Serb. Astron. J. № 208 (2024), 17 - 27
DOI: https://doi.org/10.2298/SAJ2408017C

UDC 524.7
Original scientific paper

MODELING QUASAR VARIABILITY THROUGH SELF-ORGANIZING MAP-BASED NEURAL PROCESS

I. Čvorović - Hajdinjak

Department of Astronomy, Faculty of Mathematics, University of Belgrade
Studentski trg 16, 11000 Belgrade, Serbia

E–mail: *iva.cvorovic@gmail.com*

(Received: March 3, 2024; Accepted: May 29, 2024)

SUMMARY: Conditional Neural Process (QNPy) has shown to be a good tool for modeling quasar light curves. However, given the complex nature of the source and hence the data represented by light curves, processing could be time-consuming. In some cases, accuracy is not good enough for further analysis. In an attempt to upgrade QNPy, we examine the effect of the prepossessing quasar light curves via the Self-Organizing Map (SOM) algorithm on modeling a large number of quasar light curves. After applying SOM on the SWIFT/BAT data and modeling curves from several clusters, results show the Conditional Neural Process performs better after the SOM clustering. We conclude that the SOM clustering of quasar light curves could be a beneficial prepossessing method for QNPy.

Key words. Methods: data analysis – Galaxy: nucleus – Galaxies: active – Quasars: emission lines

1. INTRODUCTION

Quasars are the luminous central regions of galaxies. (LaMassa et al. 2015). Their emission lines are observed at all wavelengths from radio to gamma rays. In all parts of electromagnetic spectra, variability is strong, ranging from a few minutes to months (Zhang et al. 2023). Optical lines are produced in clouds of gas, rapidly moving in the potential of the black hole - broad-line region (Urry and Padovani 1995). Analyzing the optical variability in quasars has allowed us to understand the underlying physical processes (Wagner and Witzel 1995, Ulrich et al. 1997) and the structure of quasars (Antonucci 1993, Richards et al. 2006, Bonfield et al. 2010, Shang et al. 2011). Spectroscopic analyses have been instrumental in investigating the composition and kinematics of quasars. Complementary, quasar light curves provide an integrated temporal spectrum, capturing the aggregate spectral output as a function of time. These temporal profiles reflect the flux variations due to the dynamic astrophysical processes occurring in the accretion disk and surrounding environments of supermassive black holes. Therefore, analyzing these light curves is important for understanding the stochastic and transient events that characterize quasar variability which is essential for constraining the physical models that describe their emission mechanisms.

Quasar light curves are nonlinear, irregular, sparse, and could have gaps and signatures of extreme flares or noise. Such a structure could be challenging to model. Several tools have been developed and used for their modeling: Damped Random Walk (Kelly et al. 2009, Kozłowski 2017, Sánchez-Sáez et al. 2018), Gaussian Process (GP) (Shapovalova et al. 2017, Kovačević et al. 2019, Shapovalova et al. 2019), Autoencoders (Tachibana et al. 2020, Bank et al. 2021, Sánchez-Sáez et al. 2021), Conditional Neural Process (Čvorović-Hajdinjak et al. 2022) and Latent Stochastic Differential Equations (SDEs) (Fagin et al. 2024). Modeling of quasar light curves could be of great importance for further analysis (Jankov et al. 2022) and studying their structure. Deep learning of quasar light curves on a massive scale (Kovačević et al. 2023) is still improving. Quasar light curves could vary significantly, and their clustering before processing via deep-learning networks, might be beneficial to the performance of the deep-learning quasar light curves process and results (Kovačević et al. 2023). Namely, through its training process, SOM groups similar light curves together. Light curves with similar intrinsic variability patterns will be located close to each other on the map, while those that differ significantly (potentially due to observational noise or other factors) will be placed further apart. Grouping light curves of similar patterns together makes the subsequent modeling or analysis with Neural Processes or other methods more effective by focusing on the similar variability patterns of the quasars rather than being misled or obscured by noise.

