Department of Physics and Astronomy
University of Heidelberg

Master Thesis in Physics
submitted by

Justin Mathew

born in Iritty, India

2024

Use of machine learning for predicting stellar encounters

This Master Thesis has been carried out by Justin Mathew at the
Max Planck institute for Astronomy
under the supervision of
Prof. Dr. Coryn Bailer-Jones

Zusammenfassung

In dieser Arbeit wird die Möglichkeit der Verwendung neuronaler Netze zur Vorhersage naher Begegnungen mit der Sonne anhand der aktuellen Phasenraumkoordinaten von Sternen aus Gaia DR3 untersucht. Die Bahnintegration von Sternen zusammen mit den Surrogaten, die aus der Unsicherheit generiert werden, macht es rechnerisch so aufwendig, nahe Begegnungen zu finden. Je mehr Quellen mit radialen Geschwindigkeiten in zukünftigen Gaia-Versionen enthalten sind, desto anspruchsvoller wird der Prozess. Da die meisten Sterne einer annähernd linearen Umlaufbahn folgen, bis sie ihre Position der nahen Begegnung erreichen, kann ein neuronales Netzwerk die anfänglichen Phasenraumkoordinaten und ihre entsprechenden Unsicherheiten nehmen und dann die Positionen der nahen Begegnung zusammen mit ihren Unsicherheiten und Korrelationen zurückgeben. Das neuronale Netz wurde anhand von 2.951.517 Sternen innerhalb eines Bereichs von 1 kpc um die Sonne trainiert und sagte erfolgreich Zeit, Entfernung und Geschwindigkeit der nahen Begegnung mit ihren Unsicherheiten und Korrelationen voraus. Das Netzwerk erreichte einen mittleren absoluten Fehler von 1,7 pc für die Entfernung der Begegnung, 81 kyr für die Zeit der Begegnung und 0,27 km/s für die Geschwindigkeit der Begegnung. Für Sterne innerhalb von 100 pc von der Sonne beträgt die Streuung von d_{ph}, t_{ph} und v_{ph} jedoch 0,44 pc, 67,3 kyr bzw. 0,33 km/s, während der MAD von d_{ph}, t_{ph} und v_{ph} für diese Quellen 18,1 pc, 761 kyr bzw. 16,39 km/s beträgt. Allerdings nimmt der Vorhersagefehler des erzeugten Netzwerks zu größeren Entfernungen und kleineren Entfernungen von weniger als 10 pc hin insgesamt zu.

Abstract

This work explores the possibility of using neural networks to predict close encounters with the Sun from the present phase space coordinates of stars from Gaia DR3. Orbital integration of stars together with the surrogates that are generated from the uncertainty makes it computationally so expensive to find close encounters. As more sources with radial velocities are included in future Gaia releases, the process will become even more demanding. Since most stars follow an approximately linear orbit until they reach their close encounter position, a neural network can take the initial phase space coordinates and their corresponding uncertainties, then return the close encounter positions along with their uncertainties and correlations. The neural network was trained using 2,951,517 stars within a 1 kpc range of the Sun and successfully predicted close encounter time, distance, and velocities with their uncertainties and correlations. The network achieved a median absolute error of 1.7 pc for encounter distance, 81 kyr for encounter time, and 0.27 km/s for encounter velocity. But, for stars within 100 pc of the Sun, the scatter of d_{ph}, t_{ph} and v_{ph} is 0.44 pc, 67.3 kyr, and 0.33 km/s respectively. whereas the MAD of the d_{ph}, t_{ph} and v_{ph} for these sources is 18.1 pc, 761 kyr, and 16.39 km/s respectively. However, the network produced has an overall increase in prediction error towards higher distances and smaller distances of less than 10 pc.

Acknowledgements

First and foremost, I would like to thank my supervisor Prof. Coryn Bailer-Jones for welcoming me into his group and allowing me to explore statistics, machine learning, and astronomy. His patience in listening to my ideas, detailed explanations, and the time he dedicated to discussing solutions to complex problems were significant in making this thesis possible. Then, I would like to thank the group members Sara Jamal, Rene Andrae, and Morgan Fouesneau for their valuable feedback in the group meetings.

Then I would like to thank my family for supporting my passion for learning astronomy. A special thanks to my friends Agnus Johnson, Johnly Joshy, Vijaylakshmi Vijayakumaran Nair, and Yash Gurbani for giving their valuable feedback and support that helped me complete this thesis. Furthermore, I would like to thank Stefan Nellen for reviewing my German translation of the abstract. I also like to thank Jahnvi Golatkar, Pundari Kavipurapu for their support and for making the final days of my thesis submission fun.

I am also grateful to Prof. Björn Malte Schäfer for kindly agreeing to be my second supervisor.

Contents

List of Figures

List of Tables

1 Introduction

Stellar encounters with the Sun, often referred to as close stellar passages or flybys, are dynamic events in which another star passes near our Solar System, potentially influencing its structure and evolution. These encounters, though rare on human timescales, play significant roles in the broader context of galactic dynamics and the history of the Solar System. Throughout the history of the solar system, some stars come close to the Sun or pass through the outer edge of Oort cloud which is the distant spherical shell of icy bodies surrounding our Solar System. The Oort cloud is extended up to 0.5 pc. So, any star that passes within 1 pc of the Sun is considered as a close stellar encounter. The star's ability to perturb the Oort cloud depends on the mass, velocity, and distance of the star at perihelion. Therefore, some stars more than 1 pc away might also have a huge impact on the Oort cloud depending on its mass. A disturbance in the Oort cloud could push a lot of comets toward the inner solar system. This can increase the probability of comet impact on the Earth, which can result in dramatic changes to Earth's biosphere. For instance, there have been studies that show the Chicxulub impact that led to the Cretaceous-Tertiary extinction event that occurred 66 Myr ago was caused by long-period comets Siraj and Loeb (2021). Further studies that simulate the origin of long-period comets have been done by including stellar encounters Vokrouhlickỳ et al. (2019).

Studies regarding stellar encounters were first done by J. Oort in 1950. His studies suggested that the gravitational influence of passing stars could dislodge comets from the Oort Cloud, sending them toward the inner Solar System Oort (1950). Empirical investigations into stellar encounters were significantly advanced by high-precision astrometric missions such as Hipparcos, launched by the European Space Agency (ESA) in 1989, and more recently, the Gaia mission. The Gaia mission provides precise phase space coordinates information on 33,812,183 stars. These missions along with modern computing power enable astronomers to identify and study past and future close encounters with greater accuracy. García-Sánchez et al. (2001) utilized Hipparcos data to catalog stars that had or would come close to the Sun, establishing a foundational dataset for understanding these events. Later by the advent of computer simulations and high-precision astrometry, these encounters were studied with

much more accuracy and helped in confirming the possibility of the role of nearby stars in perturbing the orbit as well as the stability of planetary system. The passage of WISE J072003.20-084651.2 (Scholz's Star) within 0.25 pc of the Sun around 70,000 years ago is a prime example of a stellar encounter that likely caused significant perturbations to Oort Cloud. The recent study on stellar encounters by Bailer-Jones (2022) using Gaia DR3 data has analyzed 33 million stars and estimated 61 of these stars have passed or will pass within 1 pc of the Sun in $\pm$ 6 Myr. From this, the K7 dwarf Gl 710 remains the closest known encounter, with an estimated (median) encounter distance of 0.0636 pc to take place in 1.3 Myr. Unlike Scholz's Star, Gliese 710 is massive (0.6 $M_{\odot}$) and can have a significant perturbation in the Oort cloud. In addition, these encounters can also affect the climate of the Earth. Kaib and Raymond (2024) studied how the effect of passing stars should be included in the reconstruction climate evolution of Earth. Passing stars accelerated the perturbations in Earth's orbit and reduced the reliability of Earth's climate prediction by 10%. Including effects of encounters like HD 7977 which occurred 2.8 Myr ago gives a new understanding of the Earth's past orbital evolution beyond 50 Myr. The other reason to study close stellar encounters is that it helps us to assess the possibility of a nearby supernova that could become a threat to life on Earth. A supernova within 10-20 light-years could be harmful for life on Earth. The discovery of $^{60}Fe_2$ in the ocean bed which formed from an event 2.8 Myr ago is evidence for the influence of supernovas in Earth's atmosphere. Studies have shown that this supernova responsible for the $^{60}Fe_2$ happened within the 10 pc range of the Sun Knie et al. (2004).

Although there is a significant number of studies about stellar encounters, modeling these encounters is still a complex task due to a large number of gravitational interactions and also due to the computational expense of the orbital integration. For example, in the study Bailer-Jones (2015), all the samples were not subjected to high-resolution numerical integration. A linear motion was assumed for stars and the corresponding close encounter distance (d_{ph}^{lin}) was calculated (the subscript 'ph' denotes perihelion). The high-resolution numerical integration was only performed on sources that achieved $d_{ph}^{lin} < 10$ pc. There might be some stars if numerically integrated might have achieved a close encounter distance of less than 10 pc in the $d_{ph}^{lin} > 10$ pc range. So, there is a chance to miss these close encounter objects. Additionally, the future

Gaia data release is expected to have the full phase space coordinates of a lot of objects with much more accuracy. This demands even more computational power for finding close encounters with the Sun. Here is where the application of machine learning methods comes into play. This study investigates alternative techniques like the use of Feedforward Neural Networks (FNN) for predicting stellar encounters. The FNN can be trained to create a mapping between input phase space coordinates and output close encounter parameters of a star. The Neural Network will be able to predict the encounter parameters with their uncertainties. This can be used to create a distribution of the star's close encounter parameters such as close encounter time t_{ph}, close encounter distance d_{ph}, and close encounter velocity v_{ph}. This study also opens the door for exploring other machine-learning techniques to find close stellar encounters.

2 Theory & Methods

2.1 Sample selection from Gaia

The data used for training the Neural Network was generated from Gaia DR3. The Gaia mission by ESA launched in 2013 is tasked with producing precise three-dimensional maps of stars in the Milky Way and beyond. It is positioned at the L2 Lagrange point and measures the position, velocity, luminosity, temperature, and composition of stars. Gaia monitors each of its target stars about 14 times per year and tracks their position, velocities, and changes in brightness. These positions and velocities are calculated in ICRS frames's also expected to discover hundreds of thousands of new celestial objects, such as extra-solar planets and brown dwarfs, and observe hundreds of thousands of asteroids within our own Solar System.

Gaia DR3 contains 33,653,049 sources with complete astrometric solutions. From this, I only selected sources with parallax_over_error greater than 5, which are in the 1 kpc range of the Sun. There were parallax with a low signal-to-noise ratio was ignored because less precise parallax can significantly affect the accuracy of calculated encounter parameters Bailer-Jones (2022). This selection resulted in 12,044,415 sources. Then, zero point correction for these sources was done using the procedure from Lindegren et al. (2021). Parallax zero-point correction is the adjustment needed to account for the biases in parallax measurements provided by Gaia. It depends on factors like position, brightness, and color of the star. After parallax zero-point correction, there are 12,043,873 sources. The reason for missing 542 sources was because they do not phot_g_mean_mag values. So these sources were ignored in this study. After parallax correction, the overall raw parallax of sources was increased by 0.0346 mas.

It should also be noted that the observed radial velocity of the sources is not equal to the true radial velocity. Other than measurement or calibration errors, the gravitational redshift increases the observed radial velocity. According to a study by Gullberg and Lindegren (2002), the observed minus true radial velocity ranges from -0.4 km/s for F stars to +0.4 km/s for K dwarfs. However, these are less than Gaia DR3 radial velocity uncertainties. So, no corrections were made for radial velocities.

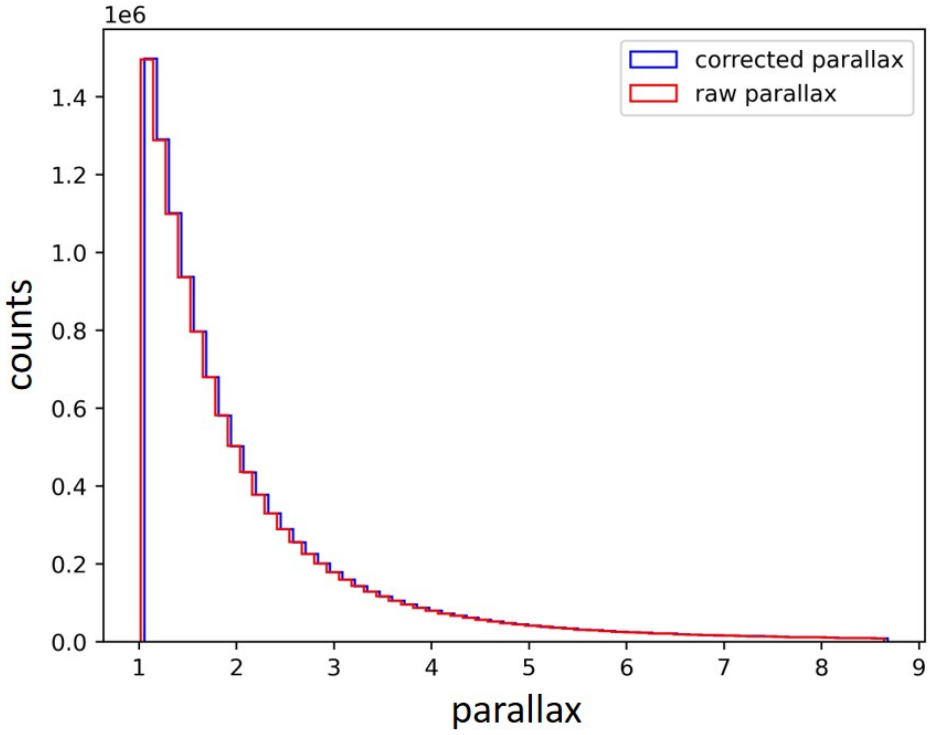


Figure 1: Normalised distribution of parallax before (red) and after zero point correction (blue). The raw parallax was increased by 0.04-0.03 mas (central 98% range). The raw parallax of the main sample covers a range of 1.01-8.64 mas (central 98%).

Finally, these sources were subjected to orbital integration as described in 2.2. Due to technical difficulties, the orbital integrations could only be completed for 4,938,031 sources.

2.2 Numercal methods for orbital integration

Various numerical techniques are available for the orbital integration. To achieve an accurate orbit for a source, the initial position, velocity, and appropriate step size should be chosen. A larger step size results in less accurate results, but the computational expense is smaller. On the other hand, if we require much more accurate results, smaller time steps should be used. So, achieving an accurate result is a balance between the computational expense and the limit of precision we need for our study. Some of the common numerical techniques used for orbital integration are the Euler method, Leapfrog method, and Runge-Kutta method.

The Euler method is one of the simple methods for numerically solving differential equations. The idea of Euler's method is to use local linearity to join multiple small line segments and hence they can approximate the actual solution. In the case of the orbiting body, if x_n, v_n are the position and velocity of the body in the n-th time step, then

$$x_{n+1} = x_n + v_n \Delta t \tag{2.1}$$

$$v_{n+1} = v_n + a_n \Delta t \tag{2.2}$$

is the updated position and time of the body. Here, a_n is the acceleration of the body in the n-th time step which is calculated from the gravitational potential equation (refer section 2.2.1). It should be also noted that numerical solutions can also be unstable. The instability of a numerical solution is the situation in which the numerical solution diverges away from the exact solution. As an example, consider the equation $y' = \alpha y$ where α is constant. The exact solution is $y(t) = y_0 e^{\alpha t}$. Applying Euler's method to this with step size h gives the following equation

$$y_{n+1} = y_n + h\alpha y_n \tag{2.3}$$

To ensure the stability of the numerical solution, the step size h should satisfy the condition $|1 + h\alpha| \leq 1$. For this particular example, if $\alpha = -2.7$, then the value of $h \leq 1$ for satisfying the condition $|1 + h\alpha| \leq 1$. From figure 2, we can see how the solution diverges and becomes unstable for larger step sizes. So, to avoid such situations, it is important to pick the optimum step size.

The range-kutta method is another numerical method that approximates solu-

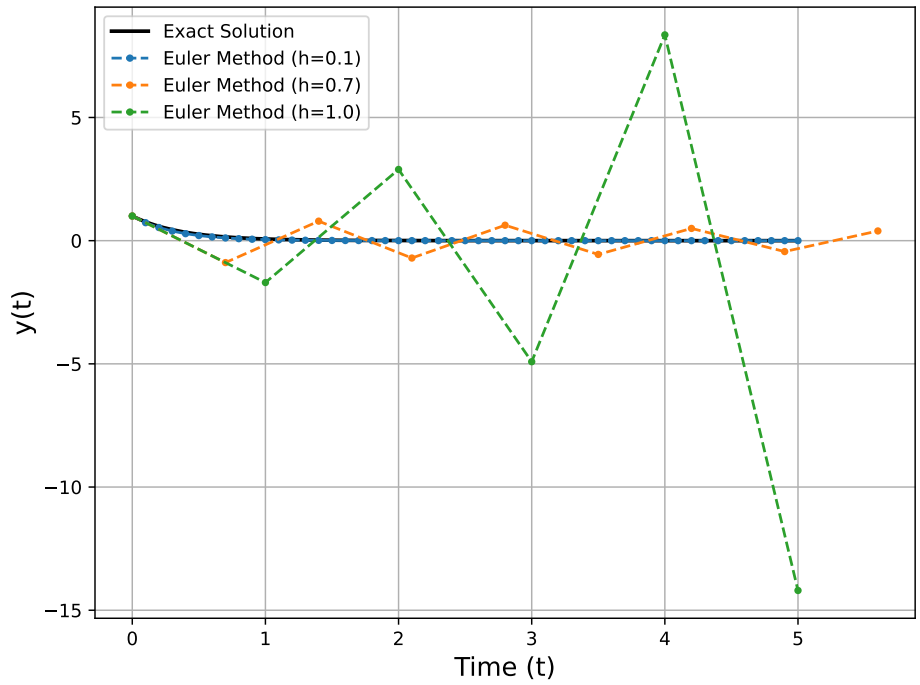


Figure 2: The plot shows the instability of the numerical solution from the exact solution $y(t) = y_0 e^{\alpha t}$ with $\alpha = -2.7$ for the differential equation $y' = \alpha y$ as the step size is increased. The solution should approach zero as time goes to infinity. But, when $h > 0.7$ the solution oscillates and diverges from zero.

tions in four different points in a single interval. The fourth-order runge-kutta method (RK4) is mostly used for its computational efficiency and accuracy. This method was developed by Carl Runge and Wilhelm Kutta around 1900. Suppose our differential equation is of the form,

$$y' = f(t, y) \tag{2.4}$$

Let the initial condition be $y(t_0) = y_0$. Then, the solution is approximated by

$$y_{n+1} = y_n + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) \tag{2.5}$$

in which k_1, k_2, k_3 and k_4 is given by

$$\begin{aligned} k_1 &= f\left(t_n, y_n\right) \\ k_2 &= f\left(t_n + \frac{h}{2}, y_n + \frac{h}{2}k_1\right) \\ k_3 &= f\left(t_n + \frac{h}{2}, y_n + \frac{h}{2}k_2\right) \\ k_4 &= f\left(t_n + h, y_n + hk_3\right) \end{aligned} \tag{2.6}$$

where h is the step size. The RK4 has a local truncation error of order $O(h^5)$ and a global truncation error of $O(h^4)$. The local truncation error is the amount of error introduced in a single step of the numerical method when approximating the differential equation solution. Global truncation error is the cumulative error caused by the large number of steps. For Euler's method, this local truncation error is $O(h^2)$ while the global truncation error is of order $O(h)$ Vuik et al. (2023). Figure 3 gives the comparison numerical solution of a differential equation $y' = \alpha y$ for both RK4 and Euler's method. The stability and accuracy of RK4 method make it a good choice for our orbital integration.

