






# Machine Learning the Sixth Dimension: Stellar Radial Velocities from 5D Phase-space Correlations

Adriana Dropulic<sup>1</sup> , Bryan Ostdiek<sup>2,3</sup> , Laura J. Chang<sup>1</sup> , Hongwan Liu<sup>1,4</sup> , Timothy Cohen<sup>5</sup> , and Mariangela Lisanti<sup>1</sup> 

<sup>1</sup>Department of Physics, Princeton University, Princeton, NJ 08544, USA; [dropulic@princeton.edu](mailto:dropulic@princeton.edu)

<sup>2</sup>Department of Physics, Harvard University, Cambridge, MA 02138, USA

<sup>3</sup>The NSF AI Institute for Artificial Intelligence and Fundamental Interactions, USA

<sup>4</sup>Center for Cosmology & Particle Physics, Department of Physics, New York University, New York, NY 10003, USA

<sup>5</sup>Institute for Fundamental Science, Department of Physics, University of Oregon, Eugene, OR 97403, USA

Received 2021 April 16; revised 2021 June 3; accepted 2021 June 9; published 2021 July 1

## Abstract

The Gaia satellite will observe the positions and velocities of over a billion Milky Way stars. In the early data releases, the majority of observed stars do not have complete 6D phase-space information. In this Letter, we demonstrate the ability to infer the missing line-of-sight velocities until more spectroscopic observations become available. We utilize a novel neural network architecture that, after being trained on a subset of data with complete phase-space information, takes in a star's 5D astrometry (angular coordinates, proper motions, and parallax) and outputs a predicted line-of-sight velocity with an associated uncertainty. Working with a mock Gaia catalog, we show that the network can successfully recover the distributions and correlations of each velocity component for stars that fall within  $\sim 5$  kpc of the Sun. We also demonstrate that the network can accurately reconstruct the velocity distribution of a kinematic substructure in the stellar halo that is spatially uniform, even when it comprises a small fraction of the total star count.

*Unified Astronomy Thesaurus concepts:* [Radial velocity \(1332\)](#); [Neural networks \(1933\)](#); [Dark matter \(353\)](#); [Milky Way dynamics \(1051\)](#)

## 1. Introduction

Gaia has ushered in a new age in astrometry, with the goal of providing precise positions and velocities for an unprecedented number of stars in the Milky Way (Gaia Collaboration et al. 2016, 2018). This complete phase-space information will revolutionize our understanding of both disk and halo dynamics. In the current data release (EDR3), most Gaia stars only have 5D astrometry available (two angular coordinates, two proper motions, and parallax); less than 1% of the stars have a measured line-of-sight velocity (Gaia Collaboration et al. 2021). We demonstrate how to use regressive neural networks to successfully predict stellar line-of-sight velocities with an associated uncertainty from 5D astrometry. This approach increases the scientific output of early Gaia data releases until more spectroscopic data is available.

The science applications that benefit from having line-of-sight velocities are vast—see Wilkinson et al. (2005) for a review—and include measuring the Milky Way potential, obtaining the local dark matter density, distinguishing the thin and thick disk, and mapping substructure in the stellar disk. One particular case where having full stellar phase-space information is highly beneficial is the identification of stellar remnants of disrupted satellite galaxies in the Milky Way (Johnston et al. 1996; Johnston 1998; Bullock & Johnston 2005). Such mergers are a natural consequence of hierarchical structure formation (White & Rees 1978), and in addition to depositing stars that form the stellar halo, also leave behind dark matter substructures.

The neural network approach proposed in this Letter has the potential to dramatically and immediately increase the subset of Gaia data that can be used toward these applications. We present the results of training, validating, and testing the network using a simulated Gaia mock catalog that models both the smooth stellar halo and disk, as well as kinematic

substructure in the halo. The network is trained on a subset of stars with full 6D phase-space information using a simple network loss function similar to the method described in Nix & Weigend (1994). While 5D astrometry alone is insufficient to provide a reliable prediction of the line-of-sight velocity for every individual star, incorporating the learned uncertainty allows us to obtain accurate *distributions* of the line-of-sight velocity for the full population of stars, as well as the correlations between different velocity components in the Galactocentric frame. Additionally, the confidence of the network prediction of the line-of-sight velocity of each star can be inferred from the learned uncertainty.

The machine-learning approach introduced in this Letter is intended to work in tandem with spectroscopic surveys, such as Apache Point Observatory Galactic Evolution Experiment (APOGEE), the Gaia-ESO-Survey, Galactic Archaeology with HERMES (GALAH), Large Sky Area Multi-Object Fibre Spectroscopic Telescope (LAMOST), Radial Velocity Experiment (RAVE), and Sloan Extension for Galactic Understanding and Exploration (SEGUE; Yanny et al. 2009; Cui et al. 2012; Gilmore et al. 2012; Majewski et al. 2017; Kunder et al. 2017; Buder et al. 2021). These surveys provide high-quality line-of-sight velocities, as well as abundances and stellar parameters. However, their sky coverage typically overlaps with only a small fraction of the Gaia catalog. While direct observations remain the gold standard, we show that neural network-based inference can serve a complementary role in studying the Milky Way's stellar phase-space distribution.

This Letter is organized as follows. Section 2 introduces the simulated mock Gaia catalog, and overviews the machine-learning architecture and the training procedure. Section 3 summarizes the network's success in predicting stellar line-of-sight velocities. We conclude in Section 4 and include three appendices that are referenced in the text. The neural network

used in this work is provided at the following github repository [https://github.com/adropulic/ML\\_6th\\_Dimension](https://github.com/adropulic/ML_6th_Dimension).

## 2. Methodology

In this section, we provide a description of the mock data set used in this work, followed by a detailed discussion of the neural network setup and training strategy.

### 2.1. Mock Data Catalog

To build our mock catalog, we start from the mock Gaia DR2 catalog introduced in Rybizki et al. (2018). This catalog was generated by applying a 3D dust extinction map to mock Milky Way stellar data simulated using the public code GALAXIA (Sharma et al. 2011). GALAXIA employs the Besançon Galactic model (Robin et al. 2003), which includes the bulge, thin and thick disk, as well as the stellar halo. The final mock catalog of  $\sim 1.6$  billion stars, each with full astrometric and photometric properties, was created by applying the combined 3D extinction map from Bovy et al. (2016) to model dust attenuation along with the Gaia selection criteria. Mock stars were populated by directly sampling the analytic phase-space distribution. This final point is important for our application, as the network should ideally learn the original kinematic distribution, not artifacts that may have been introduced in the sampling procedure that populates the mock catalog. Comparatively, generating mock catalogs from numerical galaxy simulations with finite stellar-mass resolution can introduce artifacts into the stellar kinematics.

For this study, we begin with the full GALAXIA mock catalog of stars<sup>6</sup> and select only those with relative parallax uncertainty  $\delta\varpi/\varpi < 0.1$  and line-of-sight velocity  $v_{\text{los}} \in [-550, 550] \text{ km s}^{-1}$ . This results in a sample of  $\sim 75$  million stars, concentrated within  $\sim 5$  kpc of the Sun. This sample only models the smooth component of the stellar halo and disk. To study the effect of kinematic substructure on the neural network learning, we supplement the GALAXIA catalog with a population of spatially uniform stars whose velocity distribution resembles that of Gaia Enceladus (Belokurov et al. 2018; Helmi et al. 2018). The Enceladus-like stars are introduced into the training, validation, and test sets by replacing 50% of stars with low-metallicity ( $[\text{Fe}/\text{H}] < -1.3$ ) in each of these sets with stars drawn from the Enceladus distribution in Necib et al. (2018).

The training and validation sets contain 7 and 1 million randomly selected stars, respectively, both with  $G < 13.5$  and  $T_{\text{eff}} \in [3550, 6900] \text{ K}$ . This is intended to parallel the subset of Gaia data with full 6D information (Katz et al. 2019). The test set consists of  $\sim 10$  million stars of any magnitude and temperature.

Approximately 0.5% of the training and validation set stars and 0.3% of the test set stars are in substructure. Only  $\sim 14\%$  of stars in the test set fall into the magnitude and effective temperature range of the training set, including the substructure stars.

### 2.2. Neural Network Architecture and Training

A feedforward neural network is trained to predict a line-of-sight velocity for each star in the input catalog, with an associated uncertainty on this prediction. The network is implemented in Keras (Chollet et al. 2015) using the

Tensorflow backend (Abadi et al. 2016). It is a combination of two halves that are structured identically except for the last layer (Figure A1 in Appendix A). This compound network structure permits two outputs: the line-of-sight velocity as well as the uncertainty on the network’s prediction. Each half of the network consists of six layers: the input, four hidden layers, and the output. The input consists of five quantities per star: Galactic longitude ( $\ell$ ), Galactic latitude ( $b$ ), proper motion in R.A. ( $\mu_{\alpha}$ ), proper motion in decl. ( $\mu_{\delta}$ ), and parallax ( $\varpi$ ). The hidden layers comprise 30 nodes each, and use a hyperbolic tangent activation function. All of the network weights are initialized with the gloriot uniform method (Glorot & Bengio 2010). Comprehensive comparisons of different network architectures were performed to determine this optimal setup (Appendix A). For example, we trained the network on only Galactic positions  $x, y, z$ , and found suboptimal results.