In this paper, we investigate the potential enhancement of the Conditional Neural Process framework, specifically designed to model quasar light curves in Python (QNPy; Čvorović-Hajdinjak et al. 2022, Pavlović et al. 2024). We propose an augmentation of this framework by incorporating a preprocessing step that employs clustering through the Self-Organizing Map (SOM) algorithm. The clustering implementation has been executed in Python via the MiniSom library[1]. We aim to show that the SOM-based clustering can improve the QNPy models of variability of the SOM-clustered quasar light curves.

The structure of this paper is organized as follows. The 'Data' section details the quasar light curves datasets and preprocessing steps. In the 'Methods' Section, we explain the algorithmic approach and implementation specifics of the Conditional Neural Process with Self-Organizing Map clustering. The 'Results' and 'Conclusion' Sections present our findings, interpret the modeling outcomes, and explore their implications. Finally, the 'Future Work' Section concludes the paper with a recapitulation of the main contributions and a prospective outlook on potential research directions.

2. DATA

The SOM method was trained and tested with quasar light curves, detected by the SWIFT Burst Alert Telescope (9-Month BAT Survey, (Tueller et al. 2008)). Their optical light curves were taken from the ASAS-SN (All-Sky Automated Survey for Supernovae) database (Holoien et al. 2019), despite of its inherent uncertainties, due to its wide coverage and large dataset. Data points are the measured magnitude of selected objects in the g-band, in time (modified Julian date). The data set consists of 139 light curves that have between 100 and 600 data points (Fig. 1) and a cover range of up to 2000 days.

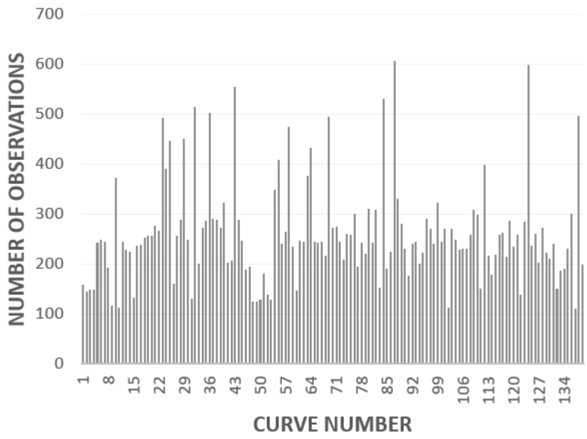

Fig. 1: Number of observations in quasar light curves collected in the first nine months of the all-sky survey by the Swift Burst Alert Telescope.

Optical light curves taken from the ASAS-SN database (Shappee et al. 2014, Kochanek et al. 2017) have a good representation of 80 percent sky coverage, and show particularities, such as flares, possible quasi-periodic oscillations, time gaps, and irregular cadence, enabling us to test our clustering with challenging data sets. Testing QNPy on light curves with large uncertainties helps to assess the robustness and adaptability of the QNPy model. The observational errors have been incorporated as a part of the data augmentation strategy to increase the diversity of the training dataset, reduce the risk of overfitting, and enhance the model's ability to generalize to unseen data. Curves from this data set have been used for training and testing the Conditional Neural Process (Čvorović-Hajdinjak et al. 2022), which was upgraded (QNPy) for modeling the mass amount of curves simultaneously. The need for such a tool comes from an immense amount of data that will be gathered by the next generation time domain surveys, such as Vera Rubin Observatory Legacy Survey of Space and Time (LSST) (LSST Science Collaboration et al. 2009, Ivezić et al. 2019). This survey will provide observations with different cadences over ten years for millions of AGN sources (Brandt et al. 2018, Bianco et al. 2022) in six filters - ugrizy. Such a significant amount of data will require efficient tools for processing. Given that QNPy has already been trained and tested on SWIFT/BAT light curves, possible benefits of SOM clustering on the processing efficiency of QNPy, would be most efficiently examined on the same data set.

