GROUP2306 –Estimation and Analysis of Merging Binary Black Holes Time Delay

Edoardo Renato Bucalo, Camilla Forza, Anna Garbo, and Federico Ruffini
(Dated: June 25, 2023)

Binary black holes are nowadays one of the most interesting subject in astrophysics, in particular their study is strictly correlated with another of the most challenging fields of interest: gravitational waves. The work proposed in this paper, starting from three populations of stellar binaries with three different metallicities generated by SEVN [1], implements and compares different methods to find an estimation of time delays and to study their distribution. Results suggest that, apart from the Runge-Kutta method, which is taken as the reference method, the machine-learning based XGBoost method is able to ensure the best performances both in terms of accuracy and of computational time. The distributions of the time delays are studied to search for anomalies and different binning approaches are tried. The Bayesian Blocks binning method turns out to be the most informative. The power-law behaviour of the distributions is verified, with fluctuations of the exponent vanishing in high time delays limit.

INTRODUCTION

In the universe, binary systems are initially couples of stars rotating not only around their own axis, but also along elliptical orbits sharing the same center. If these stars both evolve into black holes, they form a binary black holes system, which is the case we are interested in. In this paper, the time required for the stellar binary evolution to form a binary of black holes is called t_{form}. Some of the systems, satisfying particular conditions, merge into a single black hole in a time called t_{GW}. This is the time required, due to the Gravitational Wave decay, to bring the black holes close enough to merge. The sum of the two defined time intervals is the t_{delay}:

$$t_{delay} = t_{form} + t_{GW} \tag{1}$$

The calculation of t_{GW} is made solving the Peters' equations:

$$\frac{da}{dt} = -\frac{64}{5}\frac{G^3 Mm(M+m)}{c^5 a^3 (1-e^2)^{7/2}}\left(1 + \frac{73}{24}e^2 + \frac{37}{96}e^4\right) \tag{2}$$

$$\frac{de}{dt} = -\frac{304}{15}e\frac{G^3 Mm(M+m)}{c^5 a^4 (1-e^2)^{5/2}}\left(1 + \frac{121}{304}e^2\right) \tag{3}$$

where a is the semimajor axis, e is the eccentricity, M and m are the two masses of the black holes. These equations describe both the circularizing and orbital decay. After solving the Peters' equations, t_{GW} is defined as the time at which the periastrum becomes smaller than three times of the Schwarzchild's radius. The purpose of this work is to compare different methods to estimate the time delays of the SEVN populations. In particular, Euler method and Runge-Kutta method are implemented to estimate the time delay from a direct resolution of the Peters' equations. Two methods based on machine learning techniques, i.e. XGBoost and DNN, are used to estimate the time delay without solving the Peters' equations. The distributions of the obtained time delays are finally analysed, looking for hints about the physics underlying them.

METHODS

Time delays estimation

Two methods are implemented to solve Peters' equations: Euler method and fourth-order Runge-Kutta method (RK4). The Euler method consists in approximating the curve with a polygonal one where, given a differential equation $\frac{dy}{dt}$, the initial values t_0 and $y_0 = y(t_0)$, and the time step h, the y_{n+1} values is updated as follows:

$$y_{n+1} = y_n + hf(t_n, y_n) \tag{4}$$

where $t_{n+1} = t_n + h$ and $\frac{dy(t_n)}{dt} = f(t_n, y_n)$.
The second technique is fourth-order Runge-Kutta method (RK4). Taking the same differential equation with the same initial conditions, the approximated solutions are found implementing an algorithm which finds the y_{n+1} element summing the present value y_n to the weighted average of four increments. The following relations explicit the passages:

$$y_{n+1} = y_n + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4) \tag{5}$$

$$t_{n+1} = t_n + h \tag{6}$$

where k_1, k_2, k_3 and k_4 are defined as follows:

$$k_1 = f(t_n, y_n) \tag{7}$$

$$k_2 = f\left(t_n + \frac{h}{2}, y_n + h\frac{k_1}{2}\right) \tag{8}$$

$$k_3 = f\left(t_n + \frac{h}{2}, y_n + h\frac{k_2}{2}\right) \tag{9}$$

$$k_4 = f(t_n + h, y_n + hk_3) \tag{10}$$

These two techniques usually work with fixed increments of time. The choice to switch to an adaptive time step, in our case, is motivated by the very large time scale of the time delays, which can be of the order of the age of the universe. Adapting the time step gives the possibilities to solve very precisely the Peters' equations over such time ranges without running out of memory. In particular, the adaptive time step technique increases or decreases the time step h whether the relative variation of the y function is smaller or bigger of a certain tolerance. In our case, the tolerance is chosen studying a small subset of the original data-set ($\sim 1\%$). Using this data-set, the tolerance is maximized requiring that the mean relative error in the time delays, with respect to a benchmark case with $tol = 10^{-3}$, keeps under the 1%. Plot of the relative error, for the analysed tolerances, are reported for both RK4 and Euler in the following picture:

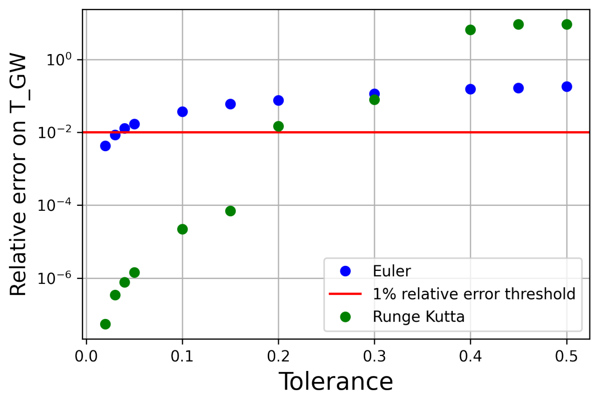


FIG. 1: Adaptive time step tolerance vs relative error with respect to benchmark case. Grid Search to find the best tolerance that keeps the error under 1%.

The chosen tolerances are:

$$tol_{Eu} = 3 \cdot 10^{-2} \tag{11}$$

$$tol_{RK4} = 0.15 \tag{12}$$

As an additional step, in order to reduce the computational time needed for the analysis of the full data-set, two unsupervised machine learning techniques are implemented: a Deep Neural Network (DNN) and XGBoost Regressor. XGBoost (Extreme Gradient Boosting) is a machine learning algorithm used for classification and regression tasks. It combines weak models, like decision trees, into an ensemble to provide accurate predictions. XGBoost iteratively trains models to correct mistakes made by previous ones, improving accuracy. It also uses regularization techniques to prevent overfitting and allows tuning hyperparameters for better performance. Deep Neural Networks, or deep learning, are machine learning models consist of interconnected layers of artificial neurons. DNNs excel at capturing complex patterns in large data-sets. They use forward propagation for predictions and back-propagation with gradient descent to adjust internal parameters. Training DNNs can be computationally intensive and requires labeled data. In our case, a training data-set is built generating 150000 rows from uniform distributions between the minimum and the maximum for each variable of the Peters' equations in the original data-set: major mass M, minor mass m, logarithm of the semi-major axis a and eccentricity e. The Runge-Kutta method is applied to this dataset to estimate the merging time t_{GW}, and then the machine is trained with, as input variables, the eccentricity and the approximate t^0_{GW} expression for circular orbits ($e = 0$):

$$t^0_{GW} = \frac{5}{256}\frac{c^5}{G^3}\frac{a^4}{M_1 M_2 (M_1 + M_2)} \tag{13}$$

The reason behind the choice of this parameter was motivated by the fact that the correct t_{GW} is expected to be made up of the sum of t^0_{GW} with a function of the eccentricity only. In this way the machine is facilitated in understanding the relation between the parameters of each system and its time delay. The hyper-parameters for both XGBoost and DNN models were optimizated by Grid-search, and are reported in the following tables:

XGBoost	**DNN**
$l_{rate} = 0.2$	opt=Adam($l_{rate} = 0.0001$)
$reg_\lambda = 0.01$	$n_{epochs} = 300$
$max_{depth} = 16$	$batch_{size} = 256$
$n_{estimator} = 64$	

TABLE I: Hyper-paramameter found for machine learning techiques, DNN & XGBoost

Layer (type)	**Output Shape**	**Parameters (#)**
dense_90 (Dense)	(None,32)	96
dense_91 (Dense)	(None,64)	2112
dense_92 (Dense)	(None,64)	4160
dense_93 (Dense)	(None,32)	2080
dense_94 (Dense)	(None,1)	33
Total parameters: 8481		
Trainable parameters: 8481		
Non-trainable parameters: 0		

TABLE II: This is the DNN structure made by a Sequential() model with 3 hidden layers Dense(). The first 2 have 64 nodes, while the third one has only 32, and all of them has a 'relu' activation function. The input has dimension 2 (Eccentricity and approximated Merge Time), 32 nodes and the same activation function. The output has dimension 1 (Time Merge) with the 'linear' activation function.

