B., et al. 2010, ApJ, 716, 1
Chollet, F., et al. 2015, Keras, https://Keras.io
Clevert, D.-A., Unterthiner, T., & Hochreiter, S. 2015, arXiv:1511.07289
Dal Tio, P., Pastorelli, G., Mazzi, A., et al. 2022, ApJS, 262, 22
Dugas, C., Bengio, Y., Bélisle, F., Nadeau, C., & Garcia, R. 2000, in Advances in Neural Information Processing Systems, ed. T. Leen, T. Dietterich, & V. Tresp, 13 (Cambridge, MA: MIT Press) https://proceedings.neurips.cc/paper_files/paper/2000/file/44968aece94f667e4095002d140b5896-Paper.pdf
Hahnloser, R. H. R., Sarpeshkar, R., Mahowald, M. A., Douglas, R. J., & Seung, H. S. 2000, Natur, 405, 947
Hendrycks, D., & Gimpel, K. 2016, arXiv:1606.08415
Hunter, J. D. 2007, CSE, 9, 90
Ivezić, Ž., Kahn, S. M., Tyson, J. A., et al. 2019, ApJ, 873, 111
Ivezić, Ž., Sesar, B., Jurić, M., et al. 2008, ApJ, 684, 287
Jeffrey, N., & Wandelt, B. D. 2020, arXiv:2011.05991
Jurić, M., Ivezić, Ž., Brooks, A., et al. 2008, ApJ, 673, 864
Kingma, D. P., & Ba, J. 2014, arXiv:1412.6980
Klambauer, G., Unterthiner, T., Mayr, A., & Hochreiter, S. 2017, in Advances in Neural Information Processing Systems 30 (NIPS 2017), ed. Isabelle Guyon, Ulrike von Luxburg, Bengio, Wallach, Fergus, Vishwanathan, & Garnett (Red Hook, NY: Curran Associates, Inc.)
Li, L., Jamieson, K., DeSalvo, G., Rostamizadeh, A., & Talwalkar, A. 2018, JMLR, 18, 1
Loebman, S. R., Ivezić, Ž., Quinn, T. R., et al. 2012, ApJL, 758, L23
Loebman, S. R., Ivezić, Ž., Quinn, T. R., et al. 2014, ApJ, 794, 151
Macêdo, D., Zanchettin, C., Oliveira, A. L., & Ludermir, T. 2019, ESWA, 124, 271
Nair, V., & Hinton, G. E. 2010 in Proc. 27th Int. Conf. on Int. Conf. on Machine Learning, ICML'10 (Madison, WI: Omnipress), 807
Palaversa, L., Ivezic, Z., Caplar, N., et al. 2025, AJ, 169, 119
Ramachandran, P., Zoph, B., & Le, Q. V. 2017, arXiv:1710.05941
Virtanen, P., Gommers, R., Oliphant, T. E., et al. 2020, NatMe, 17, 261
Wilson, D., & Martinez, T. 2001, in IJCNN'01. International Joint Conference on Neural Networks. Proceedings (Cat. No.01CH37222) (New York: IEEE), 115