The output layer from one half of the network, which we will call the “velocity predictor,” consists of a single node with linear activation in order to attain a continuous value, the line-of-sight velocity. The output layer from the other half of the network, which we will call the “uncertainty predictor,” uses a ReLU activation function in order to constrain the uncertainty on the predicted line-of-sight velocity to positive values. This network is similar to the mixture density network presented in Bishop (1994).

The goal of training is to minimize the weighted Gaussian log-likelihood loss function:

$$\mathcal{L} = \sum_{i=1}^N \frac{w_i}{N} \left[ \frac{(v_{\text{los},i} - v_{\text{los},i}^{\text{pred}})^2}{(\sqrt{2} \sigma_{\text{los},i}^{\text{pred}})^2} - \log \left( \frac{1}{\sqrt{2\pi} \sigma_{\text{los},i}^{\text{pred}}} \right) \right], \quad (1)$$

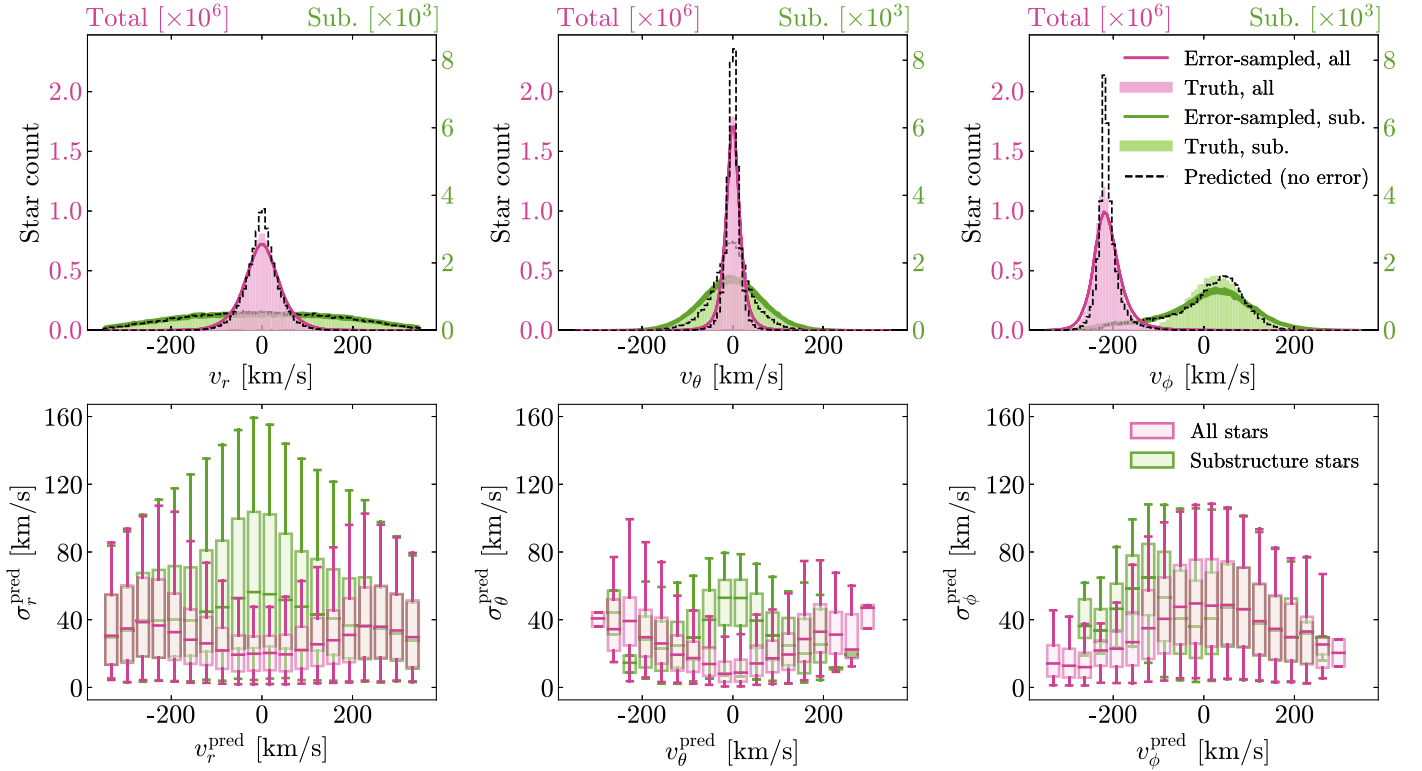
where  $v_{\text{los},i}$  is the true value of a star’s line-of-sight velocity,  $v_{\text{los},i}^{\text{pred}}$  is the network’s predicted value of a star’s line-of-sight velocity,  $w_i$  is the sample weight of a star,  $\sigma_{\text{los},i}^{\text{pred}}$  is the network’s predicted uncertainty, and  $N$  is the total number of stars. The sample weights are a function of both  $\ell_i$  and  $v_{\text{los},i}$ , and are used to force the network to learn the tails of the velocity distribution (Appendix A).

The velocity predictor is first trained using  $\sigma_{\text{los},i}^{\text{pred}} = 1$  for the loss in Equation (1). Then the uncertainty predictor is trained using the  $v_{\text{los},i}$  prediction from the velocity predictor in Equation (1) without allowing the velocity predictor to update. Finally, both the velocity predictor and the uncertainty predictor are updated simultaneously using both the predicted  $v_{\text{los},i}^{\text{pred}}$  and  $\sigma_{\text{los},i}^{\text{pred}}$  in the loss function. This produces a predicted line-of-sight velocity value and an uncertainty value per star.

During a training epoch, the stars are partitioned into batches of  $10^4$  stars, and the network is optimized on a given batch. The Adam optimizer (Kingma & Ba 2017) is used to minimize the loss function in Equation (1).

We use an initial learning rate of  $10^{-3}$  while the default values are used for the other parameters of the optimizer. If the loss computed on the validation stars does not improve for 10 epochs, the learning rate is decreased by a factor of 10, with a minimum learning rate of  $10^{-5}$ . Training is stopped when the validation loss has not improved for 40 epochs. We verified that the training and validation losses were similar to check for overfitting.

<sup>6</sup> <https://dc.zah.uni-heidelberg.de/gdr2mock/q/download/static/>



**Figure 1.** The top row shows the velocity distribution for Galactocentric velocity components. The solid pink (green) histogram is the true distribution of all (substructure) test-set stars sampled from the mock catalog; note the separate scales for the separate star counts. The network is trained on 7 million stars ( $\sim 35,000$  substructure stars) from the same catalog. The black dashed lines show the network’s predicted distribution. The dark pink (green) solid line shows the network prediction after it is Gaussian-sampled with the predicted uncertainty. The thickness of this line refers to the minimum and maximum of 50 Monte Carlo trials, described in the text. The error-sampled distribution provides a better approximation of the truth than the predicted distribution alone. In the bottom row, the predicted uncertainty on the Galactocentric velocities is shown as a function of the predicted velocity. The pink (green) boxes mark the 50% containment about the median, and the whiskers mark the 5% and 95% containment for all (substructure) stars.

### 3. Results

Next, we present the results of applying the neural network to our mock catalog. A powerful outcome of this procedure is the ability to construct an error-sampled line-of-sight velocity distribution using the network’s predicted uncertainty. We now want to quantify how well the resulting velocity distributions, as well as their correlations with each other, reproduce the truth. With the network’s prediction, we can perform a coordinate transformation on all three velocities into the Galactocentric frame ( $v_r$ ,  $v_\theta$ ,  $v_\phi$ ) and properly assign uncertainties using the procedure in Appendix B. This makes the presence of the Enceladus-like substructure more readily apparent.