Data Analysis

The aim of the Data Analysis section is to analyse the distribution of the time delays obtained with the methods explained above. It is important to point out that the time delays analysed are only the ones smaller than the age of the universe. The starting point is represented by the fact that the time delays are expected to be distributed as a power-law [1]. For this reason, we decided to fit the histograms of the normalized distributions with a sum of power-laws having different slopes. Subsequently, we analysed each subset of the distributions having a given slope and tried to correlate the time delays and eccentricities of these subset of the systems with the exponent of the power-law. Furthermore, to realize the histograms of the time delays distributions, a fundamental issue was represented by the choice of the number of bins in the histograms of the distributions. Two methods have been used:

- Arbitrary number of 50 logarithmic bins.

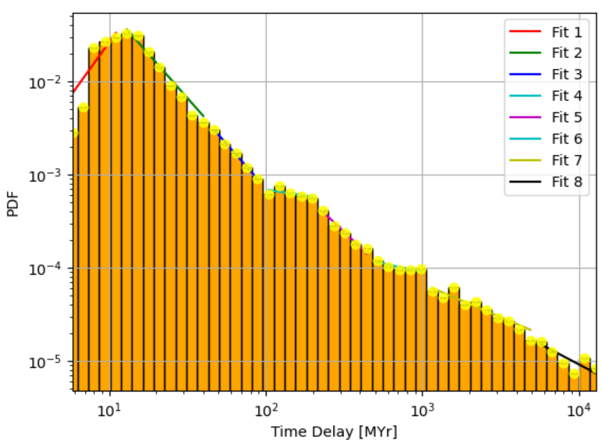


FIG. 2: Histogram of time delays distribution estimated with Runge-Kutta method, 50 logarithmic bins.

- Bins adapted with the Bayesan Blocks algorithm. The Bayesian Blocks algorithm allows for non-arbitrary binning of the histogram. It is a powerful method primarily developed and applied in astrophysics for data discretization. It is based on statistical and Bayesian principles and, through the optimization of a fitness function, it can adaptively identify breakpoints based on the data's structure. This enables the detection of regions with significant changes or relevant temporal/spatial structures.

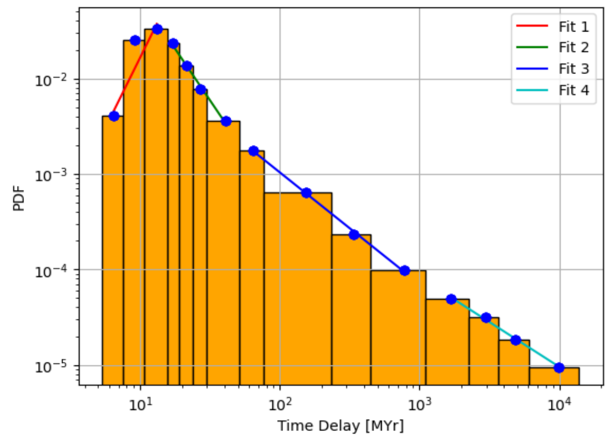


FIG. 3: Histogram of time delays distribution estimated with Runge-Kutta method, 15 bins estimated by Bayesian blocks algorithm

RESULTS

The search for the best method leads to the following results. A comparison of the performances of the studied methods, in terms of both accuracy and computational time, is provided by the following table:

Methods	**Time (s)**	**Relative error(s)**	**Max error(s)**
Euler	$3.54 \cdot 10^{-2}$	$3.89 \cdot 10^{-2}$	0.99
Runge-Kutta	$1.87 \cdot 10^{-2}$	benchmark	benchmark
XGBoost	$4.13 \cdot 10^{-6}$	$2.17 \cdot 10^{-2}$	$1.81 \cdot 10^{+1}$
DNN	$2.91 \cdot 10^{-5}$	$9.94 \cdot 10^{-2}$	$1.62 \cdot 10^{+3}$

TABLE III: Comparison of different methods for time delays estimation.