The SOM method can handle various data types regardless of their structure and homogeneity, provided each input has a consistent tensor shape (e.g. the same length or number of features). Ensuring

[1]More information on Minimalistic implementation of Self Organizing Maps, along with project description and GitHub page could be found at: https://pypi.org/project/MiniSom/

all inputs have uniform dimensions without missing values is crucial for the SOM processing (Kohonen 1990). Uniformity of input data is typically achieved through padding or feature extraction (Rejeb et al. 2022). These techniques ensure that SOM can process the data, even if the underlying data sources are irregularly sampled or have varying lengths.

Since SWIFT/BAT light curves[2] vary in the number of observations (points), each light curve has been padded by adding the mean magnitude value of that curve at the end so that all curves have the same number of points as the light curve with the maximum number of points. Additionally, the curves have been scaled to improve the clustering performance. During the testing phase, it was concluded that the best scaler for the SWIFT/BAT data was MinMax, which normalized the curves to the interval [0,1].

3. METHOD

Neural networks consist of a collection of connected units or nodes/neurons, that loosely model the neurons in a biological brain. Neurons are grouped to form layers (Liu 2023). Clustering is a type of unsupervised learning, which is used to discover patterns and relationships within data. There are many clustering algorithms: K-Means (Ordovás-Pascual and Sánchez Almeida 2014), Gaussian mixtures (Tóth et al. 2019), Spectral clustering (Ng et al. 2001), Hierarchical clustering (Yu and Hou 2022), etc. Among the many machine learning algorithms, Self-Organizing Map (SOM) stands out as a powerful tool for data visualization, clustering, and dimension reduction (Fig. 2). SOM has been used in astronomy mostly for mapping the empirical relation of galaxy color to red-shift (Buchs et al. 2019), and visualization and computation of the estimation of galaxy physical properties (Hemmati et al. 2019).

The application of SOMs for clustering quasar light curves presents an advantageous approach due to several characteristics of SOMs. Firstly, SOMs are good at reducing the dimensionality of complex data while preserving topological and metric relationships, making them ideal for handling the high-dimensional nature of quasar light curves. This is particularly beneficial for identifying underlying patterns and structures within the light curves, which may not be readily apparent. SOMs can capture non-linear and non-stationary variability in quasar light curves by clustering similar light curves based on their intrinsic properties, such as variability amplitude, and time scales of variability. By organizing the light curves into clusters, SOM facilitates a more targeted modeling approach where each cluster represents a different archetype of variability. Modeling each cluster separately allows for the construction of more specialized and accurate models that can account for the specific characteristics and noise properties of each group. This specificity can lead to more robust predictions and insights into the physical processes governing quasar variability. Moreover, it can improve the efficiency of the modeling process by focusing computational resources on distinct subsets of the data, each with its unique features, rather than applying a one-size-fits-all model to the entire dataset.

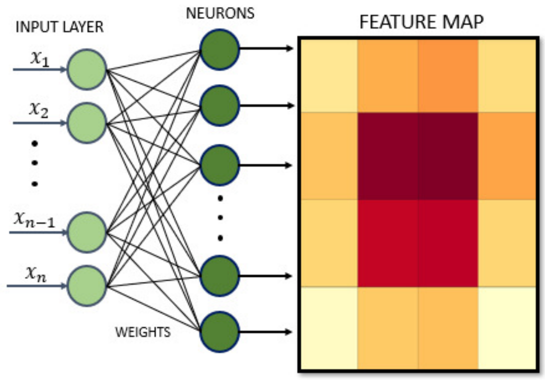

Fig. 2: Self-organizing maps are a type of neural network that is trained using unsupervised learning to produce a low-dimensional representation of the input space of the training samples, called a feature map. The feature map in this figure is an output of SOM clustering of SWIFT/BAT data. Values $x_1, x_2, ..., x_n$ are vectors (in our case - light curves) that are passed to the input layer.

The SOM, also known as the Kohonen network (Kohonen 1990) is based on competitive learning. The neurons (or nodes) compete to decide which one will be activated over a set of inputs and this neuron is called the winner or the best matching unit (BMU).