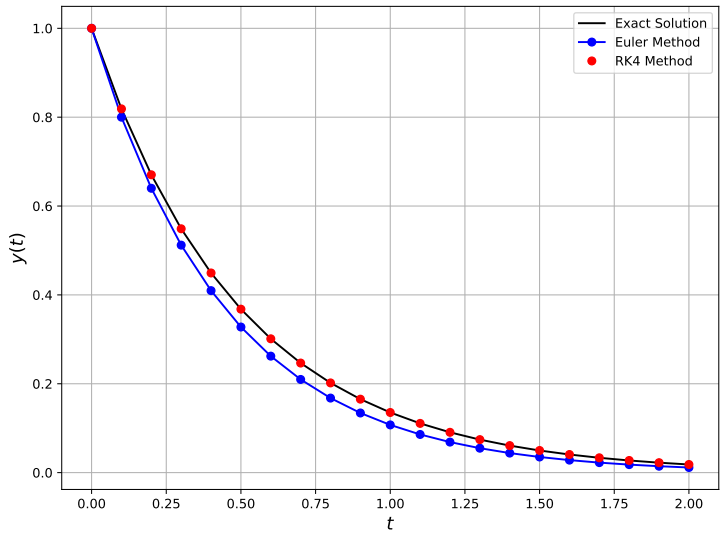


Figure 3: The plot compares the accuracy of Euler method and RK4 method for first order differential equation $y' = \alpha y$, with step size h=0.1 and α = -2.7. The RK4 method gives better accuracy than the Euler method.

For this work, I used the RK45 method also known as the Runge-Kutta-

Fehlberg method for numerical integration in Scipy Virtanen et al. (2020). The RK45 method uses adaptive step size, that is, the step size is adjusted dynamically based on estimated error. For our application, there is no aggressive variation of solution since all the stars travel through weak gravitational fields. The main application for our case is, that it controls local truncation error and is computationally efficient by estimating fourth and fifth-order solutions at the next step and compares the error between them to adjust the step size. This allows the method to take large steps in areas without rapid changes reducing overall steps needed, making it computationally more efficient. Suppose we are given an ODE $\dot{y} = f(t, y)$ with initial condition $y(t_0) = y_0$. The RK45 algorithm calculates six k-values,

$$\begin{aligned}
k_1 &= hf(t_n, y_n), \\
k_2 &= hf\left(t_n + \frac{1}{4}h, y_n + \frac{1}{4}k_1\right), \\
k_3 &= hf\left(t_n + \frac{3}{8}h, y_n + \frac{3}{32}k_1 + \frac{9}{32}k_2\right), \\
k_4 &= hf\left(t_n + \frac{12}{13}h, y_n + \frac{1932}{2197}k_1 - \frac{7200}{2197}k_2 + \frac{7296}{2197}k_3\right), \\
k_5 &= hf\left(t_n + h, y_n + \frac{439}{216}k_1 - 8k_2 + \frac{3680}{513}k_3 - \frac{845}{4104}k_4\right), \\
k_6 &= hf\left(t_n + \frac{1}{2}h, y_n - \frac{8}{27}k_1 + 2k_2 - \frac{3544}{2565}k_3 + \frac{1859}{4104}k_4 - \frac{11}{40}k_5\right)
\end{aligned} \tag{2.7}$$

From these k-values, fourth and fifth-order estimates are calculated.

$$y^4_{n+1} = y_k + \frac{25}{216}k_1 + \frac{1408}{2565}k_3 + \frac{2197}{4101}k_4 - \frac{1}{5}k_5 \tag{2.8}$$

$$y^5_{n+1} = y_k + \frac{16}{135}k_1 + \frac{6656}{12,825}k_3 + \frac{28,561}{56,430}k_4 - \frac{9}{50}k_5 + \frac{2}{55}k_6 \tag{2.9}$$

The optimal step size is then calculated by multiplying scalar s times the step size h. The scalar s is,

$$s = \left(\frac{\varepsilon h}{2|y^5_{n+1} - y^4_{n+1}|}\right)^{1/4} \tag{2.10}$$

Here ε is the error tolerance. That is, if the difference between y^5_{n+1} and y^4_{n+1}

is greater than ε, then reduce the step size, and if it is less than ε increase the step size. Thus, speeding up the computation speed Mathews et al. (2004). The comparison of computational speed for Euler's method, RK method, and RK45 method is given in figure 4. The average run time of the three methods is compared for numerically solving the ODE $y' = 2y$. The RK4 method is most time-consuming compared to Euler and RK45. The RK45 takes less time and is more accurate than Euler's method. With these advantages in mind,

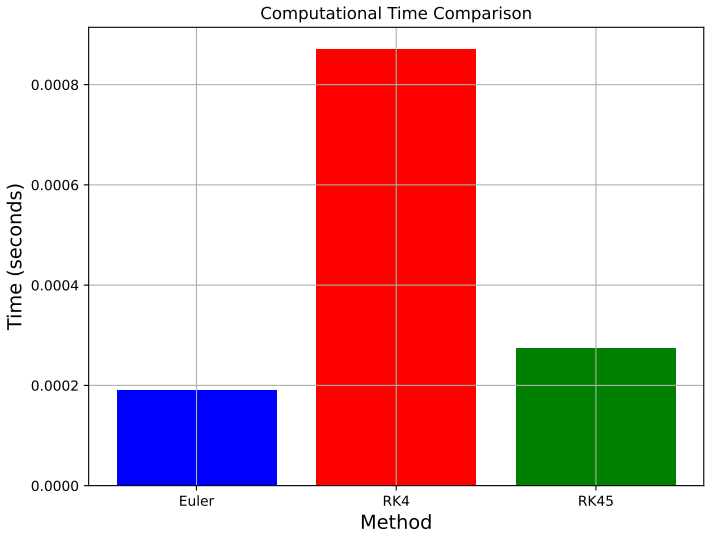


Figure 4: The comparison between the average run time of Euler, RK4, and RK45 method with the y-axis showing time in seconds.

the RK45 method was finalised for numerically integrating the orbits of stars from Gaia data for creating the training set for the neural network.

2.2.1 Choosing the Galactic Potential

The Galactic potential is one important function for calculating the orbit of stars in the galaxy. The galactic potential refers to the gravitational field produced by the distribution of stars, gas, and dark matter within the galaxy. This potential governs the motion of particles and stars inside the galaxy. The force exerted on a mass m at a position $\mathbf{r}$ is the sum of forces exerted by all the mass distributed in the galaxy. Then the force is calculated by equation,

$$\mathbf{F}(\mathbf{r}) = -m\nabla\Phi \tag{2.11}$$

where Φ is the potential. From this acceleration, $\mathbf{a}$ of the mass m can be calculated. The gravitational potential is derived from the Poisson's equation,

$$\nabla^2\Phi(\mathbf{r}) = 4\pi G\rho(\mathbf{r}) \tag{2.12}$$

where G is the gravitational constant and $\rho(\mathbf{r})$ is the mass density distribution. There exist various models for gravitational potential Φ of the galaxy. The simplest of them is the Plummer model Plummer (1911) which describes a spherically symmetric mass distribution. This is often used to describe globular clusters and central bulges of galaxies. The potential $\Phi(r)$ is given by,

$$\Phi(r) = \frac{GM}{\sqrt{r^2 + b^2}}. \tag{2.13}$$

Here, M is the total mass of the system and b is a constant that has the dimension of length. The most popular model used for characterizing the gravitational potential of the galaxy is given by Miyamoto and Nagai (1975),

$$\Phi(R, z) = \frac{-GM}{\sqrt{R^2 + [a + (z^2 + b^2)^{1/2}]^2}} \tag{2.14}$$

This is called the Miyamoto-Nagai Potential. It is an axisymmetric potential where R is the cylindrical radial coordinate that measures distance from the galactic center and z is the vertical axis of the cylindrical coordinate that measures distance above or below the galactic plane. The constant a is the scale length that affects radial distribution, that is, a larger value of a corresponds to a disk that is more spread out and b is the scale height that affects the thickness of the disk, that is, a larger value of b corresponds to a thicker disk. So, to characterize the entire potential of the Milky Way, the potential from the Galactic disc Φ_d, bulge Φ_b, and halo Φ_h is added together $\Phi = \Phi_d + \Phi_g + \Phi_h$, where,

$$\Phi_d = \frac{-GM}{\sqrt{R^2 + [a + (z^2 + b^2)^{1/2}]^2}} \tag{2.15}$$

$$\Phi_{g,h} = \frac{-GM}{\sqrt{R^2 + z^2 + b^2}} \tag{2.16}$$

and the values for the mass M, constants a and b are taken from Bailer-Jones (2015) given in table 1. There are some other models for potential

which are much more accurate, MWPotential and McMillan Model are among them. MWPotential was created using data from the Sloan Digital Sky Survey (SDSS) and it consists of a double-exponential disk, a Hernquist bulge, and NFW (Navrro-Frenk-White) halo. The Model from McMillan (2016) is made with observational data from Gaia DR1 and DR2, and is considered very accurate as it fits well with the observed rotation curve of the Milky Way. For our application, the encounter parameters are not very sensitive towards the potential as the sample we consider for this study is within 1 kpc of the Sun and the encounter distance is much smaller than the scale lengths of the potential. Figure 5 shows the rotation curve of the potential used in our study, and the individual components.

Component	**M/M$_\odot$**	**a/pc**	**b/pc**
Disk	7.91×10^{10}	3500	250
Bulge	1.40×10^{10}	–	350
Halo	6.98×10^{11}	–	24000

Table 1: Parameter values used for Milky Way gravitational potential

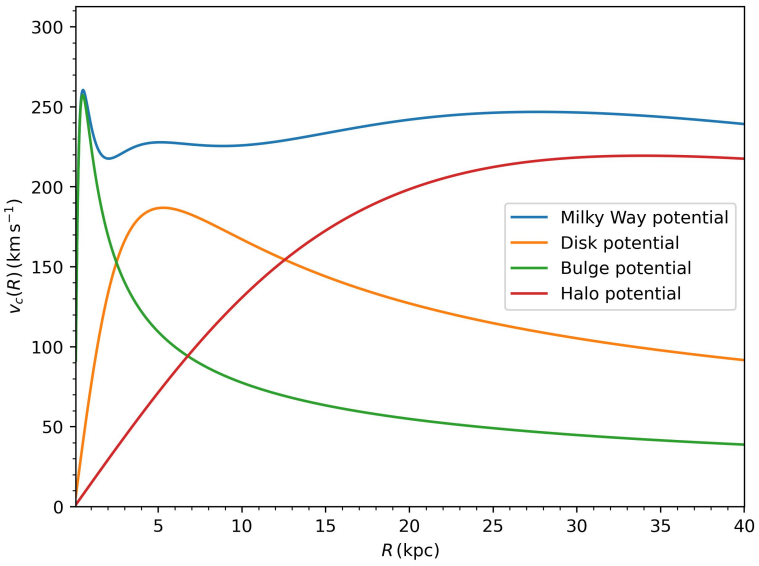


Figure 5: The rotation curve produced by equation 2.14 with individual components. The Milky Way potential is the sum of potential from Disk, Bulge, and Halo.

2.2.2 Orbital Integration

Once the right galactic potential (eq. 2.14) and numerical method (eq. 2.9) are chosen, then we should decide on the coordinate system to use for the orbital integration. The initial phase space coordinates $\mathbf{D} = (\alpha, \delta, \omega, \mu_{\alpha^*}, \mu_\delta, v_r)$ for the stars are obtained from Gaia DR3 in ICRS coordinate system where the origin is the Barycenter of the Solar system. These astrometric coordinates $\mathbf{D}$ are converted to Galactic phase space coordinates $(r, \phi, z, \dot{r}, \dot{\phi}, \dot{z})$ using `Astropy` package with phase space coordinates of the Sun with respect to the galactic center chosen as $r_\odot = 8$ kpc and $z_\odot = +10$ pc. ϕ is arbitrary since it's an axisymmetric potential. For the velocity, the solar motion is defined in the right-handed UVW coordinate system as shown in figure 6. For U, V, and W the solar velocity is 11.1 km/s, 12.24 km/s, and 7.25 km/s with uncertainties of 1 km/s chosen relative to local standard of rest (LSR) from Schönrich et al. (2010).

Once the coordinates are converted to a Galactocentric frame, the orbital integration is performed. The numerical integration is performed for 1000 time steps for both the Sun and the star. The technique in which the time range of integration is chosen is explained in the section 2.3.1. The time step with minimum separation between the star and the Sun is taken as the first perihelion. But, it may not be the actual perihelion of the star due to the discretization noise of numerical integration. So, to overcome this, linear motion approximation is applied from this point to get the final encounter parameters.

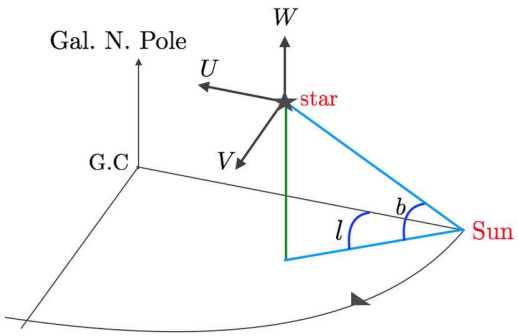


Figure 6: The diagram for galactic UVW coordinate system. G.C is the galactic center, and l, b are the galactic longitude and galactic latitude. Credits:Ramirez-Preciado et al. (2018)

2.3 Stellar encounter determination

2.3.1 Perihelion determination using integration

The method used for determining the encounter parameters is the technique used in Bailer-Jones (2015) and Bailer-Jones (2022). The Gaia DR3 contains 33,653,049 sources with complete phase space coordinates. The sources are given their phase space coordinates $\mathbf{D} = (\alpha, \delta, \omega, \mu_{\alpha^*}, \mu_\delta, v_r)$, uncertainties and correlation coefficients. The uncertainties in measurements are denoted by σ with the corresponding phase space coordinate symbol as subscript and correlation coefficients are denoted by $\rho(x_i, x_j)$ with the corresponding phase space coordinate x_i and x_j inside brackets. From this, the covariance matrix $\mathbf{\Sigma}$ is constructed for every source. Then, each phase space coordinate in $\mathbf{D}$ is resampled N times to obtain a 6D Gaussian distribution for each coordinate. As shown in figure 7, there will be N surrogate stars distributed around the actual star each having its phase space coordinates sampled using the covariance matrix and uncertainties. Each phase coordinate will have a distribution given by equation 2.17 where $\boldsymbol{\mu}$ is the mean assumed to be the measured phase space coordinate values from Gaia DR3 and $\mathbf{x}$ is the test point.

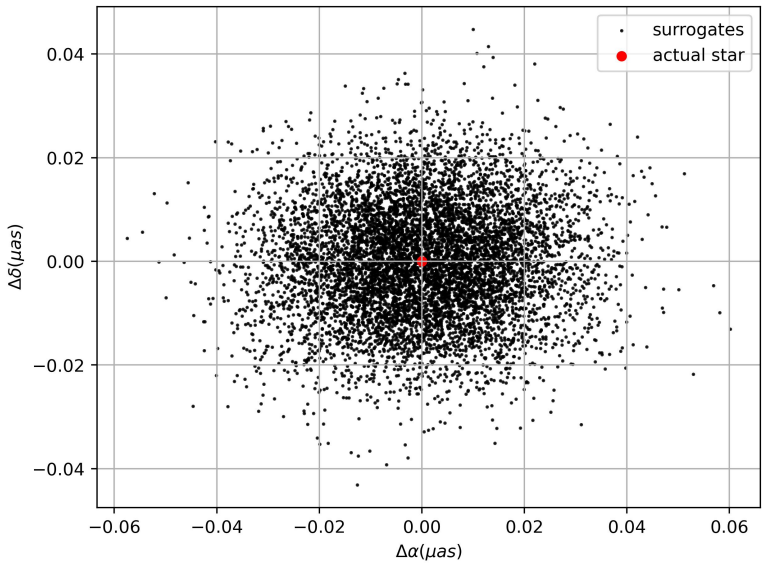


Figure 7: The actual star (red) with its phase space coordinate in the center, and its surrogate stars (black) derived from covariance matrix and uncertainty data. $\Delta\alpha$ and $\Delta\delta$ are differences in RA and Dec of surrogates from the actual star (red) given in microarcseconds.

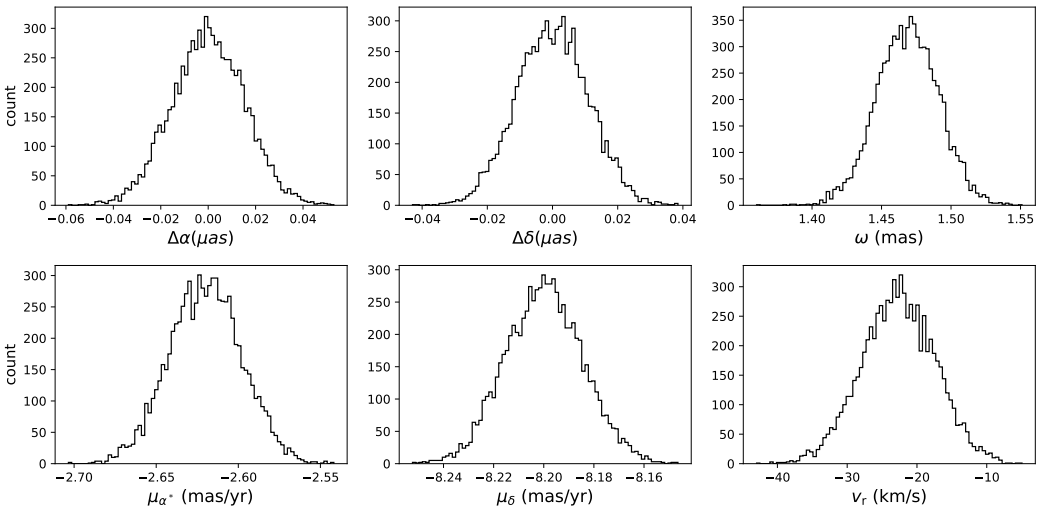


Figure 8: The distribution of phase space coordinates of surrogates of the star at t=0.