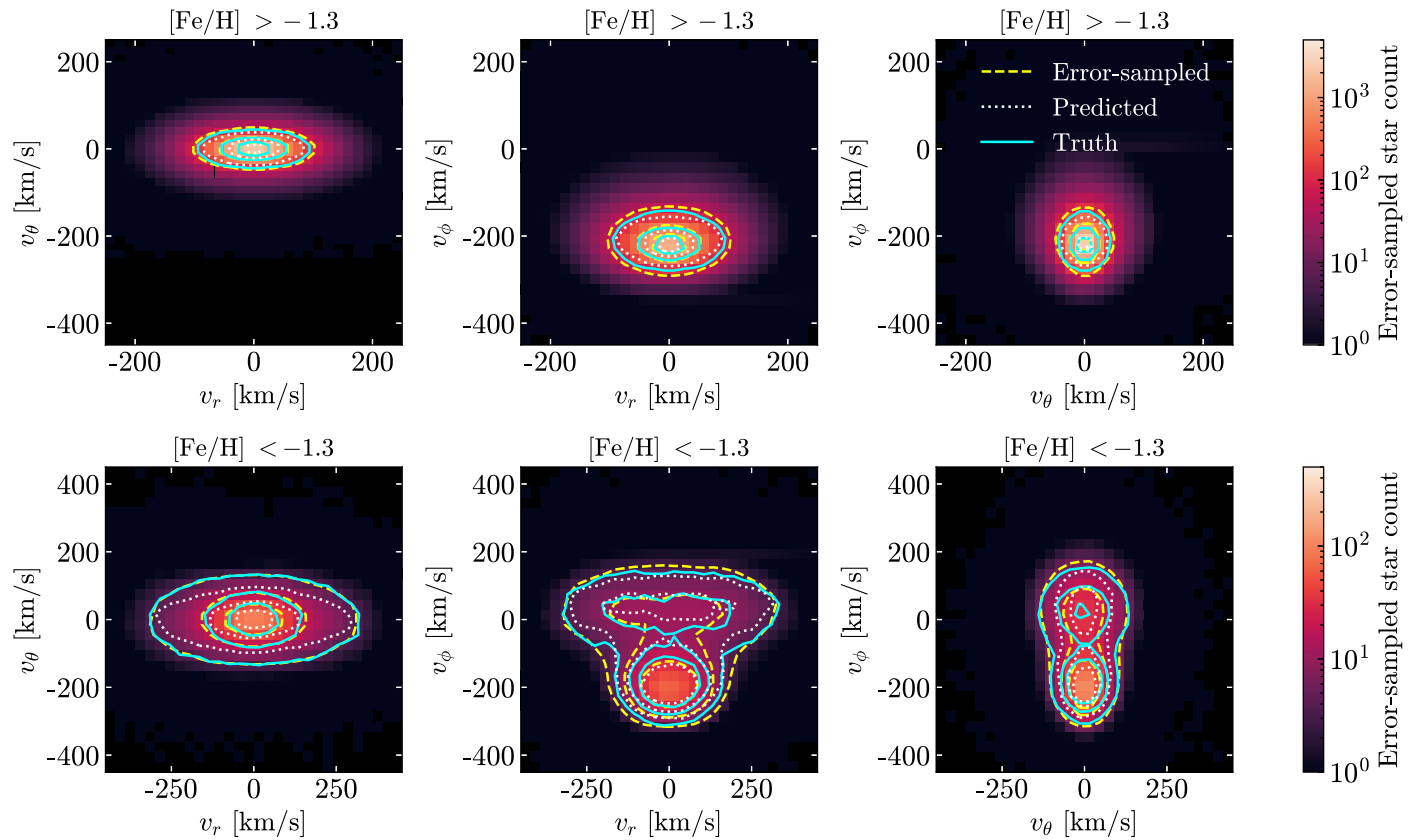
The top row in Figure 1 compares the truth, predicted, and error-sampled predicted distributions for  $v_r$ ,  $v_\theta$ ,  $v_\phi$ . The error-sampled predicted distribution is determined by the average of 50 Monte Carlo (MC) trials, which are computed by Gaussian sampling a line-of-sight velocity per star using the network’s prediction for its value and uncertainty. The sampled line-of-sight velocity along with the five known components are subsequently transformed into Galactocentric spherical coordinates. As is apparent in Figure 1, the error-sampled predicted distribution is an excellent approximation of the true distribution.

In the bottom row of Figure 1, the predicted uncertainty on the Galactocentric velocities is shown as a function of the predicted Galactocentric velocity for all stars in the test set, as well as the subset of stars that comprise the substructure. For

each velocity bin, the box denotes the 50% containment about the median uncertainty, while the whiskers denote the 5% and 95% containment. The predicted uncertainty for a given star is correlated with its spatial location and proper motion (Figure C1 in Appendix C). In general, we find that the velocity uncertainties for all stars (pink) are lowest for values of  $v_r$ ,  $v_\theta$ ,  $v_\phi$ , where there is a comparatively small contribution of substructure stars (green). This is also where the velocity uncertainties on the substructure stars are largest, as expected given that the network is forced to distinguish them from the disk stars. Indeed, we see that that the network erroneously predicts a small tail of substructure stars near  $v_\phi^{\text{pred}} \sim -200 \text{ km s}^{-1}$ .

The spread in the predicted velocity uncertainties in the Galactocentric frame can be substantial. Depending on the application, one may wish to restrict to a subset of the data with predicted uncertainties in a particular range. We find that 25% (13%) of the 10-million star test set has  $\sigma_r^{\text{pred}} \lesssim 10(5) \text{ km s}^{-1}$ , while 13% (6%) of the  $\sim 35,000$  substructure stars in the test set have  $\sigma_r^{\text{pred}} \lesssim 10(5) \text{ km s}^{-1}$ . Additionally, we find that 55% (33%) of the entire test set and 9% (4.5%) of the substructure stars in the test set have  $\sigma_\theta^{\text{pred}} \lesssim 10(5) \text{ km s}^{-1}$ , and 20% (10%) of the entire test set and 10.5% (5%) of the substructure stars in the test set have  $\sigma_\phi^{\text{pred}} \lesssim 10(5) \text{ km s}^{-1}$ . We have verified that restricting the sample of stars to those with the lowest predicted uncertainties does not bias the kinematic distributions (Figure A2 in Appendix A, and Figure C2 in Appendix C).

Figure 2 explores the extent to which the network captures the correlations among the velocity components. The top panels show the high-metallicity stars with  $[\text{Fe}/\text{H}] > -1.3$



**Figure 2.** 2D distributions are shown for the Galactocentric velocity components. The background histogram shows the network-predicted kinematic distributions of stars in the test set, sampled over the networks uncertainty prediction, to obtain the error-sampled distribution. The top row shows the stars with high-metallicity ( $[\text{Fe}/\text{H}] > -1.3$ ), and the bottom row shows the stars with low-metallicity ( $[\text{Fe}/\text{H}] < -1.3$ ) in which the stellar halo and substructure is more prominent. The contours indicate the location of 30%, 60%, and 90% containment intervals for the true (solid blue), predicted (dotted white), and error-sampled (dashed yellow) distributions. We emphasize that the correlations between metallicity and velocity are being captured by the network, even though the network has no direct access to the metallicity information.

(9,944,046 stars), and the bottom panels show the low-metallicity stars with  $[\text{Fe}/\text{H}] < -1.3$  (55,942 stars). Dividing the data set into high- and low-metallicity stars allows us to evaluate the performance of the network on disk and halo/substructure stars, respectively. To compute the density histograms in Figure 2, we take the mean of 10 (500) MC samples drawn from the high-metallicity (low-metallicity) predicted distribution; these are calculated the same way as the error-sampled distributions shown in Figure 1.

The solid blue (dashed yellow) contours in Figure 2 indicate the location of 30%, 60%, and 90% containment intervals for the true (error-sampled) distributions. For comparison, the dashed white contours correspond to the predicted distributions without sampling over the network’s error prediction; while they do a reasonable job at capturing the 2D velocity correlations, they do not exactly reproduce the truth distributions. In contrast, the error-sampled distributions nicely track the truth distributions for both the metal-poor and metal-rich samples. This is most striking for the metal-poor sample, where the velocity contours have a non-trivial shape due to contributions from the stellar disk, halo, and substructure. For example, the network captures the extended  $v_r$  distribution at the 60% containment interval in the low-metallicity  $v_r$ - $v_\phi$  plane, which is a characteristic of the Enceladus-like stars. It is important to highlight that while Figure 2 is separated into metal-rich and metal-poor categories, this is done after training; the network has no direct access to this information, and yet is

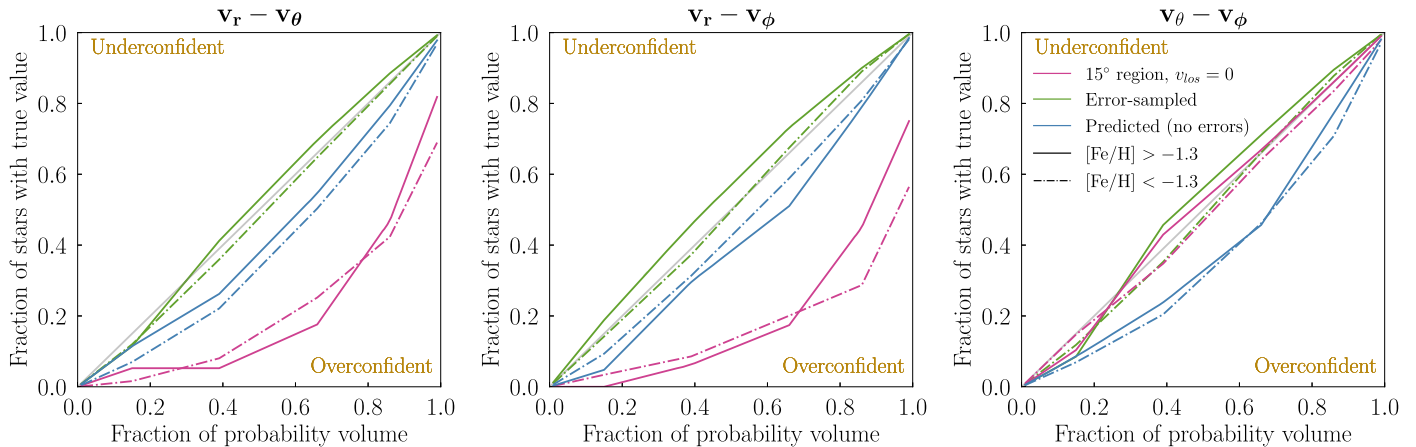
able to predict the differences in line-of-sight distributions for these categories.

Figure 3 summarizes the network’s prediction accuracy in the 3D velocity phase space. Each panel is a quantile–quantile plot that shows the fraction of stars (according to their true stellar velocities) that fall into containment intervals defined by the contours of the error-sampled and predicted distributions, examples of which are delineated in Figure 2, for different pairs of velocity coordinates. Below the gray line, the network is considered overconfident because a given containment region in the predicted distribution is too small compared to the truth expectation (i.e., the predicted  $x\%$  containment region contains  $< x\%$  of the true stars). Above the gray line, the opposite is true and the network is considered underconfident.