Each neuron is connected to a weight vector that has the same dimensions as the input data. At the beginning of the SOM process, the weights are initialized with random values. During forward propagation, input values are compared with nodes (Fig. 3) using the "neighborhood function", which calculates the weights of the neighborhood of a position in the map. The Minisom has several options for neighborhood function, but for the SWIFT/BAT data Gaussian function (with parameter sigma determining the influence on neighboring nodes of SOM, set to 2.0), showed the best results. During training progress, the algorithm calculates the Euclidean distance between every weight and the current training item:

$$d(x, w_j) = \sqrt{\sum_{i=0}^{p-1} (x_i - w_{ij})^2} \tag{1}$$

Vector $x = (x_0, \ldots, x_{p-1})$ presents the input values and w_{ij} weight for node j. Distance determines the difference between two vectors of the same size.

[2]More information on light curve sources can be found on: https://swift.gsfc.nasa.gov/results/bs9mon/

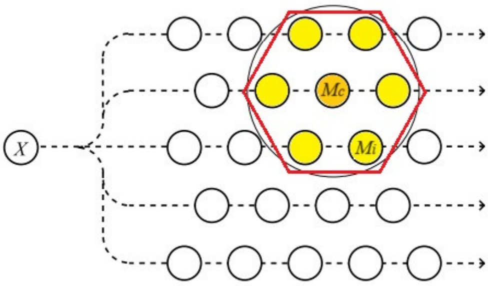

Fig. 3: The figure from (Kohonen 1990) is adapted for our SOM implementation. In the SOM algorithm, the input data item X is compared to a set of nodes M_i. The node M_c, which matches the most X is the winning one. Neighboring nodes are more similar to X than nodes outside the circled neighborhood. To determine the winning node, we have used a Gaussian neighborhood function in rectangular topology.

The least distance is BMU. The weight vectors are updated by the algorithm to adapt to the distribution of the data by:

$$w_{ij}(t+1) = w_{ij}(t) + \theta_j(t)\alpha(t)(x_i(t) - w_{ij}(t)), \quad (2)$$

where $i = 0, .., p-1$, t shows iteration, $\theta_j(t)$ is the neighbourhood function and $0 < \alpha(t) < 1$ that decreases monotonically in time is the learning rate. The neighborhood function determines to which extent each output node receives a training adjustment from the current training pattern. The Gaussian function is a common choice for a neighborhood function:

$$f(x, \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{x-\mu}{\sigma})^2}, \quad (3)$$

where x represents input, while μ and σ are the mean and the standard deviation (respectively).

The Minisom method also gives options to choose a topology for clustering, between rectangular and hexagonal, as well as different scalers ('minmax', 'standard', 'robust'), learning rate, etc. After careful testing with various sets of parameters and analyzing the results, it has been concluded that for the SWIFT/BAT light curves the best set of parameters is:

(i) scaler = 'minmax' - normalization function, used in preprocessing phase
(ii) neighborhood function = 'gaussian' - neighborhood function, used for updating weights
(iii) epochs = 50000 - number of epochs to train SOM
(iv) topology = 'rectangular' - topology of SOM
(v) learning rate = 0.01
(vi) sigma = 2.0 - influence on neighboring nodes of SOM

Minmax scaler is developed in sklearn.preprocessing.MinMaxScaler[3] and transforms input to range [0,1]. Transformations are given by the equation:

$$x_{\text{std}} = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \quad (4)$$

$$x_{\text{scaled}} = x_{\text{std}}(\max - \min) + \min \quad (5)$$

where min and max determine the range in which we want to scale our features (in our case min = 0 and max = 1). Clustering of this dataset wasn't computationally demanding. The whole training/clustering process lasted ≈ 5 minutes.