Then, the phase space coordinates of the actual star will have an initial distribution as shown in figure 8.

$$p(\mathbf{x} : \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{\exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right)}{\sqrt{(2\pi)^6|\boldsymbol{\Sigma}|}} \tag{2.17}$$

Before doing a full orbital integration, the close encounter parameters of each star and its surrogates are calculated through Linear Motion Approximation (LMA). In this method, the star is assumed to have zero acceleration. This technique works because stars are moving through weak gravitational fields, so the velocity of the stars only goes through a small change. Suppose the star is at a relative position $\mathbf{r}$ and moves with velocity $\mathbf{v}$ with respect to the Sun at time t. Let the initial position ($t = 0$) be at $\mathbf{r}_0$. Assuming no gravity or zero acceleration, the position at any time t is given by,

$$\mathbf{r} = \mathbf{r}_0 + \mathbf{v}t. \tag{2.18}$$

At the time of the close encounter t_{ph}^{LMA}, the position vector $\mathbf{r}$ and velocity $\mathbf{v}$ becomes perpendicular. So, $\mathbf{r}.\mathbf{v} = 0$. Substituting equation 2.18 to $\mathbf{r}.\mathbf{v}$ gives,

$$\mathbf{r}_0.\mathbf{v} + \mathbf{v}.\mathbf{v}t_{ph}^{LMA} = 0 \tag{2.19}$$

which gives the perihelion time using LMA,

$$t_{ph}^{LMA} = -\frac{\mathbf{r}_0.\mathbf{v}}{\mathbf{v}.\mathbf{v}}. \tag{2.20}$$

Substitution t_{ph}^{LMA} to equation 2.18 gives the perihelion distance d_{ph}^{LMA}. This can be written in terms of astrometric data measured from Gaia, where the initial position $\mathbf{r}_0$ is $1/\omega$, radial velocity v_r, and transverse velocity v_T. Then,

$$t_{ph}^{LMA} = -c_1 \frac{1}{\omega} \frac{v_r}{v^2} \quad \text{in years} \tag{2.21}$$

$$d_{ph}^{LMA} = 10^3 \frac{1}{\omega} \frac{v_T}{v}. \quad \text{in parsec.} \tag{2.22}$$

The v_T and v are given by,

$$v_T = c_2 \frac{\sqrt{\mu_{\alpha^*}^2 + \mu_\delta^2}}{\varpi} \qquad \text{is the transverse velocity in km/s,}$$
$$v = \left(v_T^2 + v_r^2\right)^{1/2} \qquad \text{is the total velocity in km/s,}$$

where $c_1 = 0.97779 \times 10^9$ in units of parsecs per kilometer per year and $c_2 = 4.74047$ in units of AU per kilometer per year, which are the conversion factors. The velocity v_{ph}^{LMA}, d_{ph}^{LMA}, and t_{ph}^{LMA} are the encounter parameters calculated for every source neglecting the presence of gravity.
Here, I will briefly describe the method used in Bailer-Jones (2022) to identify close encounters. Let's consider the phase space coordinate of one source from Gaia DR3.

(1) The phase space coordinates $\mathbf{D}$ are resampled 50 times creating 50 surrogates for the source.

(2) Each of the surrogate stars is subjected to LMA and if any one of the 50 surrogates approaches within 7.07 pc of the Sun, all of the 50 surrogates of that particular source will be numerically integrated through the potential mentioned in equation 2.15. The integration is performed for 50 uniform time steps over an interval 0 to $2t_{ph}^{LMA}$. The t_{ph}^{LMA} is the close encounter time in LMA of the particular surrogate star that we are integrating.

(3) If any of the surrogate stars in the previous step reached within the 7.07 pc of the Sun, then that Gaia source is again resampled to produce 1000 surrogate stars, and they are integrated with 500 uniform time steps.

(4) The median value of close encounter parameters of all surrogates of a single star is taken to obtain the final encounter parameters t_{ph}, v_{ph} and d_{ph} of the star.

It is important to note that surrogates may pass through either side of the Sun, and if we then calculate the average value of signed cartesian coordinates of these surrogates at close encounters, they cancel out positive and negative coordinates, then the calculated close encounter distance will be much smaller than the actual one. Therefore, always calculate the close encounter distance of each surrogate d_{ph}^{surr}, and then take the median value to obtain the final d_{ph} The calculation of close encounter parameters for a single star with 1000 surrogates took about 33 seconds on a 1.60 GHz x86_64 single-core CPU. For 33 million stars, it would take about years if not parallelised. For future Gaia releases, the number of stars with radial velocities will increase and it will require more computational resources to calculate encounters from this.

2.3.2 Machine learning methods

The previous section described the process of estimating close encounter parameters t_{ph}, v_{ph} and d_{ph}. It gave an idea regarding the time it would take to estimate these parameters for a large number of stars. So, instead of applying numerical integration, the problem can be formulated in terms of finding a relationship between the initial position, the encounter position, and the close encounter time of the star. Since there are a large number of sources in Gaia DR3, it is possible to train a machine-learning model to find this relationship. This is a regression problem of mapping input $\mathbf{D}$ to outputs t_{ph}, d_{ph} and v_{ph}. From the most simple case of LMA described in section 2.3.1, it is understood the problem is non-linear. In addition, each of the encounter parameters is determined by the combination input phase space coordinates. So, we aim for a method in which it takes six input phase space coordinates and returns three encounter parameters with each of these encounter parameters uniquely defined by the combination of these input coordinates. There are various machine learning algorithms available for doing non-linear regression. Since, this

problem when orbital integration is taken into consideration for finding close encounter parameters, the functions which map this relationship between input phase space coordinates and close encounter parameters become even more complex. In addition, the model should also take the measurement uncertainties from Gaia and the correlation matrix of the phase space coordinates and predict the uncertainties (calculated from surrogates) and correlations of the estimated encounter parameters. A complex relationship of this nature can be mapped using the Feedforward Neural Network (FNN).

A neural network can be considered as a universal function approximation Hornik et al. (1989). Suppose we have inputs $\mathbf{x}$ and outputs $\mathbf{y}$. The mapping between $\mathbf{x}$ and $\mathbf{y}$ is given by a complex function $f(\mathbf{x})$. A neural network can approximate this function $f(\mathbf{x})$ given it has enough hidden layers. This feature of neural networks helps us to solve our problem of mapping the initial input phase space coordinates of a star along with its uncertainties and correlations to the output close encounter parameters, uncertainties, and the correlations between them, figure 9.

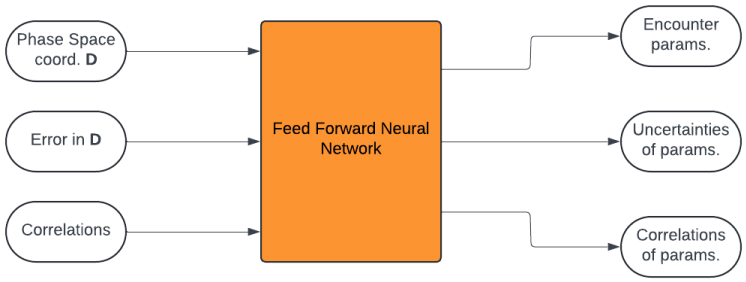


Figure 9: The block diagram shows the inputs and outputs of the neural network for finding the encounter parameters (t_{ph}, v_{ph}, d_{ph}) their uncertainties, and correlations.

The basic design of a neural network consists of one input layer, a hidden layer, and an output layer. The input layer takes the input features and passes this through hidden layers and finally, the predictions come through the output layer. In our case, these inputs will be the phase space coordinates along with their measurement error and correlations, and the output will be the close encounter parameters, their uncertainties, and correlations as shown in figure 9. To get an idea regarding the working, let's consider a simple network with 4 inputs, a hidden layer with one neuron, and an output layer with one output node as shown in figure 10. This is called a Perceptron. Each input x_i will be

assigned a weight w_i and this weight is multiplied by the corresponding input and added together. That is, if $\mathbf{x} = \{x_1, x_2, ..., x_n\}$ and $\mathbf{w} = \{w_1, w_2, ..., w_n\}$, then a dot product between vector $\mathbf{w}$ and $\mathbf{x}$ calculated, and a bias b is added to this to move the activation function either left or right. Let $z = \mathbf{wx} + b$, this value z is then passed through an activation function $\phi(z)$, which helps produce the nonlinearity in the output of the neural network. Examples of some activation functions are given in figure 11. The value from $\phi(z)$ will be the final output from the neural network. This series of steps is called one forward pass in a neural network.

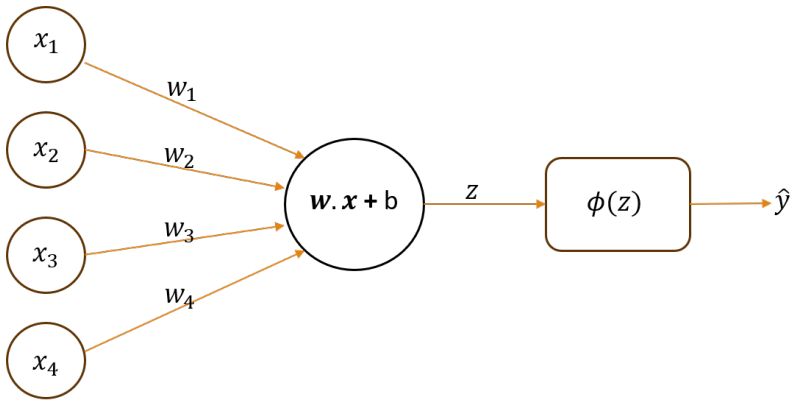


Figure 10: The figure represents a perceptron or a single neuron that takes input vector $\mathbf{x} = \{x_1, x_2, x_3, x_4\}$ each with its weights represented by vector $\mathbf{w} = \{w_1, w_2, w_3, w_4\}$. The weighted sum $\mathbf{w}.\mathbf{x}$ is combined with a bias b which results in value z and this is passed through an activation function ϕ to produce the output $\hat{y}$.

The output $\hat{y}$ that we get from the first forward pass is a result of weights $\mathbf{w}$ and bias b that is randomly initialised. For the network to learn from the training data, the $\hat{y}$ is compared with the true y using a loss function such as mean absolute error (MAE) or mean squared error (MSE). This error is then propagated backwards from output to input layer and this is called backpropagation. As an example, let's choose MAE given by $|y_i - \hat{y}|$ as a loss function. The loss functions show, how far is the predicted output from the actual value. This loss is then calculated for the entire training set to give cost function J given in equation 2.23

$$J = \frac{1}{n}\sum_{i=1}^{n} |y_i - \hat{y}|. \tag{2.23}$$

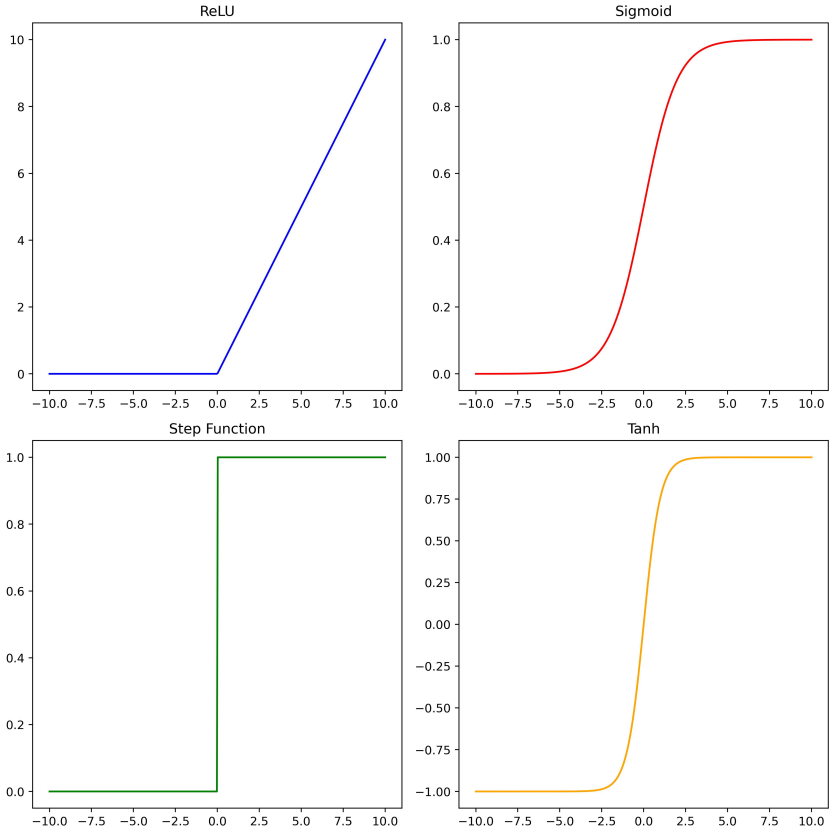


Figure 11: plots visualise the common activation functions used in neural networks: ReLU, Sigmoid, Step Function, Tanh.

The gradient of the cost function with respect to weight and bias is calculated using the chain rule for each layer. In this case, the chain rule is,

$$\frac{\partial J}{\partial \mathbf{w}} = \frac{\partial J}{\partial \hat{y}} . \frac{\partial \hat{y}}{\partial z} . \frac{\partial z}{\partial \mathbf{w}}. \tag{2.24}$$

Once the gradient is computed, the weights and biases are updated in such a way that the cost function reaches the minimum value. This is done using various optimisation algorithms like Stochastic Gradient Descent (SGD) Ruder (2016). The SGD updates weights and biases according to the equations 2.25

$$\begin{aligned} \mathbf{w} &= \mathbf{w} - \alpha \frac{\partial J}{\partial \mathbf{w}} \\ b &= b - \alpha \frac{\partial J}{\partial b} \end{aligned} \tag{2.25}$$

Here α is the learning rate. It decides the step size in which weights and biases

should be adjusted after each backpropagation. The learning rate should not be much larger or smaller. If the learning rate is much larger, it may miss the minima, and if it is much smaller it may stay in the local minima. Therefore an optimum size should be chosen for learning rate. This updation is continued until the cost function converges and reaches a minimum value. Then, the weights and biases that produced the minimum value for the cost function will be the final model that is used to make predictions.

Perceptron is the most simple neural network. But for practical applications, most neural networks contain more than one layer and nodes. As we will see in the coming sections, the final neural network used for predicting close encounter parameters contains three layers each with 300 nodes. The training of this neural network is a time-consuming process, as it involves a lot of fine-tuning to obtain the minimum value for the cost function. But, once a trained neural network is created that performs well on test data, then it could be used for making accurate predictions for any new sample. It is also possible to retrain this neural network for new incoming data from Gaia to improve its performance. In the coming sections, I will be describing the methods regarding how the training data is created and the steps in how the architecture for the neural network was chosen so that it can be used for predicting the close encounter parameters.

2.4 Neural Network for encounter determination

2.4.1 Training data creation

The model requires Gaia phase space coordinates and the correlation data, as well as the error associated with each phase space coordinate. GDR3 contains 1.46 billion sources with a limiting magnitude of about $G \approx 21$ and a bright limit of about $G \approx 3$. The full astrometric solution has been done for 585 million sources Lindegren et al. (2021). In our case, we only take stars in the 1 kpc range. This is because the average approaching velocities of all the stars is around 21 km/s, taking this into account, mostly only stars from less than 1 kpc range will reach close to our solar system before 5-20 million years. Calculating encounters for larger than these time scales is not practical because the galactic potential will have significant change from the present state in 20 million years. So, the study has been restricted to stars in the 1 kpc range. The data was extracted for all sources with parallax over error values greater than 5 having a full astrometric solution using the ADQL query.

```
SELECT source_id, ra, dec, parallax, pmra, pmdec,
    radial_velocity, ra_error, dec_error, parallax_error,
    pmra_error, pmdec_error, radial_velocity_error, ra_dec_corr,
    ra_parallax_corr, ra_pmra_corr, ra_pmdec_corr,
    dec_parallax_corr, dec_pmra_corr, dec_pmdec_corr,
    parallax_pmra_corr, parallax_pmdec_corr, pmra_pmdec_corr,
    parallax_over_error FROM gaiadr3.gaia_source WHERE parallax
    > 1 AND parallax_over_error > 5 AND radial_velocity IS NOT
    NULL;
```

This contained 12,044,415 sources. From this parallax zero-point correction was done, which resulted in 12,043,872 sources. The missing 543 sources are the rare cases for which phot_g_mean_mag is not available. So, these sources are ignored.

Once the dataset is obtained, the encounter parameters of these stars were found using orbital integration as mentioned in section 2.3.1. But, due to technical limitations, orbital integration could only be completed for 4,919,099 sources. So, these 4,919,099 sources were used for making the train-test data. This dataset does have a small bias towards some areas of the sky, but from a statistical point of view, this does not matter for this study.

As a first step in developing the right architecture for the network, training

was done for stars in the 100 pc range. This contained 169,741 stars with parallax_over_error > 5. From this 127,306 sources were picked at random for the training set and the remaining 42,435 sources were used for testing. Different designs were experimented for the problem as explained in section 2.4.2 to decide the structure. Then, training was done for a neural network from one to three layers of 300 nodes with various learning rates. From the loss curve of all these models, one with better convergence was chosen, which we later trained on stars from the 1 kpc range. Figure 12 shows the distribution of input coordinates as well as the output encounter parameters of the dataset used for training the final model for the 1 kpc range.