The solid and dashed–dotted lines in Figure 3 correspond to the  $[\text{Fe}/\text{H}] > -1.3$  and  $[\text{Fe}/\text{H}] < -1.3$  panels in Figure 2, respectively. The green lines show the results for the error-sampled distributions. We see that both the metal-poor and metal-rich distributions are well-calibrated in the  $v_r$ - $v_\phi$ ,  $v_r$ - $v_\theta$ , and  $v_\theta$ - $v_\phi$  planes. For comparison, the blue lines show the same results, but without taking into account the predicted errors. The network is more overconfident in this case, especially in the  $v_\theta$ - $v_\phi$  plane. These results reflect the qualitative features of Figure 2.

Without machine-learning techniques, reconstructing the full 6D Galactocentric phase space of stars with only 5D





**Figure 3.** Quantile–quantile plots on the neural network posteriors in Galactocentric velocity components, showing the fraction of stars with true values that fall into containment intervals defined by the contours of the error-sampled and predicted distributions. The solid green (blue) line and the dashed–dotted green (blue) line correspond to the error-sampled (predicted) contours in the metal-rich ( $[\text{Fe}/\text{H}] > -1.3$ ) and metal-poor ( $[\text{Fe}/\text{H}] < -1.3$ ) panels in Figure 2, respectively. In general, the network is well-calibrated for the error-sampled distributions (i.e., the green lines track the diagonal gray line in each panel). For comparison, we compare the network results with those of setting  $v_{\text{los}} = 0$  for stars that lie in the  $15^\circ$  region toward and away from the Galactic Center (pink lines; Koppelman et al. 2019). Restricting to this region of interest produces a stellar sample for which we are relatively sure of  $v_\theta$  and  $v_\phi$ , as reflected in the calibration in the  $v_\theta - v_\phi$  (third panel). However, the neural network yields a much better calibration in the  $v_r - v_\phi$  and  $v_r - v_\theta$  planes.

information can only be achieved under certain restrictive conditions. For example, Koppelman et al. (2019) proposed setting  $v_{\text{los}} = 0$  for stars that lie in the region defined by  $\sqrt{\ell^2 + b^2} \leq 15^\circ$  or  $\sqrt{(\ell - 180^\circ)^2 + b^2} \leq 15^\circ$ . In this region,  $v_\theta$  and  $v_\phi$  roughly correspond to the proper motion of stars, and therefore can be reconstructed reasonably accurately, with limited dependence on  $v_{\text{los}}$ . Restricting to this region, we have 508,798 metal-rich stars and 1504 metal-poor stars. We then compare the predicted velocity distributions, obtained by simply setting  $v_{\text{los}} = 0 \text{ km s}^{-1}$ , to the truth distribution; however, it is important to note that there is no principled way to assign errors to  $v_{\text{los}}$  with this prescription. As seen in Figure 3, the results are well-calibrated and even outperform the network in the  $v_\theta - v_\phi$  plane, where this method is designed to excel. We can see, however, that the same method is overconfident in the  $v_r - v_\phi$  plane and  $v_r - v_\theta$  plane for both the metal-rich and metal-poor subsets. Therefore, our neural network approach has the benefit of being applicable beyond a narrow spatial region, and can additionally predict uncertainties.

#### 4. Conclusions

In this Letter, we have demonstrated that a neural network can successfully predict a star’s line-of-sight velocity and associated uncertainty from 5D astrometric inputs, after being trained on a subset of data with complete 6D phase-space information. We trained, tested, and validated the network on mock data that contained a disk, stellar halo, and Enceladus-like substructure. The error-sampled network prediction successfully reproduced the individual velocity distributions, as well as their correlations. The final results also captured the expected metallicity dependence of the velocity distributions, even though metallicity was not provided as an input to the network. The network successfully reconstructed the velocities of the substructure stars, even though they comprised only  $\sim 0.5\%$  of the training set. This result demonstrates that the network is learning more than just the bulk motion of the stars.

A critical feature of the network design is its ability to provide an uncertainty on its velocity prediction. The predicted

velocity distributions more reliably reproduced the true distributions when properly sampled over these errors. The mean network uncertainty on the predicted line-of-sight velocity was  $\sigma_{\text{los}}^{\text{pred}} = 38 \text{ km s}^{-1}$ , with about 8%(1%) of all stars in the test set having  $\sigma_{\text{los}}^{\text{pred}} \lesssim 30(20) \text{ km s}^{-1}$ . We stress that these uncertainties should not be directly compared to the *measured* errors on Gaia’s line-of-sight velocities. In particular,  $\sigma_{\text{los}}^{\text{pred}}$  is the network’s *predicted* uncertainty on its projected value of line-of-sight velocity—for instance, a correct network prediction for  $v_{\text{los}}$  can still be associated with a large uncertainty. However, we do find that stars with smaller  $\sigma_{\text{los}}^{\text{pred}}$  are typically associated with more accurate velocity predictions. Furthermore, restricting to a subset of stars with these smaller uncertainties does not bias the overall velocity distributions.

The simple idea of regressing missing kinematic information can be accomplished by a variety of architecture choices. For example, Bayesian neural networks (Mackay 1995; Bhat & Prosper 2005; Neal 1996; Bai et al. 2016; Gal & Ghahramani 2016; Blei et al. 2017; Bollweg et al. 2020; Charnock et al. 2020; Wagner-Carena et al. 2021) may provide an alternative method for incorporating uncertainties on the network output. In this case, the network would be rerun many times over the same inputs, while the weights float within some prior distribution. Since these approaches require a more sophisticated neural network architecture, we leave a detailed comparison for future work.

We optimized the neural network architecture by applying it to a mock Gaia catalog. In practice, the network will be trained on the subset of Gaia data with complete 6D phase space, eliminating any systematic uncertainty on the network output associated with using simulated data. Our training set will expand with Gaia DR3, which will provide line-of-sight velocities for an additional  $\sim 30$  million stars (Gaia Collaboration et al. 2021). The success of our machine-learning approach motivates further studies on other potential applications, such as recovering individual stellar streams and/or multiple substructures that overlap in phase space.

Thus far, we have trained and tested the network on stars concentrated within 5 kpc of the Sun, but it would also be

beneficial to adapt the method presented here to apply to stars farther out in the halo. This extension would be particularly relevant for mapping the potential of the Milky Way halo and identifying dwarf galaxies and other substructures, as just two important examples. For current Gaia data releases, the primary challenge is the lack of a sufficient training sample, limited by the number of stars at these distances with complete 6D information. One possibility is to train the network on a subset of Gaia stars whose line-of-sight velocities come from cross-matches to other spectroscopic surveys. Whether this approach will succeed depends on both the number of cross-matched stars and whether the selection function of the spectroscopic surveys complicates the network inference. If successful, this machine-learning approach can potentially be adapted to the upcoming Rubin Observatory.

This Letter shows that there is often incredible information waiting to be extracted from high dimensional correlations, which can be used to maximize the utility of real world data.

The authors acknowledge H. Koppelman, P. Melchior, S. Mishra-Sharma, L. Necib, O. Slone, and N. Weiner for fruitful conversations. M.L. gratefully acknowledges financial support from the Schmidt DataX Fund at Princeton University made possible through a major gift from the Schmidt Futures Foundation. B.O. was supported in part by the U.S. Department of Energy (DOE) under contract DE-SC0013607 and DE-SC0020223. L.J.C. was supported in part by a Paul & Daisy Soros Fellowship and the NSF GRFP Award Number DGE-1656466. T.C. is supported by the DOE under Award Number DE-SC0011640. H.L. and M.L. are supported by the DOE under Award Number DE-SC0007968. This work is supported by the National Science Foundation under Cooperative Agreement PHY-2019786 (The NSF AI Institute for Artificial Intelligence and Fundamental Interactions, <http://iaifi.org/>). The work presented in this Letter was performed on computational resources managed and supported by Princeton Research Computing. This research made use of the `astropy` (Robitaille et al. 2013), `corner` (Foreman-Mackey 2016), `h5py` (Collette 2013), `IPython` (Perez & Granger 2007), `Jupyter` (Kluyver et al. 2016), `matplotlib` (Hunter 2007), `NumPy` (van der Walt et al. 2011), `pandas` (Wes McKinney 2010), and `SciPy` (Jones et al. 2001) software packages.

## Appendix

In Appendix A, we elaborate on the details of our network architecture, as well as its optimization. In Appendix B, we explain how to transform the line-of-sight velocity and uncertainty provided by the network to Galactocentric velocities and associated uncertainties. Finally, in Appendix C, we provide several supplemental figures that are referenced in the main text.

### Appendix A Network Information

#### A.1. Tested Network Architectures

In this appendix, we elaborate on the procedure used to select the network architecture described in Section 2.2. The final configuration, illustrated in Figure A1, was chosen after extensive tests that varied the number of hidden layers, the number of nodes per layer, the activation function associated with each layer, the sample weights, and the input parameters.