It is clear from the table that the methods based on the direct resolution of the Peters' equations require a computational time which is way higher with respect to the one of the machine learning methods. This is an expected result, since these methods learn a structure which connects the physical parameters of the system to the time delay, and don't need to solve Peters' equations. From an accuracy point of view, XGBoost is the most performing method, being able to minimize the relative error with respect to the RK4 method (see Table III). Thus, both in terms of accuracy and in terms of computational time, the most efficient method proved to be XGBoost, which represents the most valid alternative to Runge-Kutta method. If high accuracy needs to be reached, the latter is still the best method to use but its relevant computational time needs to be taken into account. In order to perform a deeper analysis, the relative error for each method has been studied in relation with the physical parameters of the systems, searching for possible correlations. Results show that the only variable which seems to be correlated to the relative error is the eccentricity. The correlation plot for the Euler method is shown in the following:

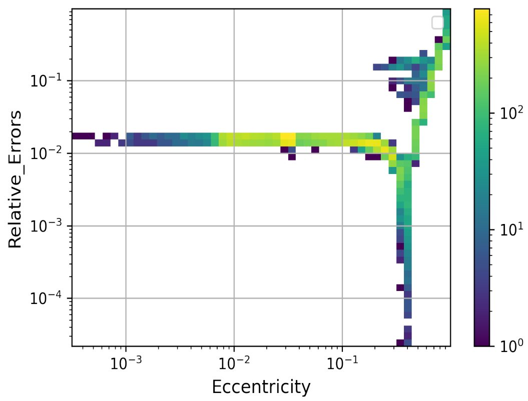


FIG. 4: Correlation between relative error for Euler method and eccentricity for Metallicity 0.1

The correlation plots for the other two metallicities, from which the same conclusion can be drawn, are reported in the Appendix. For the XGBoost and the DNN methods, the correlation is less evident, but is anyway reminiscent of the Euler one. The corresponding plots are in Appendix.

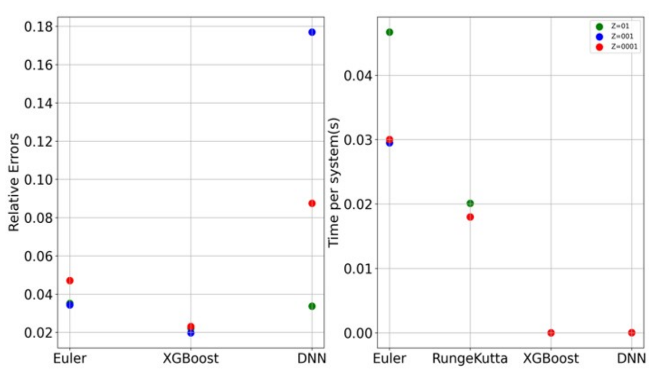


FIG. 5: On the left: Relative Error for each method and metallicity except Runge-Kutta which is the benchmark one. On the right: time execution for each method and metallicity. The interesting fact is that for numerical methods also execution time depends on metallicity.

For what concerns the analysis of the time delays distributions, the scatter-plots between slopes of different subsets of the histogram and the major features used to estimate the time delays are reported in the following. They allows to see if the analysis produces meaningful results and it has also an informative purpose.

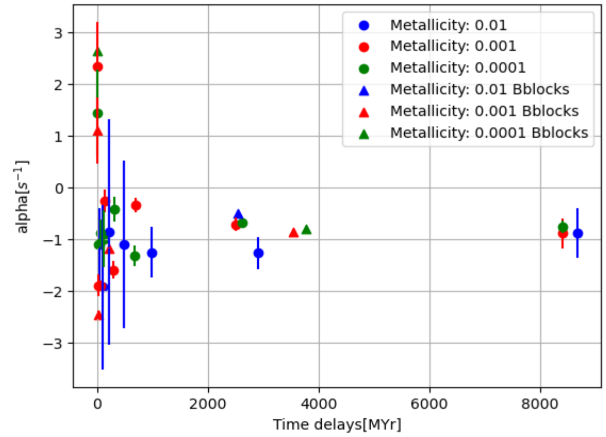


FIG. 6: Scatter-plot between different slopes of the power-law for interpolation of the bins in the histogram (alpha) and the time delay estimated with Runge-Kutta numerical method averaged in subset where the plot are considered.

It is observed that the time delays have a sort of asymptotic behaviour near $\alpha = -1$ (FIG. 6) which corresponds to the power-law expected for the distribution [1]. It is also worth highlighting that deviations from $\alpha = -1$ case happen at small time delays.