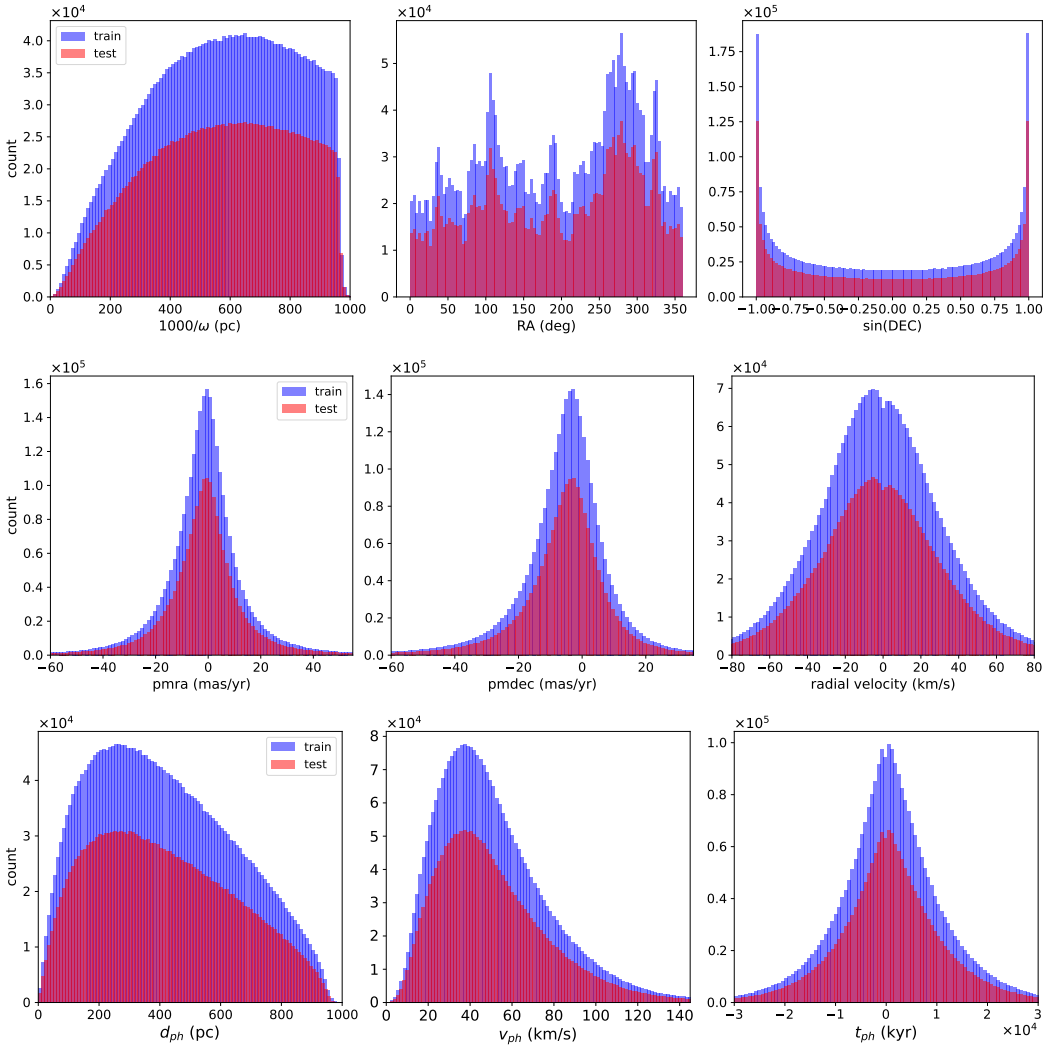


Figure 12: Distribution of training and testing data for input phase space coordinates (first two rows) and the output close encounter parameters (bottom row).

The train-test split for this model was in a ratio of 66:34, which means 66% of the data were used for training and 34% of the data were used for testing. This split ratio was chosen after going through a random search for various other split ratios. There were 2,951,517 samples in the training set and 1,967,582 samples in the testing set. From the training set, 590,303 random samples were used as a validation data set. Validation data is used to monitor the performance of the network in unseen data while training.

Since our model should be able to create a distribution of encounter parameters d_{ph}, v_{ph}, t_{ph}. The input parameters for our network are ra, dec, parallax, pmra, pmdec, radial_velocity, parallax_error, pmra_error, pmdec_error, radial_velocity_error, parallax_pmra_corr, parallax_pmdec_corr, pmra_pmdec_corr from the Gaia dataset. The input phase space coordinates ra, dec, parallax, pmra, pmdec, radial_velocity will be represented by $\mathbf{D} = (\alpha, \delta, \omega, \mu_{\alpha^*}, \mu_\delta, v_r)$. We ignored the uncertainties in rα and δ because they are much smaller in comparison to uncertainties in parallax. Similarly, the correlations of the α, δ, and v_r coordinates with other variables are also ignored. For α and δ, we neglected their uncertainties because they are small, therefore we may also ignore their correlations. For radial velocity, the measurement is taken from spectra which is a different method while other coordinates are measured using a similar method, therefore, we ignore the correlation of radial velocity with other coordinates.Katz et al. (2023).

For the outputs, we have tph_med, dph_med, vph_med, tph_std, dph_std, vph_std, tph_dph_corr, tph_vph_corr, dph_vph_corr. The encounter parameters tph_med, dph_med, vph_med will be represented by $\theta = (t_{ph}, d_{ph}, v_{ph})$. The training labels (close encounter parameters) are calculated using the method specified in section 2.3.1. Using the error and correlation data provided in Gaia DR3, a 6D covariance matrix is created, and using this, I resampled the input phase space coordinates of each source 200 times. This gives 200 input phase space coordinates for each source. Then, each of these sources along with these 200 surrogates (that represent the uncertainties) were orbitally integrated to find the close encounter parameters. Then, we obtain 200 values for each encounter parameter for each source. Then, a median of this distribution of encounter parameters is taken to get $\theta = (t_{ph}, d_{ph}, v_{ph})$ which are the median values of encounter parameters, and the standard deviation is taken for each encounter parameter to obtain the uncertainty in the estimation of encounter parameter.

For the outputs that are only in the positive range, for example, the standard deviations, close encounter distance, and encounter velocities the log is taken so that the values are not confined to the positive range. Then, inputs and outputs are scaled according to the equation 2.26 using the mean and standard deviation of each feature.

$$\hat{x_i} = \frac{x_i - \mu}{\sigma} \tag{2.26}$$

Here x_i is the input or output feature and μ, σ are the mean and standard deviation of each feature. The correlations tph_dph_corr, tph_vph_corr, dph_vph_corr are not scaled because they always remain between -1 and 1. Other physical parameters and their uncertainties have physical units of a different range, hence demanding the scaling for smooth convergence of the neural network. The model predicts the encounter parameters with the respective uncertainties in scaled units. Then, these outputs are unscaled and antilog is taken for the ones that are only in the positive range (d_{ph}, v_{ph}, and all the uncertainties) to obtain the required values.

2.4.2 Network architecture

To develop an initial architecture, a simple dataset is taken from Gaia DR3 which consists of all the sources from the 100 pc range with parallax_over_error greater than 5 and has a radial velocity. This consisted of 169,710 sources with radial velocity after zero point correction (zero point correction is not done for sources with no G-band mean magnitude available, so they are removed from the dataset). Since most of the sources are from the 100 pc range, they mostly follow a linear path towards the encounter position. This dataset is then randomly split into 75% (127,282 sources) for training and 25% (42,428 sources) for testing. The same sources are also used for creating another dataset that has close encounter parameters obtained through linear motion approximation (LMA) as in section 2.3.1. In all these cases, only the input phase space coordinates $\mathbf{D} = (\alpha, \delta, \omega, \mu_{\alpha^*}, \mu_\delta, v_r)$ and outputs encounter parameters such as close encounter distance d_{ph}, close encounter time t_{ph}, and close encounter velocity v_{ph} are used for training. These values for encounter parameters are derived without resampling for the error. The distribution of encounter parameters $d_{ph}^{LMA}, t_{ph}^{LMA}$ and v_{ph}^{LMA} which are the outputs for the Neural Network trained

for LMA dataset is given in figure 14.

Initially, a neural network was developed to predict encounter parameters, assuming their motion is linear. Predicting the position of a close encounter is less complex when the sources move in straight lines, providing a foundation for designing the network's architecture for actual orbit. Using training data based on Linear Motion Approximation (LMA), I trained a set of neural networks with a single layer, experimenting with different numbers of nodes (30, 60, 100, and 200) and learning rates of 0.01 using Adams optimizer Kingma and Ba (2017) for 1000 epochs with loss function Mean Absolute Error (MAE). Subsequently, this same configuration of node counts was applied to train models with two and three layers, across the same learning rate and loss function. After comparing the performance of all these designs on the test dataset, the network with three layers consisting of 100 nodes gave the least overall loss. Then, this architecture was taken as the base architecture and was further improved by adjusting the learning rate and batch size of the training to give the best performance. Finally, the network with design 6:100:100:100:3, where 6 is the number of input nodes, 100 is the number of nodes in each hidden layer with ReLU as activation function and 3 is the number of output nodes with a linear activation function. The network was trained with a learning rate of 0.0007 and batch size 256 was trained for 10,000 epochs. Since batch size is taken as 256, the training data will be randomly shuffled and divided into equal batches of size 256. Then during the training process, forward and backpropagation is done for each batch until the entire training data is seen by the neural network. This constitutes one epoch. During the training process, the weights are automatically saved for the epoch with the least validation loss. These parameters gave the best performance for LMA data with a scatter of 3.19 kyr for t_{ph}^{LMA}, 0.048 pc for d_{ph}^{LMA} and 0.042 km/s for v_{ph}^{LMA} where median absolute deviation (MAD) of true t_{ph}^{LMA} is 751.55 kyr, d_{ph}^{LMA} is 18.25 pc and v_{ph}^{LMA} is 16.36 km/s. The scatter is defined as $median|y^{pred} - y^{true}|$, where y^{pred} is the predicted value of quantity y^{true}, and MAD measures the dispersion of the test dataset by using eq. $median|x_i - \bar{x}|$ where x_i is the true value of the data point and $\bar{x}$ is the mean of true values of the dataset. Figure 13 shows the distribution of training and test data of the inputs of the Neural Network designed for LMA.

The same 100 pc dataset is then used with a 75:25 train-test split that is

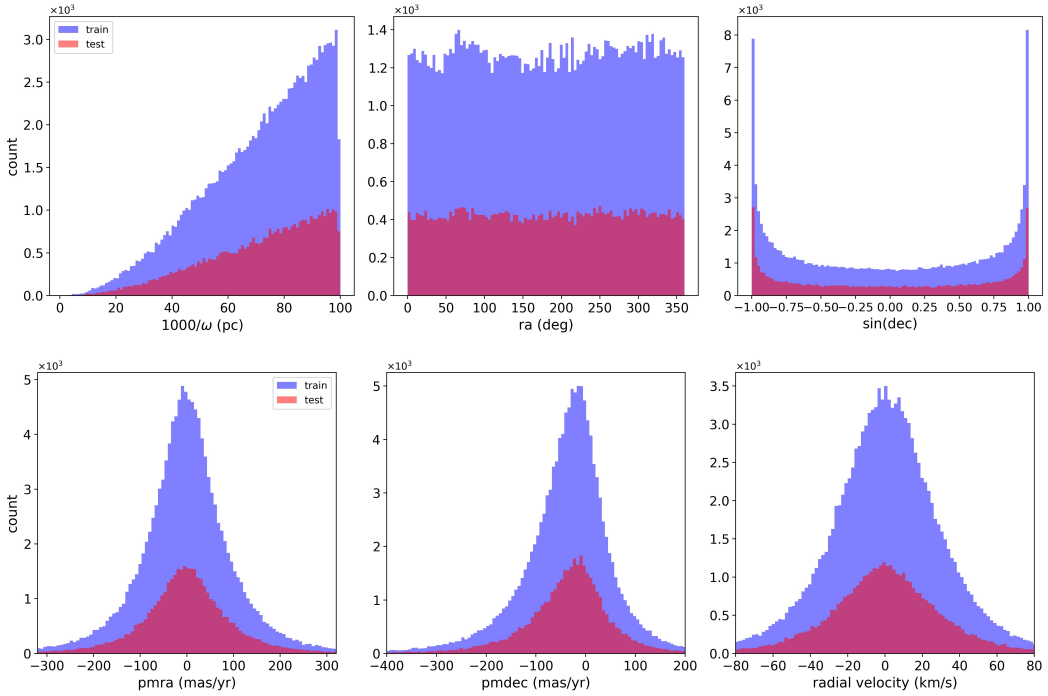


Figure 13: Distribution of train/test inputs of neural network designed for predicting encounter parameters obtained through Linear Motion Approximation (LMA) for sources in 100 pc radius of the Sun.

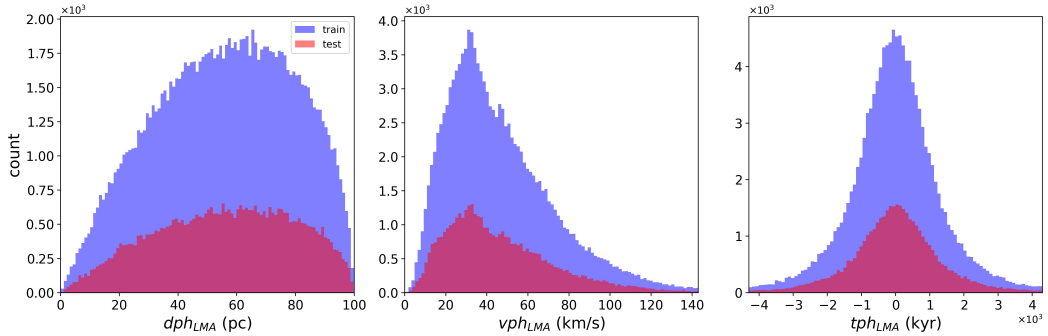


Figure 14: Distribution of train/test outputs of Neural Network designed for predicting encounter parameters obtained through Linear Motion Approximation (LMA) that is trained for sources in 100 pc radius of Sun.

with 127,282 sources in the training set and 42,428 sources in the test set is created with output as close encounter positions for curved orbits (numerically integrated) instead of linear orbits and then used for training another model. We start with the initial design used for linear orbit, that is 6:100:100:100:3 and then the number of nodes in hidden layers was adjusted till the loss reached the

minimum, for 1000 epochs. The final design that achieved the minimum mean absolute error (MAE) loss is a model with 3 hidden layers, each containing 180 nodes, resulting in a 6:180:180:180:3 architecture. ReLU was used as the activation function for the hidden layers. The training parameters for this model included a learning rate of 0.0007 with the Adam optimizer using its default parameters, a batch size of 256, and a training duration of 10,000 epochs. The distribution of inputs of the train-test set of this model is the same as in figure 13 used for training the LMA model. The distribution of outputs is different as it is the encounter parameters obtained through orbital integration. Figure 15 shows the distribution of encounter parameters for the train-test set obtained through orbital integration. The model achieved a scatter of 4.73 kyr

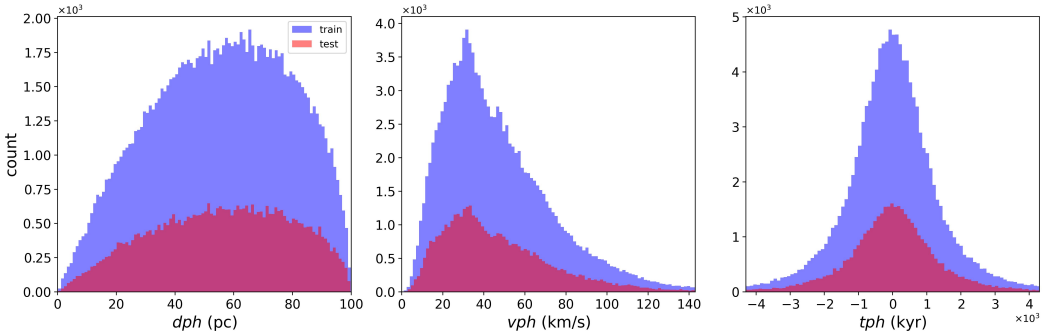


Figure 15: Distribution of train/test outputs that are obtained through orbital integration of sources within 100 pc, and used for training and evaluation of neural network designed for curved orbits.

for t_{ph}, 0.064 pc for d_{ph}, and 0.056 km/s for v_{ph}. The median absolute deviation (MAD) for test data of each encounter parameter t_{ph} is 760.49 kyr, d_{ph} is 18.22 pc and v_{ph} is 16.35 km/s. This is slightly higher than the one with linear motion approximation, which was expected because predicting curved orbit is more complicated compared to linear orbits. Figure 16 shows the absolute error, that is $|y_{pred} - y_{true}|$ of each encounter parameter obtained through orbital integration for sources within 100 pc of the Sun. Using scatter density plots, it is not possible to see the presence of any underlying weak patterns. However, it is interesting to see certain patterns that are present, when the bias $(y_{pred} - y_{true})$ of all stars is analysed in every 1 pc distance interval. Figure 17 gives the bias and absolute error of neural network prediction for encounter distance d_{ph} as a function of the current distance. Each point is

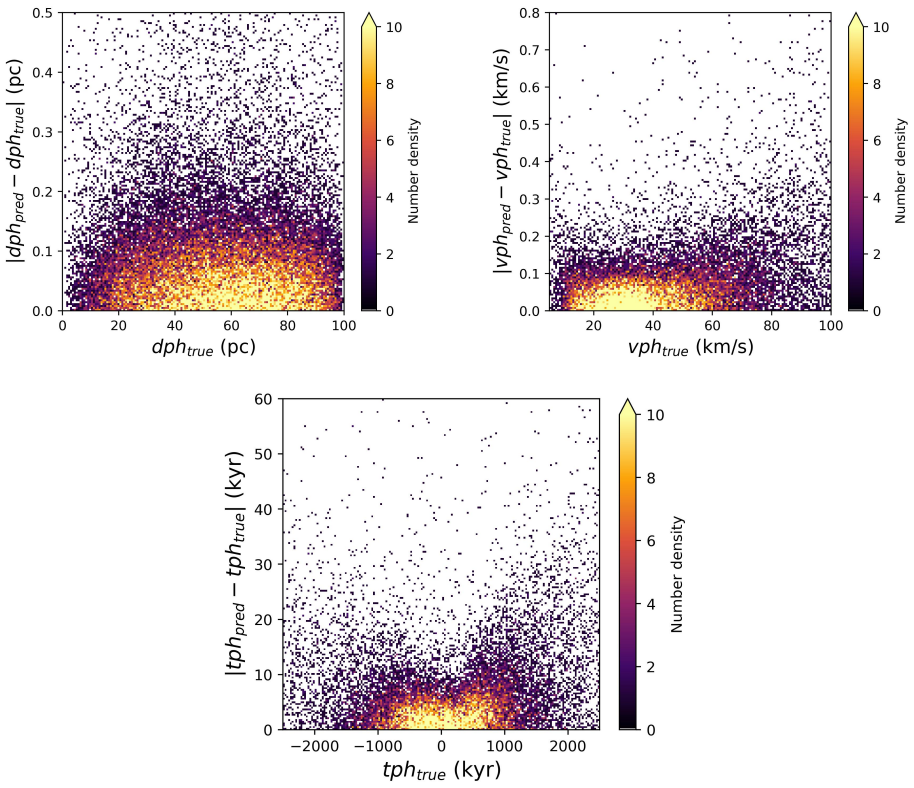


Figure 16: Scatter density plot shows the relationship between the true encounter parameter (d_{ph}, v_{ph}, t_{ph}) in the x-axis and the absolute prediction error in the y-axis for the neural network trained for encounter parameters obtained through orbital integration for sources within 100 pc of Sun. The color bar indicates the number density of the points.

the bias of stars in 1 pc bin size. It can be seen from the absolute error plot that stars close to the Sun and farther from the Sun have the least accuracy in prediction. This is because of the smaller number of sources in training data in the close distances as seen from the histogram of d_{ph} in figure 14, and for large distances also the number of sources drops which increases the absolute error. It is also interesting to see wiggles present in the bias vs. current distance plot, which shows some issues with the convergence of the neural network. The same wiggles are present also for the bias vs. current distance plot of the neural network trained for linear motion approximation. Figure 18 shows this comparison plot. Each point represents the median bias of all the sources in a 1 pc bin size interval of the current distance. The wiggles have a higher amplitude for the neural network trained in LMA data. The presence of these

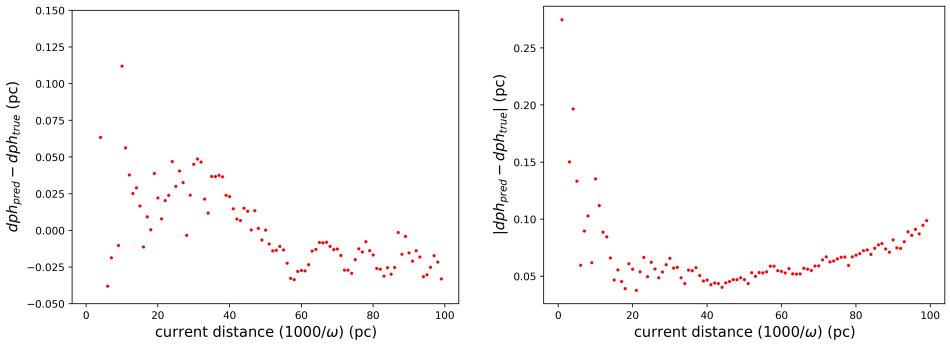


Figure 17: Left: The x-axis represents the current distance towards the source and the y-axis is the bias in encounter parameter prediction d_{ph}. Each point is the median of prediction bias in d_{ph} for all the sources present in 1 pc current distance bin size. It can be seen there are some wiggles present which shows some issues with the convergence of the neural network. Right: The absolute error is plotted against the current distance. Each point represents the median absolute error (scatter) of all sources within 1 pc current distance bin size.

wiggles will be explained in detail in section 3.1.2.