During testing with various sets of parameters it has been noted that aside from the scaler used for reshaping (preprocessing), topology defining the shape of clusters, and sigma (influence on neighboring nodes of SOM), the SOM is sensitive to outliers in light curves. Small sets of badly classified curves led to the significantly diminished performance of QNPy. After considering this, additional preprocessing has been done by removing outliers with the Z-Score function:

$$z = |\frac{m - \mu}{\sigma}| \quad (6)$$

where m is the magnitude of observation, μ is the mean value of magnitudes and σ is the standard deviation. Every point (observation) that had the z value below 2.0 was considered an outlier.

The number of clusters is also a hyperparameter in the SOM method (Kohonen 2013, 2014). The formula for calculating clusters was fine-tuned through many experiments on different data sets to get an equation suitable for various light curve data. It was important to avoid getting a large number of small clusters, containing very similar curves because that could lead to overfitting during the QNPy processing. On the other hand, a very small number of generalized clusters would not improve the QNPy performance. The length of the input data determines the number of clusters. More specifically, the padded curve length specifies the number of grids that create a cluster map. This value is calculated with the following equation:

$$g = R_u(\sqrt{\sqrt{n}}) \quad (7)$$

where g is the number of grids that create a cluster map, $R_u(x)$ is a function that returns the smallest integer greater than or equal to x, and n is the length of padded curves.

[3]More information on MinMaxScaler can be found on: https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html

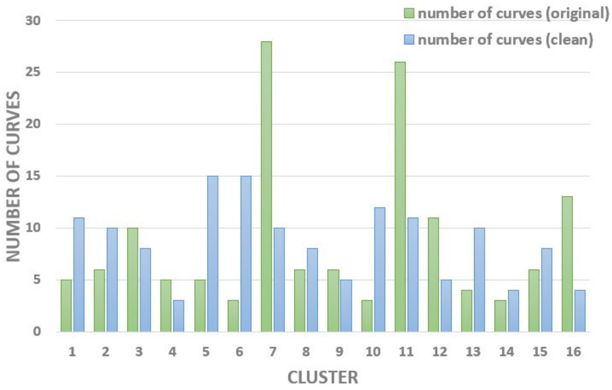

Fig. 4: Number of curves in each cluster, before (green) and after (blue) removing outliers from the SWIFT/BAT quasar light curves.

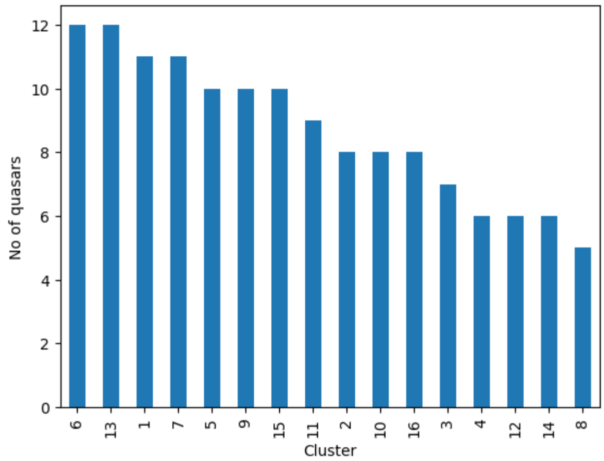

Fig. 5: Number of curves in each cluster, after the SOM processing of clean (padded, normalized, removed outliers) SWIFT/BAT quasar light curve data.

4. RESULTS

To determine whether the clustering of our set of light curves will improve their modeling via the QNPy neural network, the following steps have been taken:

(1) the whole data set, containing all 139 light curves has been modeled via QNPy, and the results were saved for letter analysis,
(2) the whole data set, containing all 139 curves has been processed via the SOM algorithm. This process included: padding the light curves to maximum length, reshaping them to the interval [0,1] via MinMax scaler, and clustering,
(3) analyzing results and changing parameters, until achieving the best possible results,
(4) selecting several clusters for further analysis,
(5) taking original curves (as they were before padding and reshaping) from each of those clusters and running QNPy only on those curves,
(6) comparing results of modeling for curves before and after clustering.