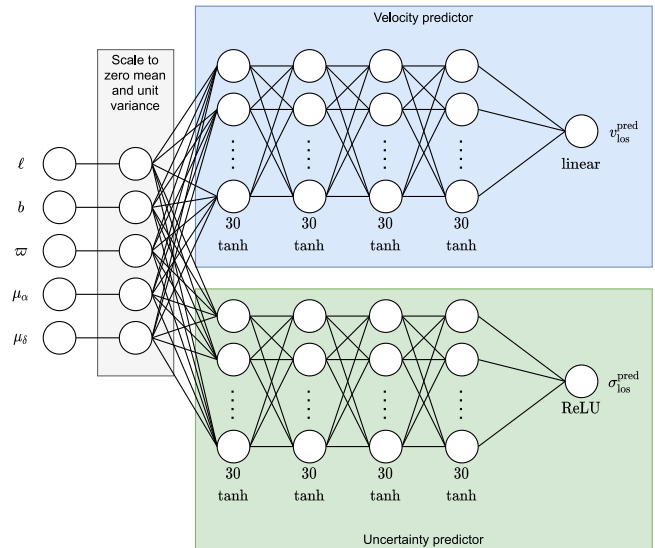


Figure A1. Diagram of the neural network used in this work.

We tested network architectures under combinations of the following specifications: 1–5 hidden layers, 30–2000 nodes per layer, ReLU/ELU/Tanh activation functions on the interior layers, and different combinations of input parameters. Deeper (more layers) and wider (more nodes per layer) networks do not yield a significantly more accurate regression, and conversely, smaller and narrower networks both perform a less accurate regression. Neither the ReLU activation function nor the ELU activation function predict the velocity distribution as accurately as the Tanh activation function.

In order to evenly sample the tails of the  $v_{\text{los}}$  distribution in our data set, we implement sample weighting to de-emphasize the contribution of stars with  $v_{\text{los}} \sim 0 \text{ km s}^{-1}$ . The simplest way to effectively “flatten” the peak of the  $v_{\text{los}}$  distribution is to take the  $v_{\text{los}}$  distribution itself to be the reciprocal of the weighting function. In practice, we histogram the  $v_{\text{los}}$  values in each data set, then interpolate the histogram to extract the probability  $p_i$  corresponding to a given star’s value of  $v_{\text{los},i}$  and take  $w_i = \frac{1}{p_i}$  to be the weight for that star. We refer to this as the “linear weights” prescription, to contrast with the “logarithmic weights” prescription described below.

We can also implement a weighting scheme that is a less-steep function of  $v_{\text{los},i}$  than  $\frac{1}{p_i}$ . To do so, we define the “logarithmic weights” as  $w_i = \log(1/p_i) + w_0$ , where  $w_0$  is chosen so that  $w_i$  is positive for all stars. In practice, we find that linear weights in  $v_{\text{los}}$  overly de-emphasize the central  $v_{\text{los}}$  values, while logarithmic weights alleviate this issue. Given the dependence of  $v_{\text{los}}$  on Galactic longitude,  $\ell$ , we also implement linear and logarithmic weights on the joint distribution of  $v_{\text{los}}$  and  $\ell$ . Of the four weighting schemes tested, the joint  $v_{\text{los}}-\ell$  logarithmic weights method is the most effective at recovering the truth distributions. We thus choose this to be our fiducial method for sample weighting.

We also tested a network that included the information from each star’s 10 nearest neighbors during training.

Because this additional information does not improve the regression, at least for the data set considered here, we do not adopt it for the fiducial method. Lastly, we compared our combined network structure, described in Section 2.2, to a

single network with two outputs, and found that the latter does not learn or train well.

### A.2. Network Diagnostics

The network architectures described above in Section B.1 were evaluated using the Kullback–Leibler (KL) divergence, as well as the  $R^2$  and  $\chi^2$  test statistics. These statistics are defined as follows:

$$\begin{aligned} \mathcal{D}_{\text{KL}}(\mathbf{p}||\mathbf{q}) &= \sum_s p_s \log_2 \left( \frac{p_s}{q_s} \right), \\ R^2 &= 1 - \sum_{i=1}^N \left( \frac{v_{\text{los},i} - v_{\text{los},i}^{\text{pred}}}{v_{\text{los},i} - \bar{v}_{\text{los}}} \right)^2, \text{ and} \\ \chi^2 &= \frac{1}{N-1} \sum_{i=1}^N \left( \frac{v_{\text{los},i} - v_{\text{los},i}^{\text{pred}}}{\sigma_{\text{los},i}^{\text{pred}}} \right)^2, \end{aligned} \quad (\text{A1})$$

where  $v_{\text{los},i}$ ,  $v_{\text{los},i}^{\text{pred}}$ , and  $\sigma_{\text{los},i}^{\text{pred}}$  are defined as in Equation (1), and  $\bar{v}_{\text{los}}$  is the mean of the true line-of-sight velocity distribution. Also,  $\mathbf{p}$ ,  $\mathbf{q}$  are the values of the kernel density estimation of the true and predicted distributions, respectively, evaluated at each histogram bin value  $s \in [-350, 350] \text{ km s}^{-1}$  in Figure 1.

The KL divergence is a measure of how one probability distribution differs from a second, reference probability distribution (Kullback & Leibler 1951). A small value of the KL divergence  $\mathcal{D}_{\text{KL}}$  indicates very good agreement between the true and predicted distributions. In other words, if the probability for an event from  $\mathbf{p}$  is large, but the probability for the same event in  $\mathbf{q}$  is small, there will be a large contribution to the statistic. The  $R^2$  test statistic provides a measure of how well the truth values are replicated by the predicted values, without including the predicted uncertainty measure,  $\sigma_{\text{los}}^{\text{pred}}$ . The  $R^2$  statistic generally ranges from 0 to 1, but can be negative if the fit is very poor.  $R^2 = 0$  is obtained if the predicted velocity is always the mean velocity, while  $R^2 = 1$  indicates a completely accurate prediction for all stars. The  $\chi^2$  test statistic is a test of the uncertainty prediction,  $\sigma_{\text{los}}^{\text{pred}}$ . In general, if  $\chi^2 \gg 1$ , then the predicted uncertainty is too small. Conversely, if  $\chi^2 \ll 1$ , then the predicted uncertainty is too large. We note that each predicted uncertainty is approximately Gaussian. That is, the quantity  $(v_{\text{los},i} - v_{\text{los},i}^{\text{pred}})/\sigma_{\text{los},i}^{\text{pred}}$  for the  $i^{\text{th}}$  star is approximately Gaussian-distributed over the full sample.

We trained and tested the different network architectures on a smaller data set than the one ultimately used in this work and analyzed these architectures using the test statistics in Equation (A1). We then calculated the test statistics for each case, evaluating how sensitive the results are to various cuts that span  $\sigma_{\text{los}}^{\text{pred}} < 30 \text{ km s}^{-1}$  to  $\sigma_{\text{los}}^{\text{pred}} < 150 \text{ km s}^{-1}$ . The results are provided in Table A1; all networks included in this table utilize the logarithmic sample weights on the joint distribution of  $v_{\text{los}}$  and  $\ell$ .

Dropout is a technique in which randomly selected nodes are ignored when the network is trained. When not using dropout and including the proper motions in the input, the results are not sensitive to cuts on the predicted uncertainty (i.e., the range of the test statistic is small). This is very desirable because the goodness-of-fit does not depend on the particular selection of stars. However, the networks with dropout have a wide range in the diagnostic metrics as cuts on  $\sigma_{\text{los}}^{\text{pred}}$  are applied. Using such

**Table A1**

Network Architectures Studied in This Work, and the Range of Test Statistics Obtained for Each Case when Restricting the Data Set by Making Various Cuts that Span  $\sigma_{\text{los}}^{\text{pred}} < 30 \text{ km s}^{-1}$  to  $\sigma_{\text{los}}^{\text{pred}} < 150 \text{ km s}^{-1}$

Input Variables	Dropout	KL	$\chi^2$	$R^2$
$\ell, b, \varpi, \mu_\alpha, \mu_\delta, x, y, z$	No	0.008–0.010	0.699–0.749	0.720–0.746
$\ell, b, \varpi, \mu_\alpha, \mu_\delta$	No	0.008–0.010	0.685–0.739	0.704–0.721
$x, y, z$	No	0.009–0.035	0.578–0.779	0.681–0.738
$\ell, b, \varpi, \mu_\alpha, \mu_\delta, x, y, z$	Yes	0.002–0.986	0.245–4.214	0.713–0.857
$\ell, b, \varpi, \mu_\alpha, \mu_\delta$	Yes	0.017–0.075	0.348–0.789	0.712–0.757
$x, y, z$	Yes	0.008–0.096	0.314–3.040	0.651–0.818

**Note.** The test statistics are defined in Equation (A1). Our fiducial model is the second row of the table.

networks thus requires extra calibration to understand predictions when departing from the nominal set of stars.