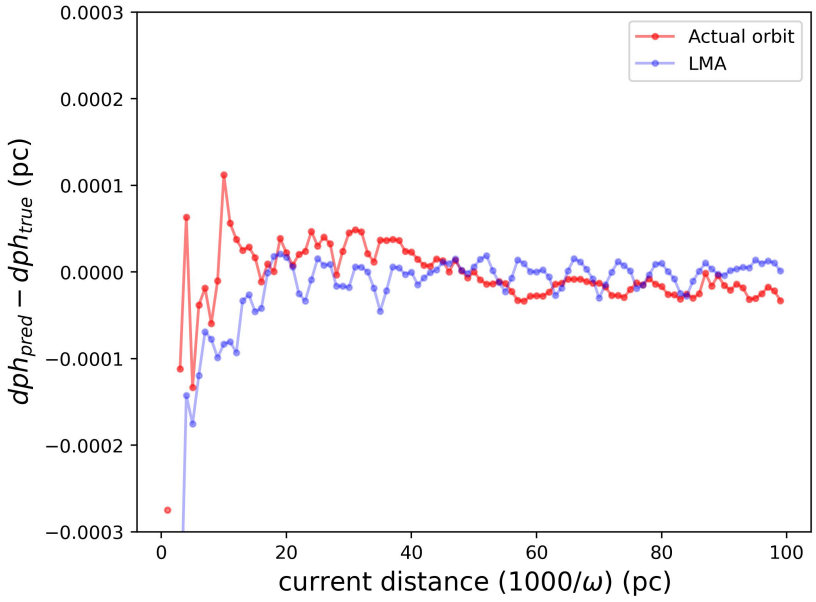


Figure 18: The plot compares the bias in predicted encounter distances against the current distance for a neural network trained on encounter parameters obtained through orbital integration (red) and linear motion approximation (blue). Both methods exhibit a similar trend, with higher negative bias at shorter distances that stabilizes at larger distances. The plot also shows the presence of wiggles in both cases, with the wiggles having a higher amplitude in the LMA method.

Experiments were also done with an approach that takes Gaia ICRS coordinates as input and provides the Galactocentric Cartesian coordinate differences relative to the Sun at the time of close encounter ($dx_{ph}, dy_{ph}, dz_{ph}, dv_x^{ph}, dv_y^{ph}, dv_z^{ph}$) along with t_{ph} as outputs. From these coordinates, the encounter parameters d_{ph} and v_{ph} are derived. This method resulted in a scatter of the encounter parameters, with values of 0.60 pc for d_{ph}, 0.38 km/s for v_{ph}, and 21 kyr's for t_{ph}, where MAD of the testing set is the same as that used for a testing model that is trained for curved orbit. In all these cases the input and outputs were re-scaled with mean and standard deviation using equation 2.26 before training.

The design of the final neural network used for estimating close encounter parameters with its uncertainties and correlations has a design of 13:300:300:300:9. The training and performance of this model will be explained in the next section.

3 Results

3.1 Neural Network Prediction

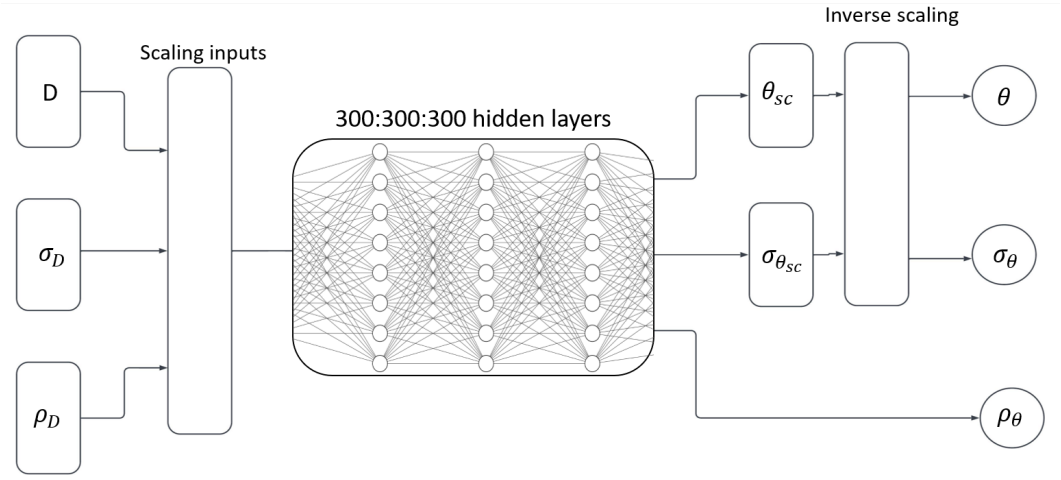


Figure 19: Final NN used for predicting encounter parameters θ. The **D** is the phase space coordinates, σ_D is the error in measurement of the phase space coordinates and ρ_D is the correlation matrix of the phase space coordinates. The encounter parameters t_{ph}, d_{ph} and v_{ph} are represented by θ. Scaled output is represented with subscript *sc*. The uncertainty in estimation of encounter parameters is σ_θ and correlation between the encounter parameters are represented by ρ_θ.

The final neural network design chosen for estimating encounter parameters (θ) with its uncertainty(σ_θ) and correlations(ρ_θ) is a design 13:300:300:300:9. A block diagram of the network is given in figure 19. The model consists three-layer fully connected dense network with a ReLU activation function. All output nodes except the ones predicting correlation have linear activation functions. The output nodes that predict correlations have tanh as an activation function, correlations are always between -1 and 1. The 13 inputs are phase space coordinates $D = (\alpha, \delta, \omega, \mu_\alpha, \mu_\delta, v_r)$, their measurement error $\sigma = (\sigma_\omega, \sigma_{\mu_\alpha}, \sigma_{\mu_\delta}, \sigma_{v_r})$ and the correlation matrix $\rho_D = (\rho(\omega, \mu_\alpha), \rho(\omega, \mu_\delta), \rho(\mu_\alpha, \mu_\delta))$. The outputs are encounter parameters $\theta = (t_{ph}, d_{ph}, v_{ph})$, their uncertainties estimated from surrogates $\sigma_\theta = (\sigma_{tph}, \sigma_{dph}, \sigma_{vph})$ and correlations between encounter parameters $\rho_\theta = (\rho(t_{ph}, d_{ph}), \rho(t_{ph}, v_{ph}), \rho(d_{ph}, v_{ph}))$. The output parameters which are positive d_{ph}, σ_{dph}, v_{ph}, σ_{vph} and σ_{tph} are converted to log scales so that it will remain as positive quantities from network output. Then, both inputs and outputs except correlations are scaled using equation 2.26.

Using these outputs from the network, a distribution of the encounter parameters d_{ph}, v_{ph}, t_{ph} of each star can be generated. The loss function used for this neural network is Mean Absolute Error (MAE) given by,

$$MAE = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{1}{9}\sum_{j=1}^{9}|y^{i}_{j,true} - y^{i}_{j,pred}|\right). \tag{3.1}$$

The training process aims to reduce this loss function. The loss function takes the y^{true} and y^{pred} for each output in scaled units and adds the loss of each output feature to give the final loss. The train-test split for this model was in a ratio of 66:34, which means 66% of the data were used for training and 34% of the data were used for testing. There were 2,951,517 samples in the training set and 1,967,582 samples in the testing set. The network was trained for 10,000 epochs, batch size 512, and learning rate 0.0001 with equation 3.1 as the loss function and Adams optimizer as the optimization algorithm. Default values are used for other parameters. The training was done in 6 CPUs and 4000 MB memory took 48 hours to complete 10,000 epochs.

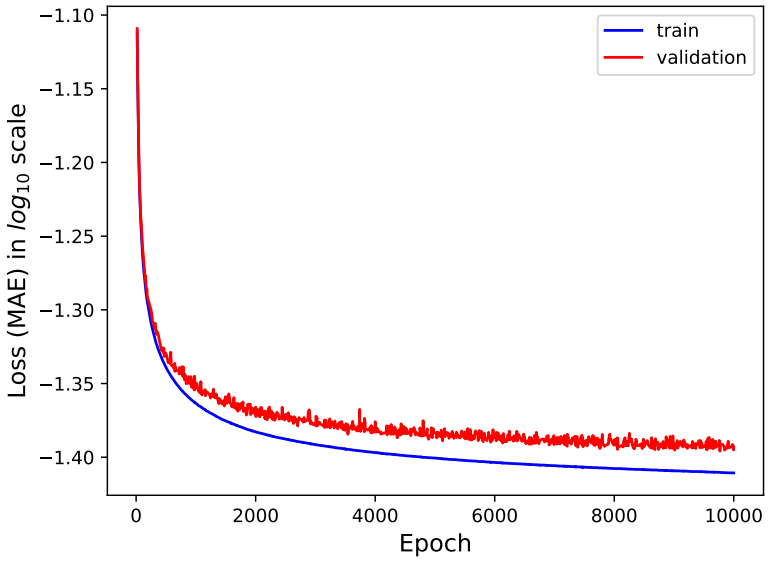


Figure 20: The overall loss curve of the model with the x-axis being the number of epochs and the y-axis being the mean absolute error (MAE) of the output in scaled units. The blue and red line is the corresponding training and validation curve.

The performance of the model was evaluated by two metrics bias and scatter.

The difference also called bias is defined as $median(y_{pred} - y_{true})$ and scatter is defined as $median|y_{pred} - y_{true}|$. The overall bias and scatter for each encounter parameter output is given in table. 2. Figure 20 shows the overall loss curve (MAE in eq.3.1) of the neural network in log_{10} scale.

3.1.1 Encounter distance d_{ph} and uncertainty σ_{dph}

The encounter distance d_{ph} and its uncertainty σ_{dph} as defined in section 2.3.1 is one of the main parameters of interest. An accurate estimate of encounter distance is one of the main goals of our neural network. The encounter parameter d_{ph} and uncertainty σ_{dph}, have a scatter of 1.7 pc and 0.52 pc respectively. The MAD of true values of d_{ph} and σ_{dph} for training data is 176.57 pc and 9.97 pc. Figure 21 shows the loss curve of d_{ph} and σ_{dph} in scaled units. The purpose of this loss curve is to show the convergence of the model. The training and validation loss, both converge and flatten without diverging much from each other ensuring that the model is not overfitting. The prediction against true values plot in figure 22 shows that the prediction of d_{ph} and σ_{dph} are strongly correlated.

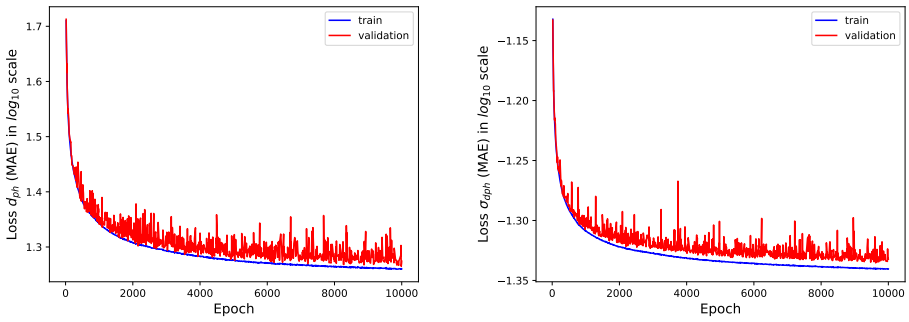


Figure 21: Left: loss of encounter distance in scaled units. the x-axis shows the epochs and the y-axis shows the loss of d_{ph} in scaled units. The purpose of this plot is to show the convergence of the network for d_{ph}. Right: loss of σ_{dph} scaled units.

Analysing the performance of the network concerning true encounter distance shows more details about how well the model functions for various distance ranges. Figure 23 shows how the d_{ph} bias varies concerning true encounter distance. From the plot, it can be seen that the median bias of d_{ph} diverges away from the zero axis as the close encounter distance increases. But, we are mostly interested in the close encounters in smaller distances. The relative bias

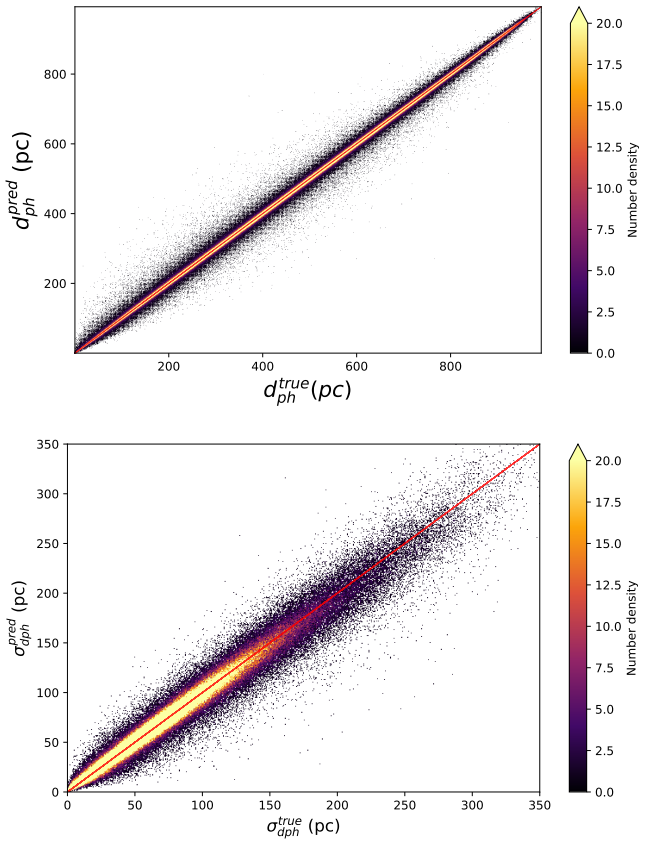


Figure 22: Top: d_{ph}^{pred} is plotted against dph_{ph}^{true} in parsec (pc), both prediction and true values correlate well since most of the points are concentrated in a diagonal red line. The overall bias for d_{ph} is -0.45 pc and the scatter is 1.7 pc, with MAD of the true value of d_{ph} being 176.5 pc. Bottom: σ_{dph}^{pred} is plotted against σ_{dph}^{true} in parsec (pc). The points are concentrated along the diagonal line showing a strong correlation between prediction and true values. The overall bias for σ_{dph} is -0.012 pc and the scatter is 0.52 pc with MAD of true values of σ_{dph} being 9.97 pc

gives a more detailed picture of how the error is changing relative to the true value. This helps us to understand in which range the encounter distance gives high error. Figure 24 shows this relative error against current distance and it can be seen that stars with smaller current distance have high relative error. This is counter to our goal of accurately predicting encounters with smaller encounter distances. The relative error in d_{ph} has a peak median relative bias value of -0.03 for stars in the 0 to 40 pc range.

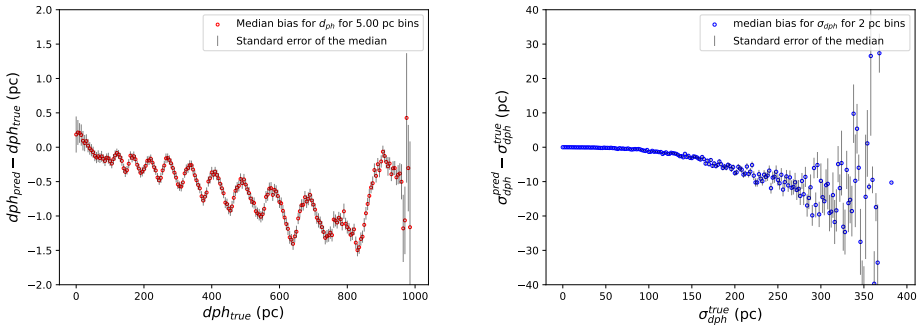


Figure 23: Left: $d_{ph}^{pred} - d_{ph}^{true}$ vs. the bias of d_{ph}^{true}. Each point denotes the median bias of all the stars in a 5 pc bin size. The wiggles present in the bias plot are explained in section 3.1.2. Right: Same plot with σ_{dph}. The spread increases towards a larger distance.