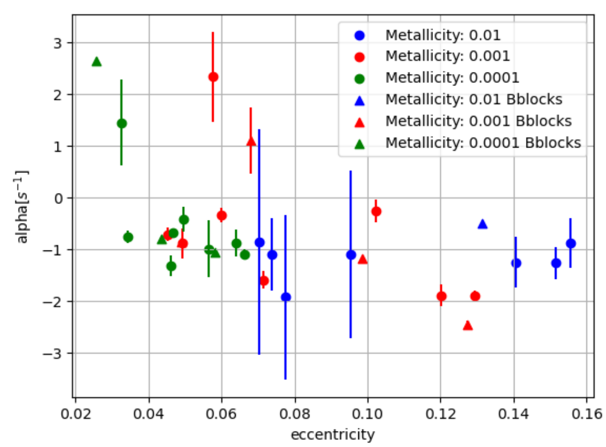


FIG. 7: Scatter-plot between different slopes of the power-law for interpolation of the bins in the histogram (alpha) and eccentricity .

In FIG. 7 the fact that all values fluctuate around $\alpha = -1$ is an hint of the physical significance of this exponent for the power-law. What is interesting is that the values are concentrated around a specific value of eccentricity, which could indicate a relation between the quantities.
The study does not provide clear hints of anomalies in the trend. The only possible guess could be done looking both to the 50-bins histogram and to the Bayesian Blocks: in the latter, the presence of the large block around 10^2 confirms the flat behaviour in that region.

No interpretations are provided in this paper, further studies should be conducted to formalize a meaningful hypothesis. The difficulty on proposing hypotheses relies on the Physics under the estimate of the t_{delay}. As mentioned in the Introduction, the t_{delay} is a quantity composed by the sum of two calculated times: the t_{form} and the t_{GW}. While the t_{GW} is well described by the Peters' equations, and the binary systems lose energy only through the emission of gravitational waves, the t_{form} is estimated combining a collection of different physical processes concerning both the single stellar evolution and the binary evolution into a black hole (i.e. wind mass transfer, common envelope, Roche-Lobe mass transfer, stellar tides, etc.). From a study on the ratio of binaries merging within the age of the universe for the three populations with metallicity: 0.01, 0.001 and 0.0001 we find the following results:

Z	0.01	0.001	0.0001
Ratio	1.47%	13.54%	16.58%

TABLE IV: Ratios calculated as the fraction of binaries merging into a black hole within the age of the universe over the total, for the corresponding metallicity.

Binaries of stars rich of Helium and Hydrogen, with a metallicity 0.0001 are more likely to merge into a black hole.

CONCLUSION

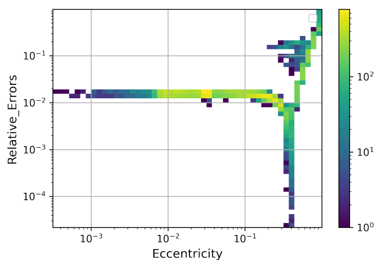
Different methods have been analysed to estimate time delays for binary black holes. They all gave satisfying results, and the most convenient method to use depends on the specific needs. In particular, if high accuracy is required, the Runge-Kutta method, based on the direct resolution of Peters' equations, is the most appropriate. Machine learning based methods, especially XGBoost Regressor, are effective at reducing the computational time at the expense of precision, and they are suitable if fast performances are needed. For what concerns the analysis of the time delays distributions, Bayesan Blocks binning proved to be the most appropriate method *to discretize the data*, being able to minimize the number of different slopes in the histograms. The power-law, instead, turned to converge towards the expected case of $\alpha = -1$ for time delays approaching the age of the universe. No significant anomalies or extra peaks in the distribution are found except for a region where the distribution seems to become more flat. The interpretation of this behaviour is difficult to guess due to the massive quantity of different physical models describing the binary black holes formation processes. Finally, the study of the t_{delays} spectrum for the three metallicities leads to some considerations about possible correlation between the power-law exponent and the eccentricity. The ratio of binary systems merging into a single black holes award the metallicity 0.0001, so the stellar binaries with this metallicity evolving in two black holes are more probable to merge within the age of the universe.

[1] Iorio, Giuliano, et al. "Compact object mergers: exploring uncertainties from stellar and binary evolution with sevn." Monthly Notices of the Royal Astronomical Society (2023): stad1630.