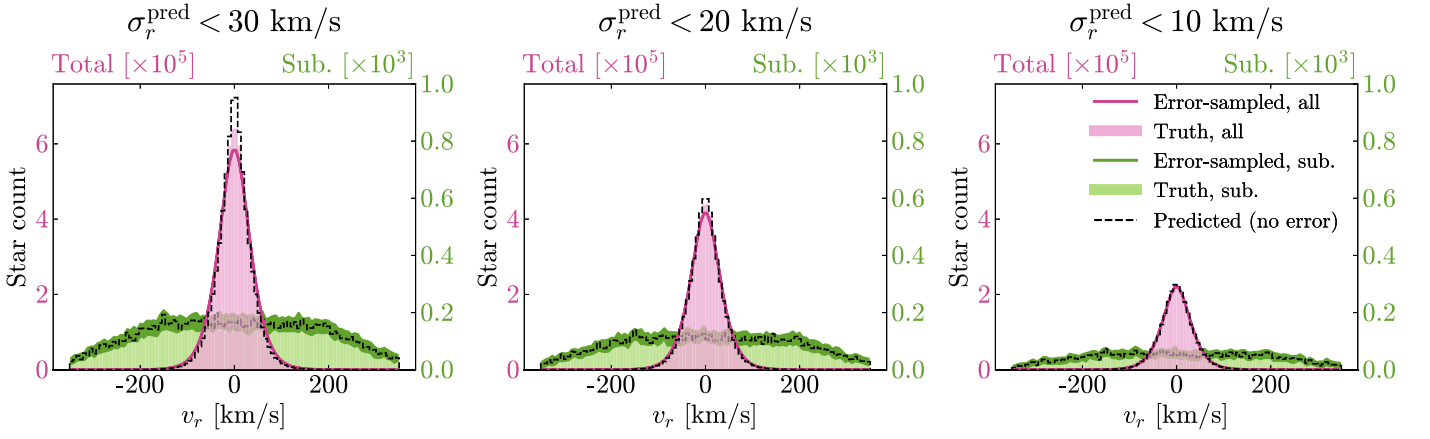
In the case without dropout, the test statistics and ranges for the  $(\ell, b, \varpi, \mu_\alpha, \mu_\delta, x, y, z)$  and  $(\ell, b, \varpi, \mu_\alpha, \mu_\delta)$  configurations are very similar, suggesting that including the redundant  $(x, y, z)$  inputs does not aid the network.

In general, the  $(x, y, z)$  network does worse overall, especially when using dropout. This provides a strong indication that the network is learning more than just the bulk stellar motion, because it uses more than just the location in the Galaxy.

When dropout is included, the network does better on the initial predictions (i.e., the  $R^2$  and KL divergence have better scores than without dropout at one end of the range), but the spread of the test statistics is still very large. Also, when looking at  $\chi^2$  for these models, we see that the estimate of the uncertainty is less accurate, and the spread of these values is much greater.

After running numerous tests on network architectures, activation functions, input data, and weighting schemes, we ultimately selected a network with 4 hidden layers with 30 nodes each and without dropout. We only include the 5D coordinates  $\ell, b, \varpi, \mu_\alpha, \mu_\delta$  as input because the extra variables do not significantly help with the regression.

As described in the main text, Figure 1 highlights the significance of including predicted uncertainty by comparing the truth, predicted, and error-sampled distributions of  $v_r$ , the Galactocentric radial velocity. The test statistics described above in Section B.2 were calculated for the  $v_r$  histograms containing all test stars. When  $\sigma_r^{\text{pred}}$  is included, the KL divergence is 0.009, compared to 0.016 when  $\sigma_r^{\text{pred}}$  is not included. Figure A2 shows the same Galactocentric radial velocity histograms as in Figure 1, but selecting stars with  $\sigma_r^{\text{pred}} < 30, 20, 10 \text{ km s}^{-1}$ . The KL divergence between the truth and error-sampled distributions decreases from 0.005 when  $\sigma_r^{\text{pred}} < 30 \text{ km s}^{-1}$ , to 0.002 when  $\sigma_r^{\text{pred}} < 20 \text{ km s}^{-1}$ , to  $< 0.001$  when  $\sigma_r^{\text{pred}} < 10 \text{ km s}^{-1}$ . Making increasingly stronger cuts on uncertainties in a certain velocity component can be used to obtain increasingly more accurate distributions in that component. It is also possible to make an uncertainty cut on two coordinates, for example  $v_r$  and  $v_\phi$ , in pursuit of a high-purity sample of Enceladus-like stars.



**Figure A2.** The lines depicted in this figure are the same as those in Figure 1. However, this figure specifically focuses on the Galactocentric radial velocity  $v_r$ , to show that the distribution is unbiased as more restrictive cuts on  $\sigma_r^{\text{pred}}$  are made.

The predicted  $v_r$  values are also evaluated using the  $\chi^2$  and  $R^2$  test statistics. The value of  $\chi^2$  for the test set is 0.68 and remains stable when cuts on  $\sigma_r^{\text{pred}}$  are made, from 0.679 when  $\sigma_r^{\text{pred}} < 30 \text{ km s}^{-1}$ , to 0.663 and 0.641 when  $\sigma_r^{\text{pred}} < 20$  and  $10 \text{ km s}^{-1}$ , respectively. The value of  $R^2$  is 0.72. As cuts on  $\sigma_r^{\text{pred}}$  are made, the value of  $R^2$  approaches 1.0, from 0.862 when  $\sigma_r^{\text{pred}} < 30 \text{ km s}^{-1}$ , to 0.937 and 0.985 when  $\sigma_r^{\text{pred}} < 20$  and  $10 \text{ km s}^{-1}$ , respectively. This again highlights the usefulness of the network’s uncertainty prediction in attaining accurate velocity predictions. The correlations for the set of stars with  $\sigma_r^{\text{pred}} < 10 \text{ km s}^{-1}$  are shown in Figure C2, in which the truth, predicted, and error-sampled distributions align very well in  $v_r$ . As expected, we see that the truth and error-sampled distributions across all velocity pairs align better overall.

### Appendix B Transforming Velocities and Uncertainties

In this appendix, we derive the affine transformation that maps proper motions and line-of-sight velocity to Galactocentric velocities. As a corollary, we also obtain the transformation mapping velocity uncertainties between these respective coordinate systems.

Consider a star with equatorial coordinate velocities, given by  $[v_{\text{los}}, \dot{\alpha}d \cos \delta, \dot{\delta}d]^\top$ , where  $\alpha$  and  $\delta$  are R.A. and decl., respectively,  $v_{\text{los}}$  is the line-of-sight velocity and  $d$  is the distance of the star from the Sun. The dot represents a derivative with respect to time. With respect to previously defined variables in the text, we have  $\mu_\alpha = \dot{\alpha}d \cos \delta$ ,  $\mu_\delta = \dot{\delta}d$ , and  $d = 1/\varpi$ . Following Bovy (2011), equatorial coordinate velocities can be transformed into Galactic coordinates velocities via

$$\begin{aligned} \begin{bmatrix} v_{\text{los}} \\ \dot{\ell}d \cos b \\ \dot{b}d \end{bmatrix} &= \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \psi & \sin \psi \\ 0 & -\sin \psi & \cos \psi \end{pmatrix} \\ \begin{bmatrix} v_{\text{los}} \\ \dot{\alpha}d \cos \delta \\ \dot{\delta}d \end{bmatrix} &\equiv \mathbf{P} \begin{bmatrix} v_{\text{los}} \\ \dot{\alpha}d \cos \delta \\ \dot{\delta}d \end{bmatrix}, \end{aligned} \quad (\text{B1})$$

where  $\psi \simeq 78.3433$  is the Galactic parallactic angle.

Next, we want to convert velocities in Galactic coordinates into Galactocentric coordinates. To do this, we begin by

writing down the relation between Cartesian coordinates in each coordinate system. Consider a star with Cartesian Galactocentric coordinates  $\mathbf{r}_{\text{GC}}$ ; its coordinates  $\mathbf{r}_{\text{g}}$  in Cartesian Galactic coordinates is given by

$$\mathbf{r}_{\text{g}} = \mathbf{M}_{\odot}^{-1}(\mathbf{r}_{\text{GC}} - \mathbf{R}_{\odot, \text{GC}}), \quad (\text{B2})$$

where  $\mathbf{R}_{\odot, \text{GC}} = (x_{\odot, \text{GC}}, 0, z_{\odot, \text{GC}})^\top$  is the Cartesian Galactocentric coordinates of the Sun (we adopt the standard convention that the Sun lies in the  $xz$ -plane in these coordinates, with the Galactic plane lying in the  $xy$ -plane). The matrix  $\mathbf{M}_{\odot}$  rotates the Galactic coordinate axes such that they become parallel to the Galactocentric coordinate axes, and is written explicitly as

$$\mathbf{M}_{\odot}^{-1} = \begin{pmatrix} \cos \beta & 0 & -\sin \beta \\ 0 & 1 & 0 \\ \sin \beta & 0 & \cos \beta \end{pmatrix}, \quad (\text{B3})$$

where  $\beta \equiv \arctan(z_{\odot, \text{GC}}/x_{\odot, \text{GC}})$ . Taking the derivative of the coordinates, we find that

$$\dot{\mathbf{r}}_{\text{g}} = \mathbf{M}_3 \mathbf{M}_4 \begin{bmatrix} v_{\text{los}} \\ \dot{\ell}d \cos b \\ \dot{b}d \end{bmatrix}, \quad (\text{B4})$$

with

$$\begin{aligned} \mathbf{M}_3 &= \begin{pmatrix} \cos \ell & -\sin \ell & 0 \\ \sin \ell & \cos \ell & 0 \\ 0 & 0 & 1 \end{pmatrix}, \\ \mathbf{M}_4 &= \begin{pmatrix} \cos b & 0 & -\sin b \\ 0 & 1 & 0 \\ \sin b & 0 & \cos b \end{pmatrix}. \end{aligned} \quad (\text{B5})$$