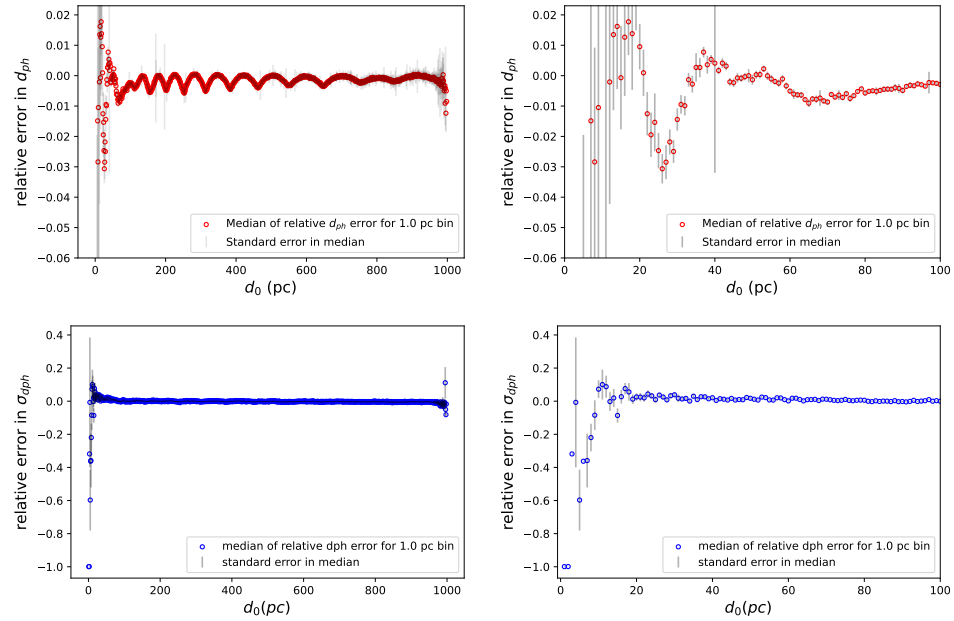


Figure 24: Top: current distance against relative dph error for all the test samples. It can be seen that samples closer to the Sun are having higher relative error in encounter distance. The top right plot is the zoom into 100 pc range with a 1 pc bin size. Bottom: Current distance against relative σ_{dph} error for all the test samples. The bottom right plot is the zoom of this sample in 100 pc.

The relative error in d_{ph} wiggles around the zero-axis after 100 pc before going to the negative side close to 1000 pc. The situation is the same for d_{ph} uncertainty σ_{dph}, the relative error goes off the zero axis in the extremes. One of the possible reasons for behavior is the comparatively lower number of stars in

training data at closer and larger distances as shown in the histogram of figure 15. Let's analyse this error based on quantity $d_0 - d_{ph}^{true}$ which is the difference between current distance d_0 and close encounter distance d_{ph}. The value of $d_0 - d_{ph}^{true}$ will be close to zero for stars that are already in close encounter position and they will have a radial velocity equal to zero and tangential velocity perpendicular to the radial distance at the time of close encounter as shown in figure 25. $d_0 - d_{ph}^{true}$ is larger for stars that traveled a large distance to the

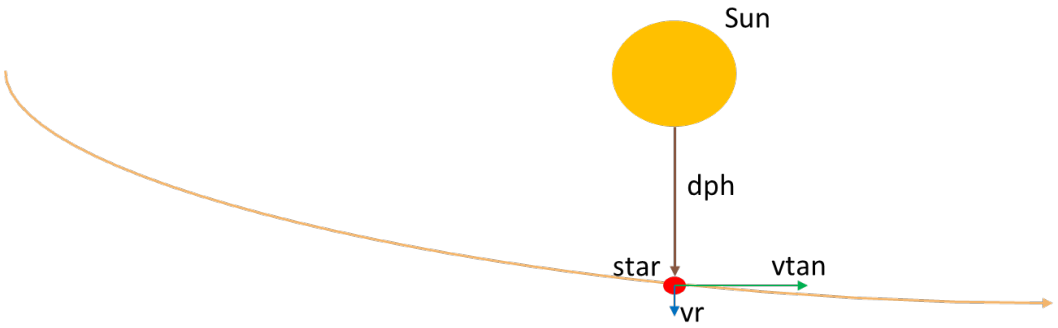


Figure 25: radial velocity of star approaches close to zero during the close encounter and tangential velocity reaches the peak value.

point of perihelion, and it is smaller for stars that are already close to the point of perihelion. The stars that travel large distances to reach the point of perihelion have a low signal-to-noise ratio (SNR), therefore, the network finds it difficult to predict the encounters for these particular stars. Similarly, some of the stars that are already in close encounters have larger current distances, hence, smaller SNR, and that makes it difficult to predict encounters. From the figure 26, it can be understood that sources with low SNR in parallax have large MAE, and they are also far. Sources closer to the Sun also exhibit higher MAE despite having a high SNR, as they are underrepresented in the training data, comprising only about 0.23% of the total sources in training data for distances less than 50 pc. So the network has high prediction error on sources that are close to the Sun.

Similarly, when the performance is analysed as a function of curvature defined as $|d_{ph}^{LMA} - d_{ph}^{true}|$ and distance traveled towards the perihelion from the current distance defined as $d_0 - d_{ph}^{true}$. It can be observed that the parallax SNR is low for sources that travel larger distances to perihelion and these sources also has high curvature in orbit. These two factors contribute together to the high

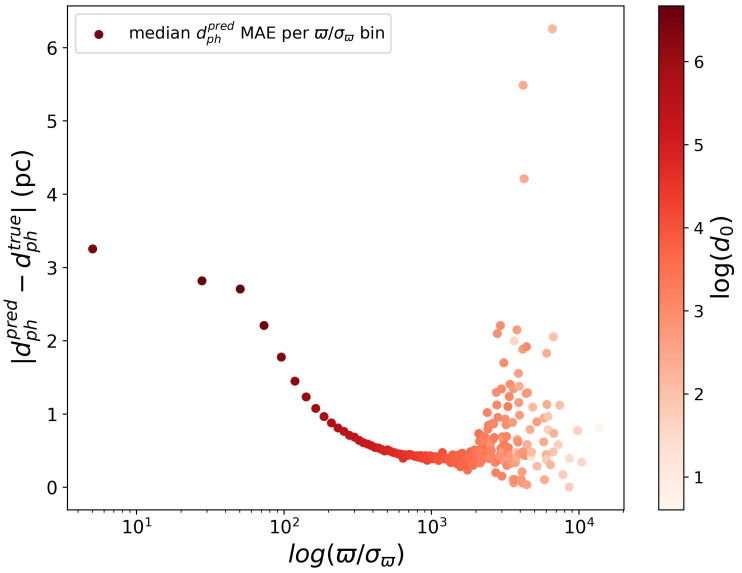


Figure 26: The $log(\varpi/\sigma_\varpi)$ is plotted against $med|d_{ph}^{pred} - d_{ph}^{true}|$ per ϖ/σ_ϖ bin with colorbar representing a log of current distance d_0 to the source $log(d_0)$. The MAE is high for lower SNR in parallax where the distance to the sources is also high. Sources far away have lower SNR in parallax, making it difficult for the model to make accurate predictions. Also, the much closer sources have high MAE even though they have a high SNR. This is because in smaller distances (less than 50 pc) only 0.23% of total training sources are present

scatter in prediction for these sources, refer to figure 27.

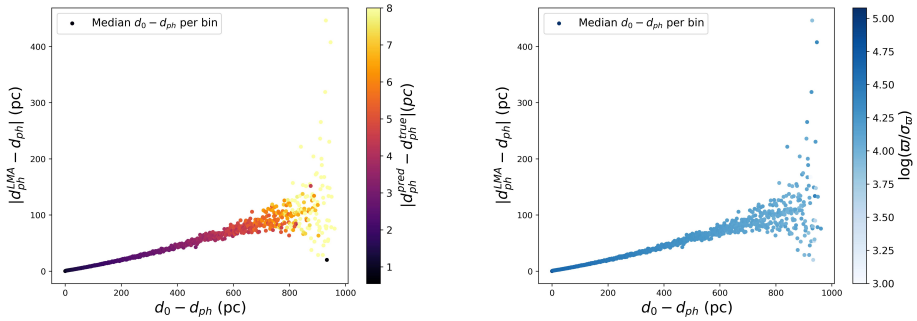


Figure 27: Left: x-axis represents the distance traveled by the source and y-axis represents the curvature of the orbit with colorbar giving the scatter of d_{ph} prediction. Right: The same plot with the x-axis represents the distance traveled by the source and the y-axis represents the curvature of the orbit, but colorbar shows the log of parallax SNR.

The uncertainty in encounter parameter σ_{dph} do not have peaks like d_{ph}, the

$|\sigma^{dph}_{pred} - \sigma^{dph}_{true}|$ increases as the current d_0 increases figure 28.

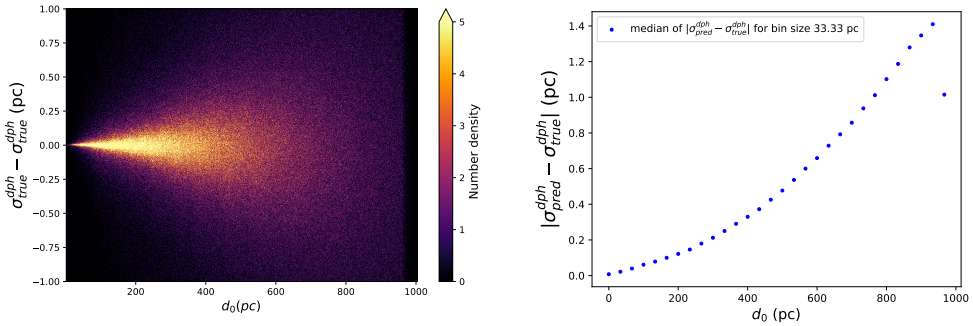


Figure 28: Left: σ^{dph} prediction bias (y-axis) against current distance d_0 (x-axis) shows that performance decreases with increasing current distance. Right:σ^{dph} scatter (y-axis) against current distance d_0 (x-axis).

3.1.2 The wiggles in d_{ph} performance plot

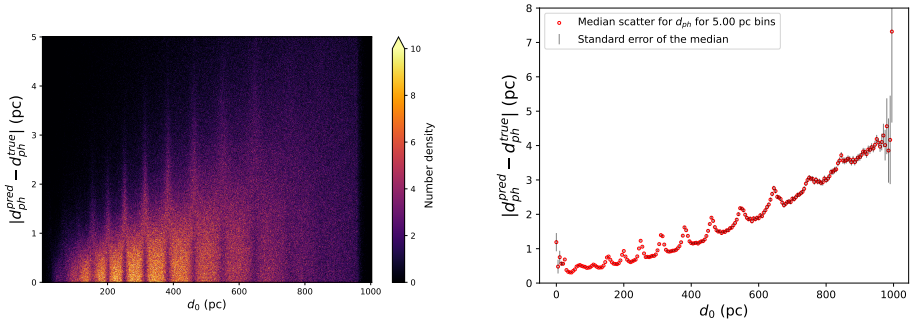


Figure 29: Left: density plot with current distance d_0 in x-axis and y-axis showing the MAE of dph prediction. Right: The same plot with each point representing a median of $|dph_{true} - dph_{pred}|$ in shows the wiggles.

The density plot of scatter of close encounter distance d_{ph} against current distance d_0 shows wiggles at certain intervals of d_0 figure 29 and these wiggles are also visible in figure 23. To see if the neural network is producing any kind of patterns seen in the training data, the wiggles were also examined in the input features such as pmra and radial velocity, as shown in figure 30. The bin size is taken in such a way, that it is possible to see the wiggles in the d_0 vs. pmra and radial velocity plot. To see if these patterns are caused because of RA/DEC selection bias, which occurred because orbital integration could only

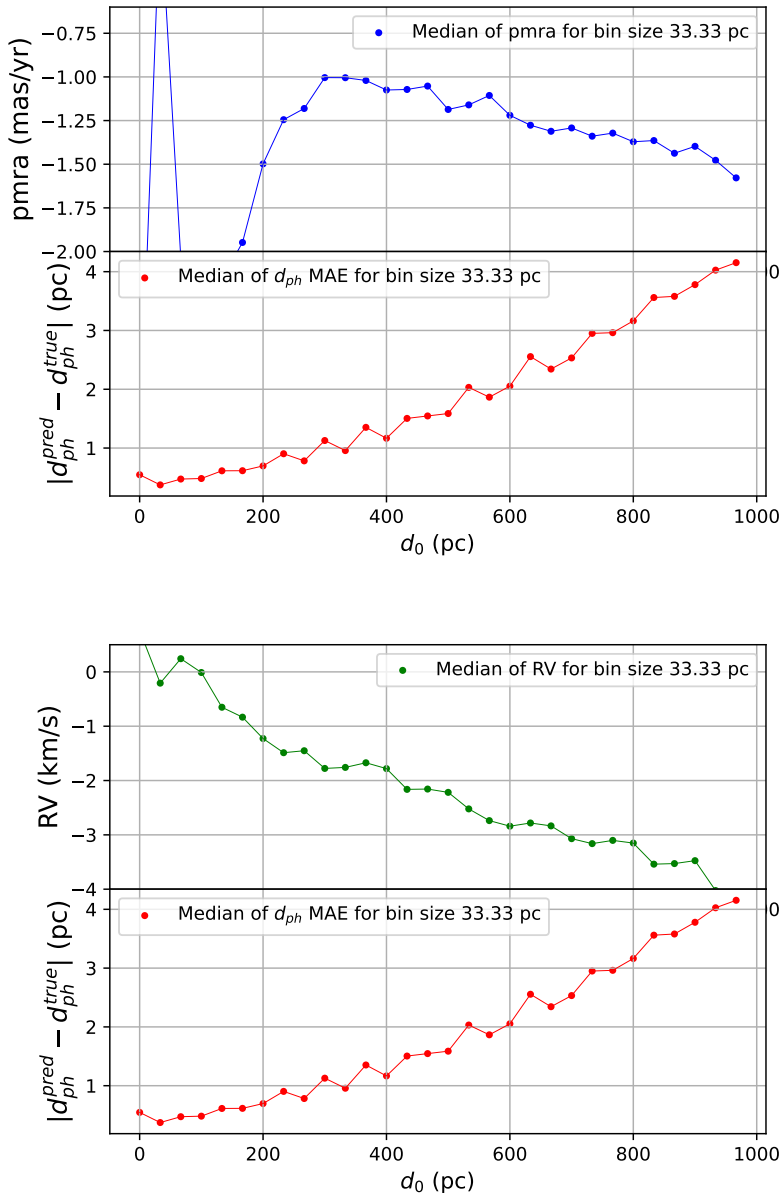


Figure 30: Top: Compares the pmra (mas/yr) with d_{ph} performance against current distance d_0 for the test data, and wiggles are seen between 300-700 pc. Bottom: Radial velocity (km/s) with d_{ph} performance is compared against current distance d_0 for test data, and wiggles for radial velocity are seen from 200 pc to 1000 pc.

be completed for about 4.9 million sources out of 12 million sources, they are compared to the full dataset in the 1 kpc range with `parallax_over_error`, greater than 5, which has about 12 million objects. The wiggles are weaker for

radial velocity and pmra in the full 12 million dataset as shown in figure 31 as compared to the test data. When the bin size is decreased to 10 pc as in figure 32, the fluctuations increase and do not seem to make any definite structure, suggesting it might be an effect of binning rather than some intrinsic feature. The histograms of training data did not seem to show any kind of wavy pattern. So, the other possibility about how these wiggles occurred in the performance is plotted in figure 29 is that the neural network may be overfitting to certain data points in the training set. This may be one of the reasons for the wiggles.

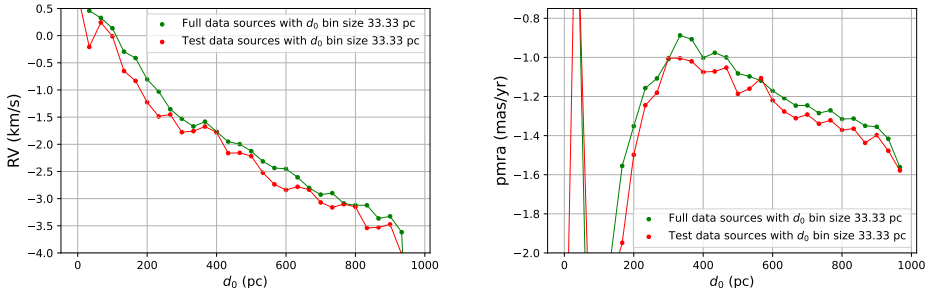


Figure 31: Left: The d_0 vs. RV of both test dataset and full data sources in 1 kpc range with `parallax_over_error` greater than 5 is compared to see whether the wiggles are present. Right: The d_0 vs. $pmra$ of both test dataset and full data sources in 1 kpc range with `parallax_over_error` greater than 5 is compared to see whether the wiggles are present. The wiggles are weaker in amplitude for the full data set for both radial velocity (RV) and $pmra$ compared to the test data.

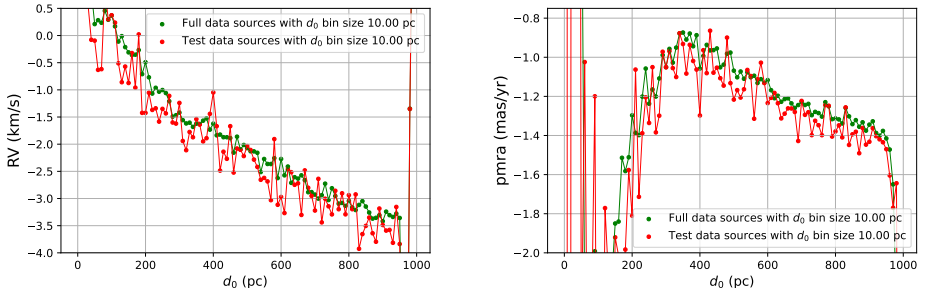


Figure 32: Left: The d_0 vs. RV of both test dataset and full data sources in 1 kpc range with bin size decreased to 10 pc. Right: The d_0 vs. $pmra$ of both the test dataset and full data sources in 1 kpc range with bin size decreased to 10 pc.

3.1.3 Close Encounter Time t_{ph} and σ_{tph}

The close encounter time t_{ph} is defined as the time taken by the star to reach the perihelion, and σ_{tph} is the uncertainty in t_{ph} measurement obtained by taking the standard deviation of the close encounter time of the surrogates. t_{ph} value of zero corresponds to the present time, negative values correspond to encounters in the past and positive value corresponds to encounters in the future. So, t_{ph} value extends from positive to negative unlike d_{ph} and v_{ph}.

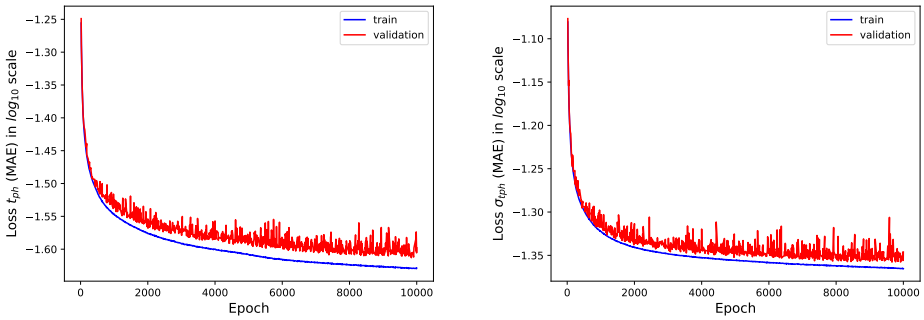


Figure 33: Left: loss of encounter distance in scaled units. the x-axis shows the epochs and the y-axis shows the loss of t_{ph} MAE in log_{10} scaled units. The purpose of this plot is to show the convergence of the network for t_{ph}. Right: loss of σ_{tph} log_{10} scaled units.