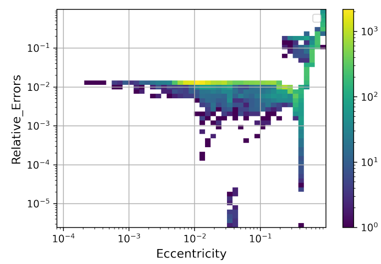
APPENDIX

In the appendix the images showing correlations between eccentricity and relative errors respect to different methods and metallicities are reported.

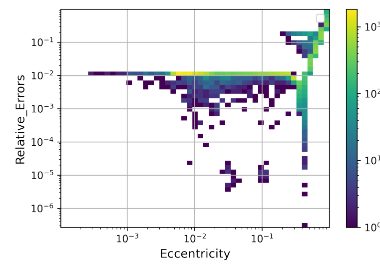
FIG. 8: Eccentricity vs Relative Error



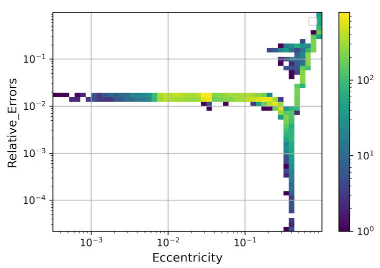
(a) SEVN-RK Metallicity001



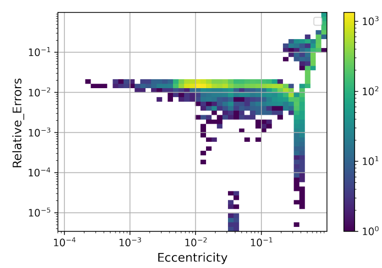
(b) SEVN-RK Metallicity0001



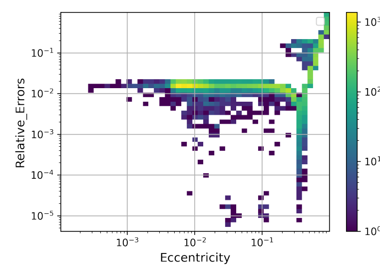
(c) SEVN-RK Metallicity01



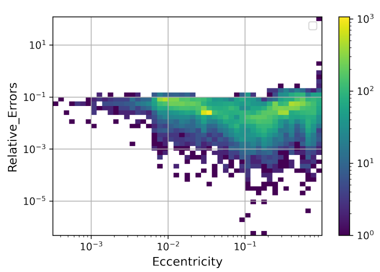
(d) Eu-RK Metallicity01



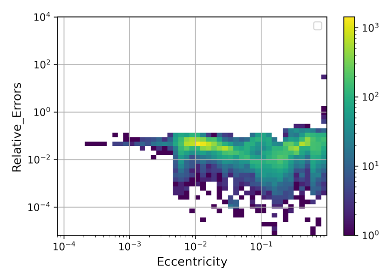
(e) Eu-RK Metallicity001



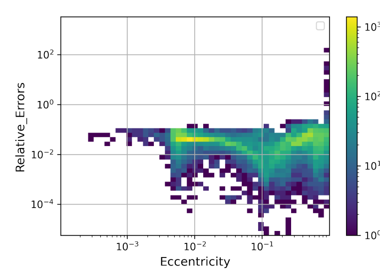
(f) Eu-RK Metallicity0001



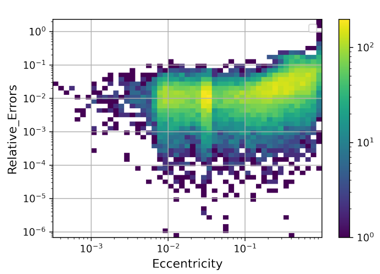
(g) XGB-RK Metallicity01



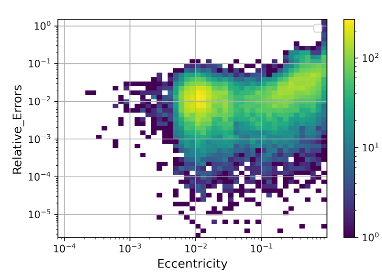
(h) XGB-RK Metallicity001



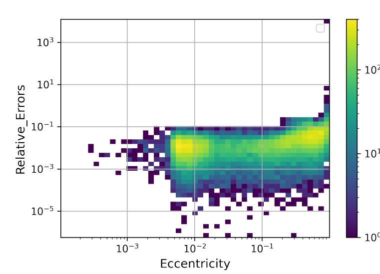
(i) XGB-RK Metallicity0001



(j) DNN-RK Metallicity01



(k) DNN-RK Metallicity001



(l) DNN-RK Metallicity0001