Similarly,

$$\dot{\mathbf{r}}_{\text{GC}} = \mathbf{M}_2^{-1} \mathbf{M}_1^{-1} \begin{bmatrix} v_r \\ v_\phi \\ v_\theta \end{bmatrix}, \quad (\text{B6})$$



with

$$\begin{aligned} \mathbf{M}_2^{-1} &= \begin{pmatrix} \cos \phi & -\sin \phi & 0 \\ \sin \phi & \cos \phi & 0 \\ 0 & 0 & 1 \end{pmatrix}, \\ \mathbf{M}_1^{-1} &= \begin{pmatrix} \cos \theta & 0 & -\sin \theta \\ 0 & 1 & 0 \\ \sin \theta & 0 & \cos \theta \end{pmatrix}. \end{aligned} \quad (\text{B7})$$

Finally, taking the derivative of Equation (B2) and using the expressions derived above, we arrive at the following transformation between Galactic and Galactocentric velocities:

$$\begin{bmatrix} v_r \\ v_\phi \\ v_\theta \end{bmatrix} = \mathbf{M} \begin{bmatrix} v_{\text{los}} \\ \dot{a}d \cos b \\ \dot{b}d \end{bmatrix} + \mathbf{M}_1 \mathbf{M}_2 \mathbf{v}_{\odot, \text{GC}}, \quad (\text{B8})$$

where we have defined  $\mathbf{M} = \mathbf{M}_1 \mathbf{M}_2 \mathbf{M}_\odot \mathbf{M}_3 \mathbf{M}_4$ , while  $\mathbf{v}_{\odot, \text{GC}}$  is the Cartesian velocity of the Sun in the Galactocentric frame. Combining this with Equation (B1), we obtain the full transformation mapping equatorial coordinate velocities to Galactocentric coordinate velocities:

$$\begin{bmatrix} v_r \\ v_\phi \\ v_\theta \end{bmatrix} = \mathbf{MP} \begin{bmatrix} v_{\text{los}} \\ \dot{\alpha}d \cos \delta \\ \dot{\delta}d \end{bmatrix} + \mathbf{M}_1 \mathbf{M}_2 \mathbf{v}_{\odot, \text{GC}}. \quad (\text{B9})$$

This result has been verified by comparison with coordinate transformations implemented by *Astropy* (Robitaille et al. 2013).

We now turn our attention to the transformation of uncertainties. Assuming Gaussian uncertainties, the differential probability  $P(\mathbf{v}_{\text{true}}; \mathbf{v}_{\text{m}})$  that the true velocity of a star is  $\mathbf{v}_{\text{true}}(v_{\text{los}}, \mu_\alpha, \mu_\delta)$  given a measured velocity  $\mathbf{v}_{\text{m}}(v_{\text{los}}, \mu_\alpha, \mu_\delta)$  is proportional to

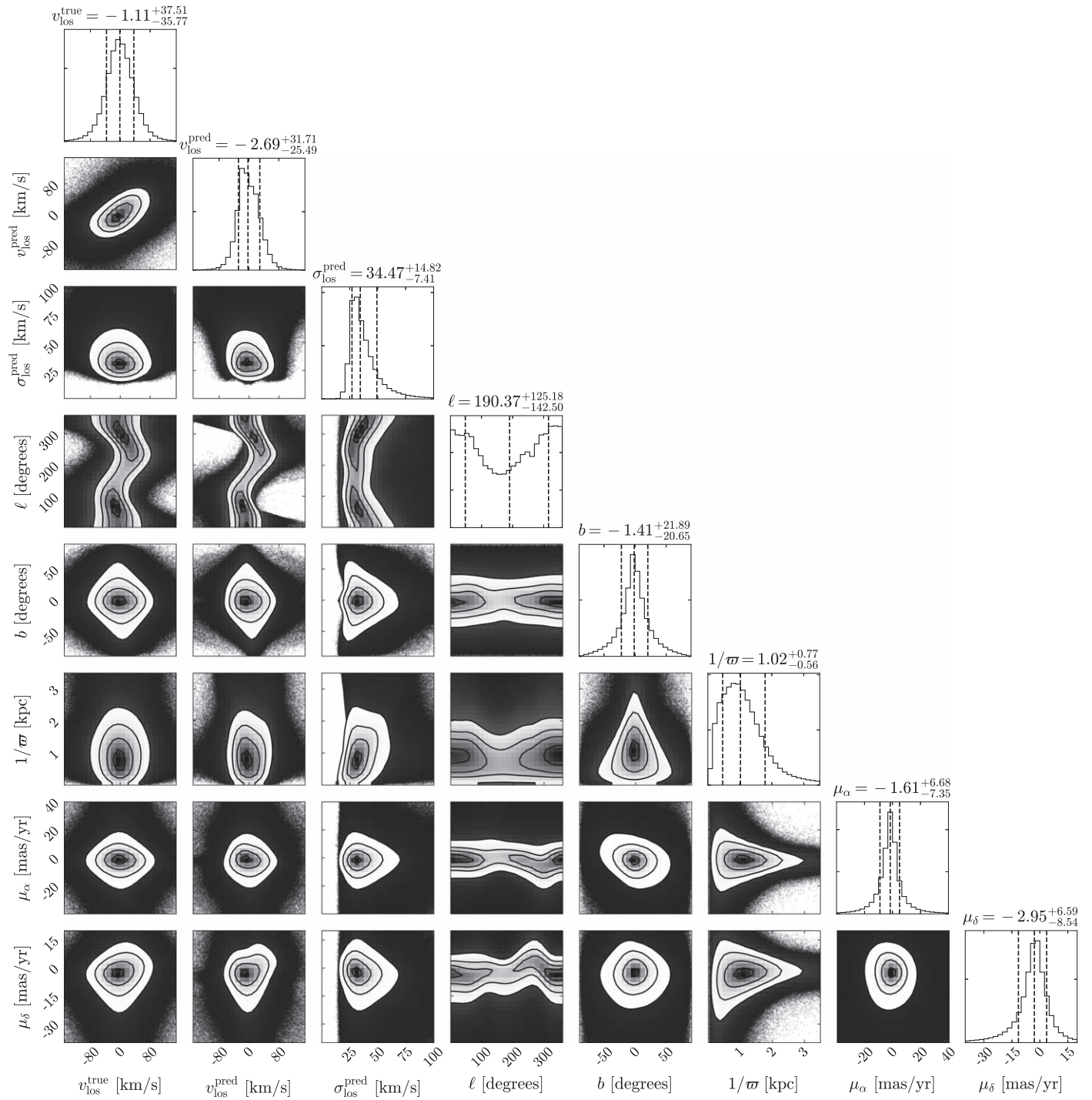
$$P(\mathbf{v}_{\text{true}}; \mathbf{v}_{\text{m}}) \propto \exp \left[ -\frac{1}{2} (\mathbf{v}_{\text{true}} - \mathbf{v}_{\text{m}})^T \mathbf{S}^{-1} (\mathbf{v}_{\text{true}} - \mathbf{v}_{\text{m}}) \right], \quad (\text{B10})$$

where  $\mathbf{S}(v_{\text{los}}, \mu_\alpha, \mu_\delta)$  is the covariance matrix in the basis of equatorial coordinate velocities. Since the differential probability is invariant under coordinate transformations, we can see immediately that the covariance matrix in the basis of Galactocentric velocities can be obtained by

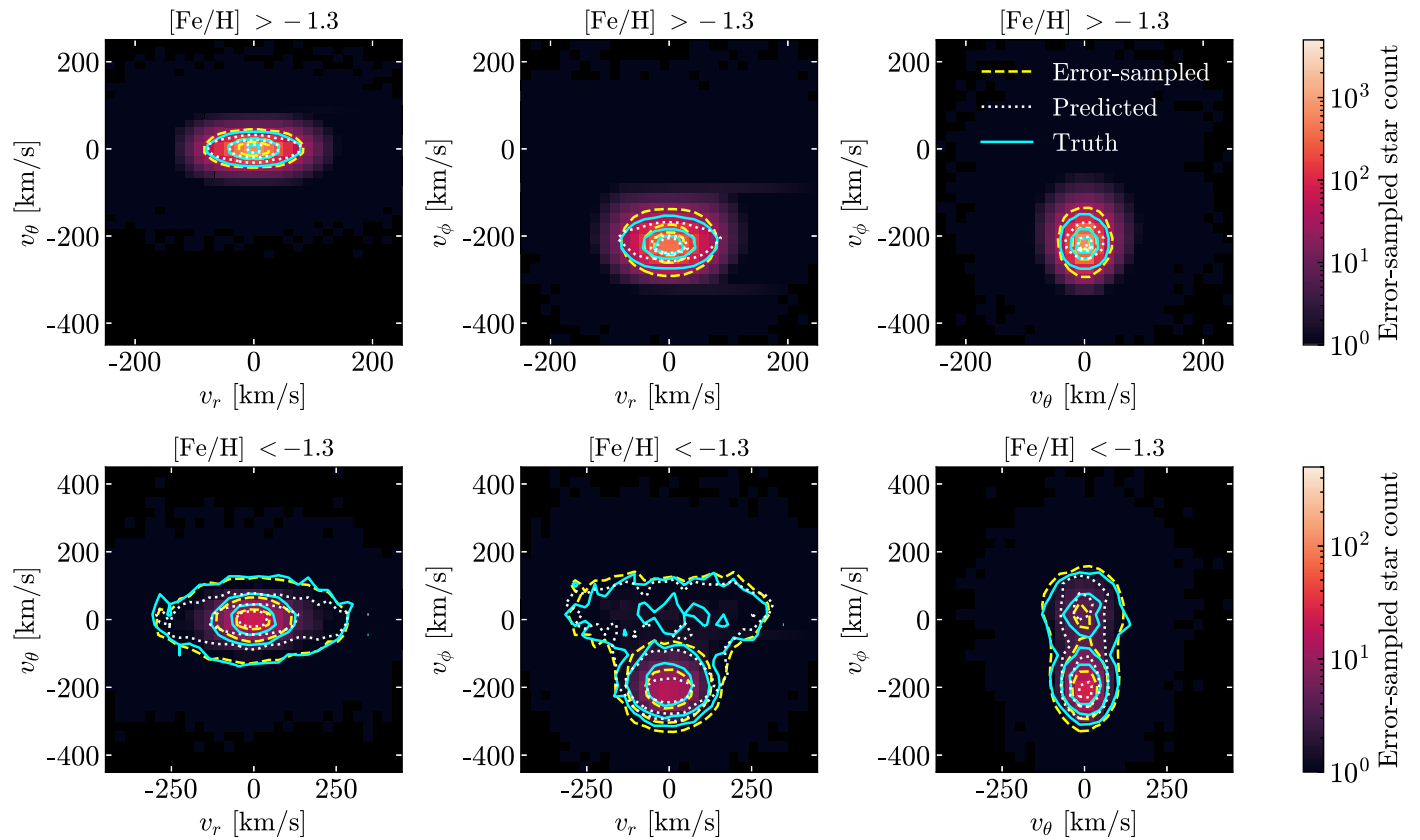
$$\mathbf{S}(v_r, v_\phi, v_\theta) = \mathbf{MP} \cdot \mathbf{S}(v_{\text{los}}, \mu_\alpha, \mu_\delta) \cdot \mathbf{P}^{-1} \mathbf{M}^{-1}. \quad (\text{B11})$$