Figure 33 shows the loss curve of t_{ph} and σ_{tph} in log scaled units. The purpose of this loss curve is to show the convergence of the model. The training and validation loss, both converge and flatten without diverging much from each other ensuring that the model is not overfitting. The prediction against true values plot in Figure 34 shows that the prediction of t_{ph} and σ_{tph} are strongly correlated. The σ_{tph} comparatively has more variance in prediction than t_{ph}. The close encounter time t_{ph} has an overall bias of -22 kyr and a mean absolute error or scatter of 81 kyr, where the MAD of t_{ph} of the testing set is 5943 kyr. The bias and scatter of σ_{tph} is -0.55 kyr and 18.25 kyr respectively with MAD of the testing set of σ_{tph} being 379 kyr. Figure 35 shows the distribution of bias of the t_{ph} and σ_{tph}, and it can be seen that the model is slightly underpredicting both tph and σ_{tph}. When studying the performance of t_{ph} bias with respect to t_{ph}^{true}, it can be seen bias is larger for stars with large close encounter times. We are only interested in encounters that happened within $\pm$20 Myrs, because the galactic potential may go through significant change over the course of

the longer time period, so that makes the encounter predictions so unreliable even with orbital integrations. For close encounter times between [-20,000, 20,000] kyr, the t_{ph} bias varies between [-200, 200] kyr, and sources within this range have median d_{ph} scatter of 1.56 pc and median t_{ph} scatter of 73.10 kyr, refer figure 38. Figure 37 shows the relative error in uncertainty prediction against true close encounter time t_{ph}^{true} with colorbar representing scatter of d_{ph} in log scale. The relative error in t_{ph} uncertainty prediction is smaller for close encounters that are near $t_{ph} = 0$. And these sources also have a low scatter in d_{ph}. But, these sources with t_{ph}^{true} near zero have close encounter distance d_{ph}^{true} around 350 - 400 pc, which is also the median close encounter distance in the training data. Therefore, sources within this range have better performance.

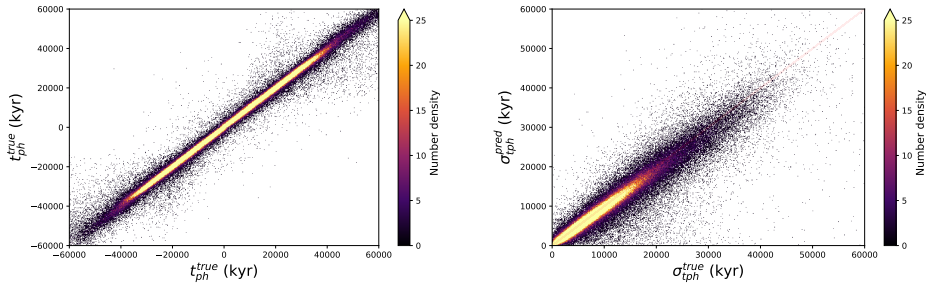


Figure 34: Left:t_{ph}^{pred} is plotted against t_{ph}^{true} in kyr, both prediction and true values correlate well since most of the points are concentrated in the diagonal red line. Right: σ_{tph}^{pred} is plotted against σ_{tph}^{true} in kyr. The points are concentrated along the diagonal line showing a strong correlation between prediction and true values.

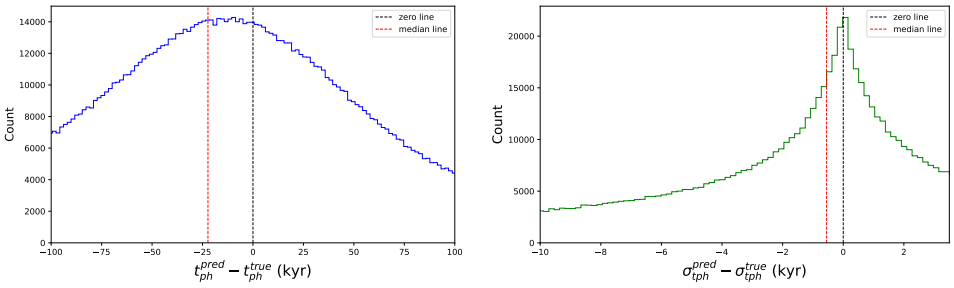


Figure 35: Left:$t_{ph}^{pred} - t_{ph}^{true}$ distribution with 85 to 15 percentile cut. The red line shows the median bias of close encounter time t_{ph}. Right: Bias distribution for the σ_{tph} with 65 to 35 percentile cut.

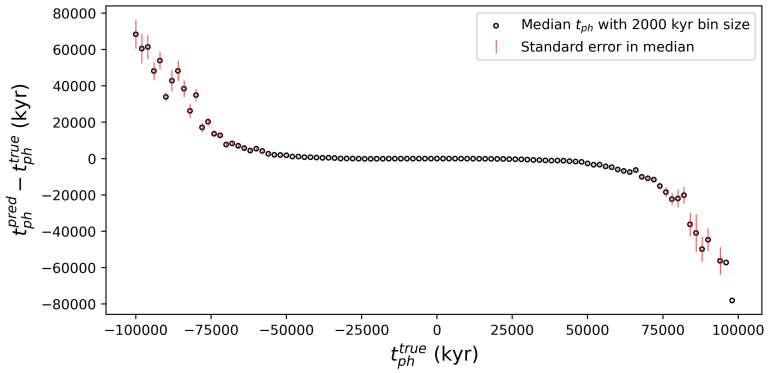


Figure 36: Each point represents the median t_{ph} bias of the stars with 2000 kyr bin. The red bar represents the standard error in median value.

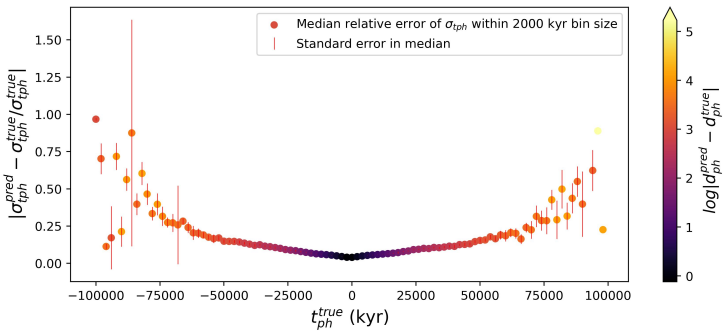


Figure 37: Each point represents the median of relative error of σ_{tph} prediction with colorbar representing d_{ph} prediction scatter in log scale.

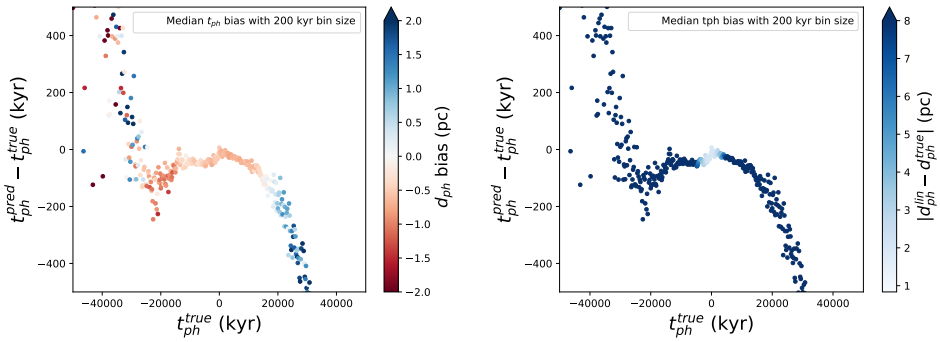


Figure 38: Left:$t_{ph}^{pred} - t_{ph}^{true}$ against t_{ph}^{true} bin plot, with a color bar showing the bias of encounter distance d_{ph}. Each point in this plot has a median t_{ph} bias of stars in 200 kyr t_{ph}^{true} bin size. Encounters within the -20,000 to 20,000 kyr have a median d_{ph} scatter of 1.56 pc and a median t_{ph} scatter of 73.10 kyr. Right: $t_{ph}^{pred} - t_{ph}^{true}$ against tph_{true} bin plot with colorbar indicating curvature $|d_{ph}^{lin} - d_{ph}^{true}|$.

3.1.4 Close Encounter Velocity v_{ph} and σ_{vph}

The close encounter velocity v_{ph} is defined as the velocity of the star at the time of its perihelion. vph defined as $\sqrt{v_{rph}^2 + v_{tph}^2}$ is always a positive quantity like close encounter distance d_{ph}. The v_{ph} prediction has a median bias of 0.0026 km/s and a scatter of 0.27 km/s, where the MAD of test data of v_{ph} is 16.61 km/s. Figure 39 shows the loss curve of v_{ph} and σ_{vph} in scaled units. The

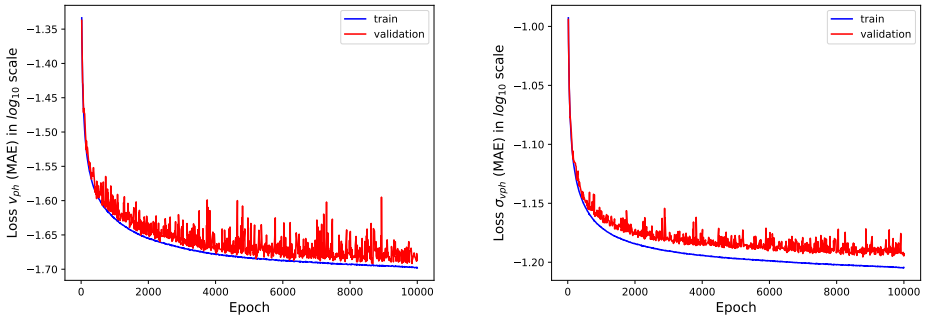


Figure 39: Left: loss of encounter distance in scaled units. the x-axis shows the epochs and the y-axis shows the loss of v_{ph} MAE in log_{10} scaled units. The purpose of this plot is to show the convergence of the network for v_{ph}. Right: loss curve of σ_{vph} with y-axis showing loss in MAE in log_{10} scaled units.

purpose of this loss curve is to show the convergence of the model. The training and validation loss, both converge and flatten without diverging much from each other ensuring that the model is not overfitting. The prediction against true values plot in Figure 40 shows that the prediction of v_{ph} and σ_{vph} are strongly correlated. The median bias of uncertainty of encounter velocity σ_{vph} is -0.000198 km/s and scatter of 0.0699 km/s, where MAD of test data of σ_{vph} is 1.11 km/s. Analyzing the relative error of both v_{ph} and σ_{vph} as a function of current distance shows that both of them have a high relative error when the current distance is smaller as shown in figure 41.

The median initial velocity and close encounter velocities of sources only have a smaller difference of about 0.97 km/s. This is because most of the sources are only weakly accelerated as mentioned in section 2.3.1. This makes the v_{ph} prediction comparatively easier for the Neural Network. When the scatter of v_{ph} prediction is analyzed based on distance bins, the wiggles are visible, refer to figure 42.

Histograms in fig.43 show the distribution of inputs parallax distance, tan-

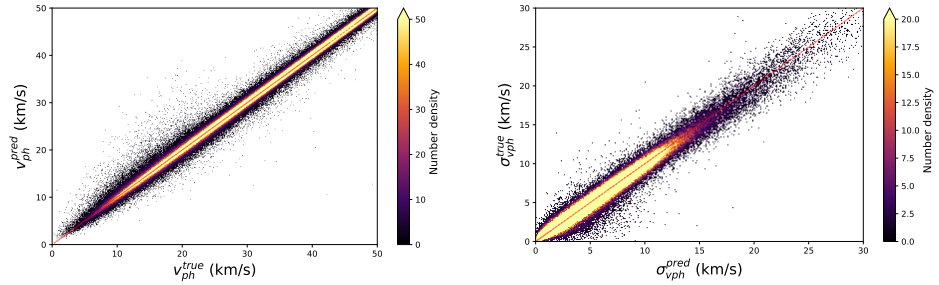


Figure 40: Left:v_{ph}^{pred} is plotted against v_{ph}^{true} in km/s, both prediction and true values correlate well since most of the points are concentrated in a diagonal red line. Right: σ_{vph}^{pred} is plotted against σ_{vph}^{true} in km/s. The points are concentrated along diagonal lines showing a strong correlation between prediction and true values.

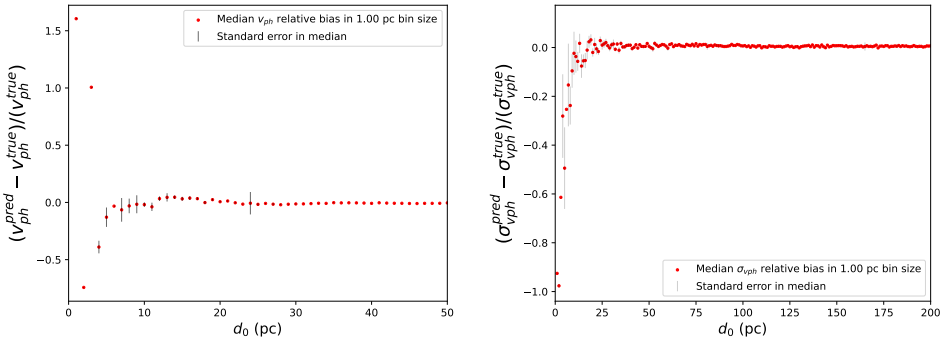


Figure 41: Relative bias of v_{ph} and σ_{vph} as a function of current distance d_0 shows that stars close to the Sun have a higher relative bias.

gential velocity, radial velocity, and encounter parameters d_{ph}, t_{ph} and v_{ph} for sources that are inside 95% confidence interval (CI) of the bias of each encounter parameters. All the sources that are common inside 95% CI of d_{ph}, v_{ph}, and t_{ph} bias (prediction - true) distribution are taken and compared with the distribution of input and output parameters of sources that are not inside 95% CI. The total sources in the test dataset is 1,967,582 and from this 1,718,800 sources are inside 95% CI of the encounter parameters prediction bias. 248,782 sources are not inside this confidence interval and their distribution has some specific properties which explains the performance issues with the model. The radial velocity distribution shows that sources outside 95% CI in of bias have radial velocities, close encounter time t_{ph}, and close encounter distance d_{ph} near zero. These are the sources that are already in close encounter positions currently. The training data mostly contains sources with d_{ph} of 150 to 750

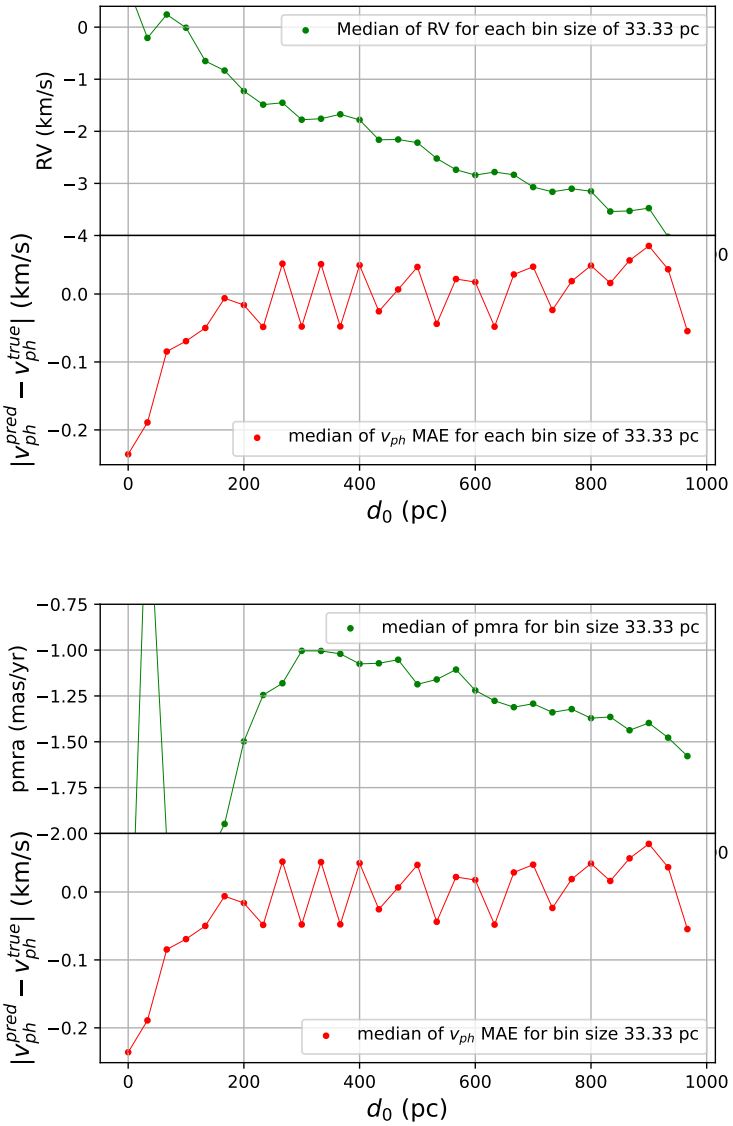


Figure 42: v_{ph} scatter as a function of current distance shows the wiggles in regular intervals. The wiggles are also visible in the input radial velocity and pmra of the test data.

pc compared to sources that are close or much farther away. so the network is mostly trained for these distance ranges and most of the sources in these ranges do not come closer than 1 pc.

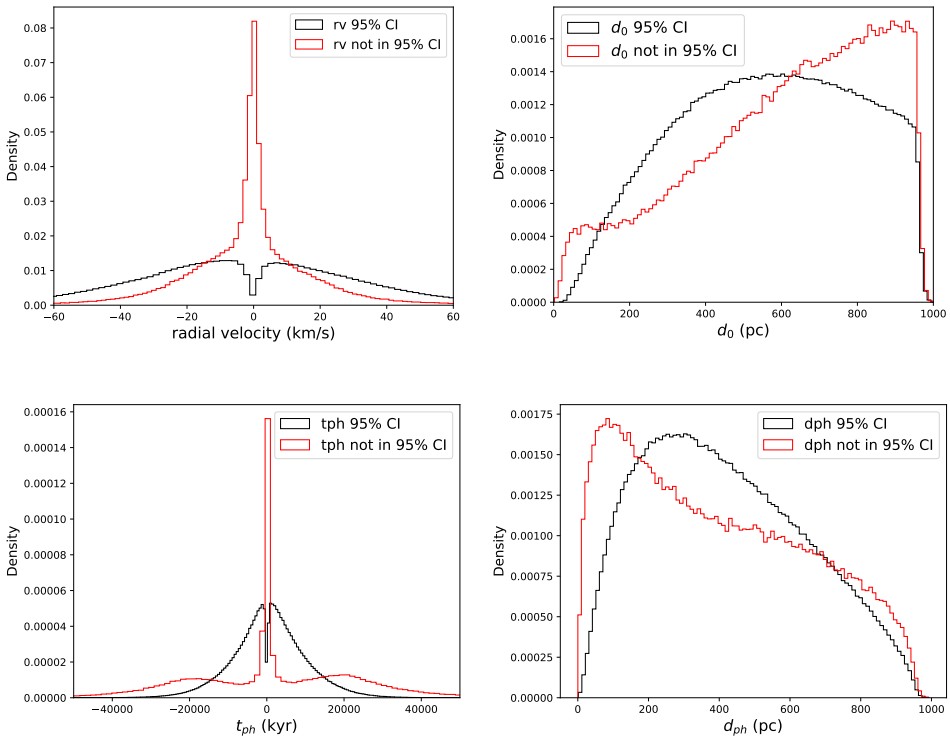


Figure 43: The sources in 95% CI of bias are separated from the test data and compared using a normalised histogram with rest of the sources.

3.1.5 Correlations between encounter parameters

The model predicts correlations between encounter parameters d_{ph}, t_{ph} and v_{ph}. The correlation coefficients for the encounter parameters are calculated using the Pearson correlation coefficient formula,

$$\rho_{x,y} = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum(x_i - \bar{x})^2 \sum(y_i - \bar{y})^2}} \tag{3.2}$$

The correlation coefficients $\rho(t_{ph}, d_{ph})$, $\rho(t_{ph}, v_{ph})$ and $\rho(d_{ph}, v_{ph})$ are distributed between -1 and +1. The scatter of correlation coefficient for $\rho(t_{ph}, d_{ph})$, $\rho(t_{ph}, v_{ph})$ and $\rho(d_{ph}, v_{ph})$ is 0.019, 0.022 and 0.020 respectively. The MAD of the test set for these coefficients is 0.83 for $\rho(t_{ph}, d_{ph})$, 0.79 for $\rho(t_{ph}, v_{ph})$ and 0.25 for $\rho(d_{ph}, v_{ph})$. Figure 44 shows the loss curve for each of the three correlation outputs and it can be seen that both training and validation curves converge without much divergence from each other. Figure 45 is the prediction against true value density plots of all three correlations. Most of the points lie in the

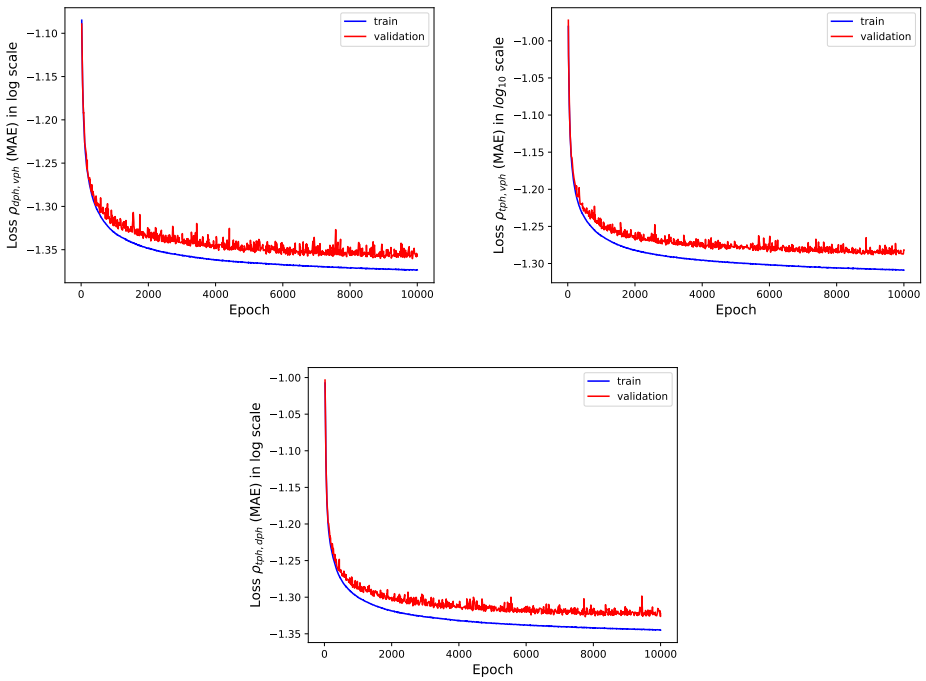


Figure 44: Loss curve of $\rho(d_{ph}, v_{ph})$, $\rho(t_{ph}, v_{ph})$ and $\rho(t_{ph}, d_{ph})$ with x-axis showing number of epochs and y-axis showing MAE loss in log_{10} scale. The blue line shows the training curve and the red line shows the validation curve

diagonal line showing that predictions and true values mostly correlate and the divergence of the points away from the diagonals is almost the same for all three correlations. Figure 46 shows the correlation coefficient as a function of current distance d_0. The correlation coefficient $\rho(t_{ph}, d_{ph})$ has a general negative bias and it stabilizes after 600 pc. For $\rho(t_{ph}, v_{ph})$ has a positive bias and is stable around 300 to 800 pc and for $\rho(d_{ph}, v_{ph})$ the bias fluctuates around zero till 600 pc and then becomes more negative.

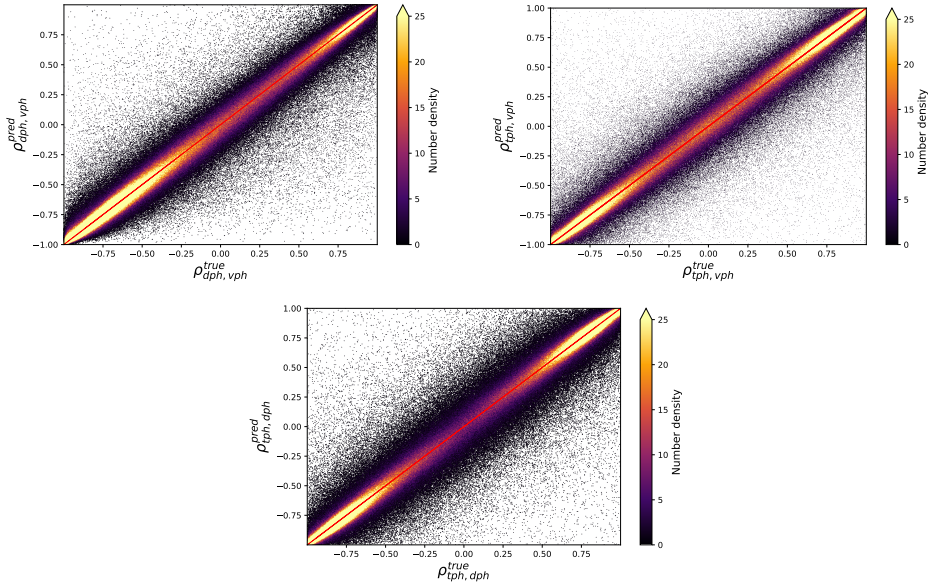


Figure 45: ρ predictions are plotted against ρ true values. Points mostly lie around diagonal showing a strong correlation between predicted and true values.

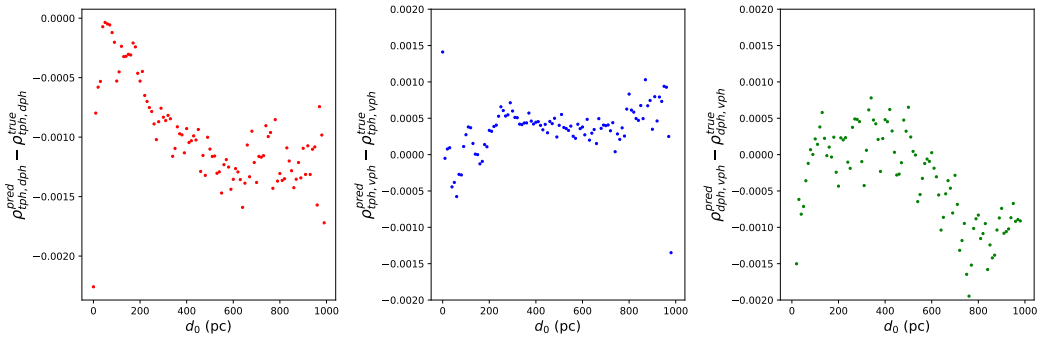


Figure 46: Bias of correlation coefficients against current distance d_0, with each point being the median of bias of stars in 10 pc bin size.

The table 2 gives the overall bias and scatter of each output of the neural network.

Parameter	Bias	Scatter	MAD of true values
t_{ph}	-22 kyr	81 kyr	5943 kyr
σ_{tph}	-0.55 kyr	18 kyr	379 kyr
d_{ph}	-0.45 pc	1.7 pc	175.57 pc
σ_{dph}	-0.012 pc	0.52 pc	9.97 pc
v_{ph}	0.0026 km/s	0.27 km/s	16.61 km/s
σ_{vph}	-0.00019 km/s	0.069 km/s	1.11 km/s
$\rho(t_{ph}, d_{ph})$	-0.00099	0.019	0.83
$\rho(t_{ph}, v_{ph})$	0.00039	0.022	0.79
$\rho(d_{ph}, v_{ph})$	-0.00022	0.020	0.25

Table 2: Bias and scatter of encounter parameter for the model trained in 1 kpc data. The encounter parameters are t_{ph}: median encounter time (kyr), d_{ph}: median encounter distance (pc), and v_{ph}: median encounter velocity (km/s).

4 Discussion & Conclusion

The stellar encounters have been studied extensively using modern numerical techniques. The current numerical techniques successfully predict stellar encounters with great accuracy. But, computationally this is so expensive. Because of how computationally expensive this technique is, some of the stars are ignored from high-resolution numerical integration as mentioned in section 2.3.1. For example, the high-resolution numerical integration in Bailer-Jones (2022) is only done for 6407 out of 29,947,046 stars that have radial velocity available and parallax SNR greater than 5. The 6407 stars were filtered out from 29,947,046 stars using linear motion approximation coupled with low-resolution numerical integration. Hence, there is a possibility of missing certain close encounter candidates. In addition, Gaia DR4 is expected to contain more than 100 million stars with RVS spectra limiting magnitude up to 16.2 mag Katz et al. (2023), and the number as accuracy of measurements is only expected to increase with Gaia DR5. To get an idea regarding the efficiency; if it takes 10 seconds to complete high-resolution numerical integration for a single star with 1000 surrogates, then it takes around 31 years to complete them for 100 million stars without parallelization. For a neural network, this would only take around 113 days assuming one forward pass takes 0.1 seconds. This estimate is only expected to reduce when the model is run on a GPU.

This study is a proof of concept of whether neural networks can be used to predict the perihelion parameters and uncertainties of the star. If a neural network can be trained to map the initial phase space coordinates to the close encounter parameters t_{ph}, d_{ph} and v_{ph}, then the computational expense of this problem can be reduced as mentioned in the first paragraph. Since a neural network is a universal function approximator, it can learn this input-to-output mapping from the training set and estimate the encounter parameters with their uncertainties and correlations. The initial plan was to select every source that has radial velocity available and a parallax signal-to-noise ratio greater than 5 within 1 kpc of the Sun. Due to technical issues, numerical integration could only be performed for 4,919,099 sources. The 66% of the sources were selected at random from this to make a training set of 2,951,517 objects. The model was then trained on this data and evaluated on a test set of 1,967,582 sources and achieved an overall scatter of 1.7 pc for d_{ph}, 81 kyr for t_{ph}, and

0.27 km/s for v_{ph}. These predictions were good enough in the sense that these scatter values were smaller than the spread (MAD) of the test sample which was 5943 kyr for t_{ph}, 175.57 pc for d_{ph} and 16.61 km/s for v_{ph}. The perihelion distance that is interesting for us comes in the 1 pc range (Oort cloud radius is around 0.5 pc). In that case, the scatter of 1.7 pc is large. But, for stars within 100 pc of the Sun, the scatter of d_{ph}, t_{ph} and v_{ph} is 0.44 pc, 67.3 kyr, and 0.33 km/s respectively. whereas the MAD of the d_{ph}, t_{ph} and v_{ph} for these sources is 18.1 pc, 761 kyr, and 16.39 km/s respectively. Overall, the scatter of d_{ph} has an upward trend with true d_{ph}. And, from current distance vs. relative error of d_{ph} plots in figure 24, it has been understood that most stars that are close to the Sun have a higher relative error in d_{ph}. For example, the stars that come closer than 10 pc of the Sun constitute only 0.008% of the training set, therefore the predictions in this distance range give a comparatively large scatter. This is caused because these stars only constitute 0.08% of the training set.

Overall, this model performed well for a smaller distance range with significant performance loss in a higher distance range. If the model could be retrained by giving more weight to the samples in a smaller distance range (say 20 pc), then it could act as a greater filter, to see possible encounter candidates that could later be used for high-precision numerical integration. This model could also give way to the development of neural networks that can be used for finding star-to-star close encounters. Star-to-star close encounters are much more computationally expensive because they involve taking every pair of stars within a given distance. For instance, just around 100 pc itself there are 169741 sources with radial velocity and parallax SNR greater than 5, which gives around 14 billion stars for orbital integration. Also, the number of stars increases with distance d by a factor of d^3. Therefore, the use of machine learning methods becomes essential in these situations.

There are other networks, that could be experimented with for orbital integration. Physics-informed Neural Networks (PINNs) are an example. The PINNs are trained not only on the statistics of the data but also on the physical laws that govern the object's motion is also taken into account. This helps PINN to converge based on the physical law that it is given by the differential equation. There are studies in the direction of applying PINNs to orbital integration Scorsoglio et al. (2023). This could be extended to orbital inte-

gration of sources from Gaia. Another possibility is using normalizing flow, which is a deep learning model that learns the transformation of a probability distribution from simple to complex distribution Kobyzev et al. (2021). The stellar encounters problem can be treated as a transformation of a probability distribution, because, the star with its surrogates has an initial distribution, and at the time of close encounter the distribution changes based on the close encounter times of the different surrogate stars. This transformation in distribution can be modeled using normalizing flows.

References

Bailer-Jones, C. (2015). Close encounters of the stellar kind. *Astronomy & Astrophysics*, 575:A35.

Bailer-Jones, C. (2022). Stars that approach within one parsec of the sun: New and more accurate encounters identified in gaia data release 3. *The Astrophysical Journal Letters*, 935(1):L9.

García-Sánchez, J., Weissman, P., Preston, R., Jones, D., Lestrade, J.-F., Latham, D., Stefanik, R., and Paredes, J. (2001). Stellar encounters with the solar system. *Astronomy & Astrophysics*, 379(2):634–659.

Gullberg, D. and Lindegren, L. (2002). Determination of accurate stellar radial-velocity measures. *Astronomy & Astrophysics*, 390(1):383–395.

Hornik, K., Stinchcombe, M., and White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366.

Kaib, N. A. and Raymond, S. N. (2024). Passing stars as an important driver of paleoclimate and the solar system's orbital evolution. *The Astrophysical Journal Letters*, 962(2):L28.

Katz, D., Sartoretti, P., Guerrier, A., Panuzzo, P., Seabroke, G., Thévenin, F., Cropper, M., Benson, K., Blomme, R., Haigron, R., et al. (2023). Gaia data release 3-properties and validation of the radial velocities. *Astronomy & Astrophysics*, 674:A5.

Kingma, D. P. and Ba, J. (2017). Adam: A method for stochastic optimization.

Knie, K., Korschinek, G., Faestermann, T., Dorfi, E., Rugel, G., and Wallner, A. (2004). F 60 e anomaly in a deep-sea manganese crust and implications¡? format?¿ for a nearby supernova source. *Physical Review Letters*, 93(17):171103.

Kobyzev, I., Prince, S. J., and Brubaker, M. A. (2021). Normalizing flows: An introduction and review of current methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(11):3964–3979.

Lindegren, L., Klioner, S., Hernández, J., Bombrun, A., Ramos-Lerate, M., Steidelmüller, H., Bastian, U., Biermann, M., de Torres, A., Gerlach, E., et al. (2021). Gaia early data release 3-the astrometric solution. *Astronomy & Astrophysics*, 649:A2.

Mathews, J. H., Fink, K. D., et al. (2004). *Numerical methods using MATLAB*, volume 4. Pearson prentice hall Upper Saddle River, NJ.

McMillan, P. J. (2016). The mass distribution and gravitational potential of the milky way. *Monthly Notices of the Royal Astronomical Society*, 465(1):76–94.

Miyamoto, M. and Nagai, R. (1975). Three-dimensional models for the distribution of mass in galaxies. *Astronomical Society of Japan, Publications, vol. 27, no. 4, 1975, p. 533-543.*, 27:533–543.

Oort, J. H. (1950). The structure of the cloud of comets surrounding the Solar System and a hypothesis concerning its origin. , 11:91–110.

Plummer, H. C. (1911). On the Problem of Distribution in Globular Star Clusters: (Plate 8.). *Monthly Notices of the Royal Astronomical Society*, 71(5):460–470.

Ramirez-Preciado, V. G., Roman-Zuniga, C. G., Aguilar, L., Suarez, G., and Downes, J. J. (2018). Kinematic identification of young nearby moving groups from a sample of chromospherically active stars in the rave catalog. *The Astrophysical Journal*, 867(2):93.

Ruder, S. (2016). An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*.

Schönrich, R., Binney, J., and Dehnen, W. (2010). Local kinematics and the local standard of rest. *Monthly Notices of the Royal Astronomical Society*, 403(4):1829–1833.

Scorsoglio, A., Ghilardi, L., and Furfaro, R. (2023). A physic-informed neural network approach to orbit determination. *The Journal of the Astronautical Sciences*, 70(4):25.

Siraj, A. and Loeb, A. (2021). Breakup of a long-period comet as the origin of the dinosaur extinction. *Scientific Reports*, 11(1):3803.

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M., Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., Carey, C. J., Polat, İ., Feng, Y., Moore, E. W., VanderPlas, J., Laxalde, D., Perktold, J., Cimrman, R., Henriksen, I., Quintero, E. A., Harris, C. R., Archibald, A. M., Ribeiro, A. H., Pedregosa, F., van Mulbregt, P., and SciPy 1.0 Contributors (2020). SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272.

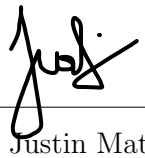
Vokrouhlickỳ, D., Nesvornỳ, D., and Dones, L. (2019). Origin and evolution of long-period comets. *The Astronomical Journal*, 157(5):181.

Vuik, K., Vermolen, F., and van Gijzen, M. (2023). *Numerical methods for ordinary differential equations*.

Erklärung:

Ich versichere, dass ich diese Arbeit selbstständig verfasst habe und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Heidelberg, September 1, 2024



Justin Mathew