## Appendix C Supplementary Figures

In this appendix, we provide some figures that supplement the discussion in the main text.



**Figure C1.** Corner plot for all stars in the test set for the following quantities: true line-of-sight velocity ( $v_{\text{los}}^{\text{true}}$ ), predicted line-of-sight velocity ( $v_{\text{los}}^{\text{pred}}$ ), predicted uncertainty on light-of-sight velocity ( $\sigma_{\text{los}}^{\text{pred}}$ ), Galactic longitude and latitude ( $l, b$ ), inverse parallax ( $1/\varpi$ ), and proper motions ( $\mu_{\alpha, \delta}$ ).



**Figure C2.** The background histogram and contours depicted in this figure are the same as those in Figure 2. However, this figure shows the resulting error-sampled background histogram and contours for the subset of stars with  $\sigma_r^{\text{pred}} < 10 \text{ km s}^{-1}$ . There are 2,490,330 stars with  $[\text{Fe}/\text{H}] > -1.3$  and 9,316 stars with  $[\text{Fe}/\text{H}] < -1.3$ .

### ORCID iDs

Adriana Dropulic <https://orcid.org/0000-0002-7352-6252>  
 Bryan Ostdiek <https://orcid.org/0000-0002-0376-6461>  
 Laura J. Chang <https://orcid.org/0000-0001-8590-2043>  
 Hongwan Liu <https://orcid.org/0000-0003-2486-0681>  
 Timothy Cohen <https://orcid.org/0000-0002-7040-3038>  
 Mariangela Lisanti <https://orcid.org/0000-0002-8495-8659>

### References

- Abadi, M., Agarwal, A., Barham, P., et al. 2016, arXiv:1603.04467  
 Bai, Y., Xu, Y., Pan, J., Lan, J. Q., & Gao, W. W. 2016, *JInst*, **11**, P07006  
 Belokurov, V., Erkal, D., Evans, N. W., Koposov, S. E., & Deason, A. J. 2018, *MNRAS*, **478**, 611  
 Bhat, P. C., & Prosper, H. B. 2005, Conf. Proc. C, 050912, 151  
 Bishop, C. M. 1994, [https://publications.aston.ac.uk/id/eprint/373/1/NCRG\\_94\\_004.pdf](https://publications.aston.ac.uk/id/eprint/373/1/NCRG_94_004.pdf)  
 Blei, D. M., Kucukelbir, A., & McAuliffe, J. D. 2017, *Journal of the American Statistical Association*, **112**, 859  
 Bollweg, S., Haufmann, M., Kasieczka, G., et al. 2020, *ScPP*, **8**, 006  
 Bovy, J. 2011, PhD Thesis, New York Univ.  
 Bovy, J., Rix, H.-W., Green, G. M., Schlafly, E. F., & Finkbeiner, D. P. 2016, *ApJ*, **818**, 130  
 Buder, S., Sharma, S., Kos, J., et al. 2021, *MNRAS*, in press (doi:10.1093/mnras/stab1242)  
 Bullock, J. S., & Johnston, K. V. 2005, *ApJ*, **635**, 931  
 Charnock, T., Perreault-Levasseur, L., & Lanusse, F. 2020, Bayesian Neural Networks, arXiv:2006.01490  
 Chollet, F., Rahman, F., Zhu, Q. S., et al. 2015, Keras, <https://keras.io/>  
 Collette, A. 2013, Python and HDF5 (Sebastopol, CA: O'Reilly Media)  
 Cui, X.-Q., Zhao, Y.-H., Chu, Y.-Q., et al. 2012, *RAA*, **12**, 1197  
 Foreman-Mackey, D. 2016, *JOSS*, **1**, 24  
 Gaia Collaboration, Brown, A. G. A., Vallenari, A., et al. 2021, *A&A*, **649**, A1  
 Gaia Collaboration, Prusti, T., de Bruijne, J. H. J., et al. 2016, *A&A*, **595**, A1  
 Gaia Collaboration, Katz, D., Antoja, T., et al. 2018, *A&A*, **616**, A11  
 Gal, Y., & Ghahramani, Z. 2016, ICML, 1050  
 Gilmore, G., Randich, S., Asplund, M., et al. 2012, *Msngr*, **147**, 25  
 Glorot, X., & Bengio, Y. 2010, PMLR, **9**, 249  
 Helmi, A., Babusiaux, C., Koppelman, H. H., et al. 2018, *Natur*, **563**, 85  
 Hunter, J. D. 2007, *CSE*, **9**, 90  
 Johnston, K. V. 1998, *ApJ*, **495**, 297  
 Johnston, K. V., Hernquist, L., & Bolte, M. 1996, *ApJ*, **465**, 278  
 Jones, E., Oliphant, T., Peterson, P., et al. 2001, SciPy: Open source scientific tools for Python, <http://www.scipy.org/>  
 Katz, D., Sartoretti, P., Cropper, M., et al. 2019, *A&A*, **622**, A205  
 Kingma, D. P., & Ba, J. 2017, arXiv:1412.6980[cs]  
 Kluyver, T., Ragan-Kelley, B., Pérez, F., et al. 2016, Positioning and Power in Academic Publishing: Players, Agents and Agendas (Amsterdam: IOS Press)  
 Koppelman, H. H., Helmi, A., Massari, D., Roelenga, S., & Bastian, U. 2019, *A&A*, **625**, A5  
 Kullback, S., & Leibler, R. A. 1951, *The Annals of Mathematical Statistics*, **22**, 79  
 Kunder, A., Kordopatis, G., Steinmetz, M., et al. 2017, *AJ*, **153**, 75  
 Mackay, D. J. C. 1995, *NCNS*, **6**, 469  
 Majewski, S. R., Schiavon, R. P., Frinchaboy, P. M., et al. 2017, *AJ*, **154**, 94  
 Neal, R. M. 1996, Bayesian Learning for Neural Networks, Vol. 118 (New York: Springer-Verlag)  
 Necib, L., Lisanti, M., & Belokurov, V. 2018, *ApJ*, **874**, 3  
 Nix, D. A., & Weigend, A. S. 1994, in Proc. 1994 IEEE Int. Conf. on Neural Networks (ICNN'94) (New York: IEEE), 55  
 Perez, F., & Granger, B. E. 2007, *CSE*, **9**, 21  
 Robin, A. C., Reylé, C., Derrière, S., & Picaud, S. 2003, *A&A*, **409**, 523  
 Robitaille, T. P., Tollerud, E. J., Greenfield, P., et al. 2013, *A&A*, **558**, A33  
 Rybizki, J., Demleitner, M., Fousneau, M., et al. 2018, *PASP*, **130**, 074101  
 Sharma, S., Bland-Hawthorn, J., Johnston, K. V., & Binney, J. 2011, *ApJ*, **730**, 3  
 van der Walt, S., Colbert, S. C., & Varoquaux, G. 2011, *CSE*, **13**, 22

Wagner-Carena, S., Park, J. W., Birrer, S., et al. 2021, [ApJ](#), **909**, 187  
Wes McKinney 2010, in Proc. 9th Python in Science Conf., ed.  
S. van der Walt & J. Millman (SciPy), 56

White, S. D. M., & Rees, M. J. 1978, [MNRAS](#), **183**, 341  
Wilkinson, M. I., Vallenari, A., Turon, C., et al. 2005, [MNRAS](#), **359**, 1306  
Yanny, B., Rockosi, C., Newberg, H. J., et al. 2009, [AJ](#), **137**, 4377