Removing outliers, finding the optimal set of parameters, and careful preprocessing resulted in a more even distribution of light curves in clusters (Fig. 4), as well as fewer misclassified curves. The SOM method divided the SWIFT/BAT set into 16 clusters (Fig. 5).

Visualization of curves in clusters (Fig. 6) shows the complexity and how the SOM algorithm managed to cluster them according to similarities (gradient changes) in their structure. We can see that in some clusters (3, 4, 8, 9, 11, 12, 14) padding is taking a big part of reshaped curves. Although 16 distinct clusters were initially identified (Fig. 6), further analysis demonstrated that these clusters could be aggregated into 4 general groups without significant loss of detail, based on high intra-group similarity and low inter-group similarity:

(1) Clusters with Low Variability (C-LV): Clusters 4, 10, and 14 exhibit tightly grouped light curves with minimal spread among individual observations, indicative of homogeneous behavior and suggesting lower intrinsic variability or observational noise.
(2) Clusters with Moderate Variability (C-MV): Clusters 2, 5, 6, 9, 12, and 13 demonstrate a moderate level of variability. Despite some spread, individual light curves within these clusters maintain a coherent pattern, categorizing them into an intermediate variability level.
(3) Clusters with High Variability (C-HV): Marked by significant divergence from the mean, Clusters 1, 3, 7, 11, and 15 display pronounced variability. This suggests a high degree of intrinsic quasar variability or substantial observational noise.
(4) Unique Clusters (C-U): Cluster 8 presents a notable trend distinct from other clusters, potentially indicating a different variability type or systematic data effect, thus categorizing it as unique. The 'S' shaped pattern observed in Cluster 8 might indicate quasi-periodic oscillations (QPOs). This oscillatory behavior suggests an underlying periodic process that could be associated with accretion disk dynamics or interactions with the central supermassive black hole's magnetic field, meriting detailed analysis in a separate paper.

Moreover, Cluster 16 exhibits a flare-like feature within its mean curve, characterized by a sharp increase in brightness followed by a gradual decrease. This pattern might suggest transient brightening events, which are significant for understanding the dynamics and physical processes within quasars. The presence of such a flare-like appearance warrants further investigation to discern the nature of these transient events and their astrophysical implications. Additionally, some clusters consist of curves that were padded with very different numbers of points (2, 5, 6). This could have influenced the clustering of padded curves. To determine whether clustering would improve the QNPy modeling of the SWIFT/BAT data set, the following clusters were selected based on several reasons (Table 1), which should cover different types and give an overview of the pros and cons of the SWIFT/BAT data set SOM clustering.

Cluster 4 is an example of a cluster with Low Variability (C-LV). This cluster consists of 6 curves with very similar structures and distinctive slopes. Fig. 7 shows the difference in modeling when 3C273 is modeled as part of the whole data set (left) and as a part of Cluster 4 (right). Loss function and mean squared error (MSE) are much smaller when QNPy processes only Cluster 4. Both modeled curves (blue lines) follow the main gradient change, but after clustering the model goes nicely through the mean of data points on the lower and upper part of the curve.

Cluster 13 is an example of a cluster with Moderate Variability (C-MV). This cluster consists of 12 curves with very similar structures and distinctive gradient changes. Fig. 8 shows the difference in modeling when Mrk6 is modeled as part of the whole data set (left) and as a part of Cluster 13 (right). Loss function and mean squared error (MSE) are smaller when QNPy processes only Cluster 16. Both modeled curves (blue lines) follow gradient changes, but after clustering the model fits data better through the whole length. While the first modeled curve manages to model the gradient change in the middle, it stays near the mean. The second curve nicely follows the whole pattern of the curve, but still does not reach the top of the first maximum and the minimal values in the middle.

Cluster 1 is an example of a cluster with High Variability (C-HV). This cluster consists of 11 curves with very variable structures. Fig. 9 shows the difference in modeling when IC4329A is modeled as part of the whole data set (left) and as a part of Cluster 1 (right). Loss function and mean squared error (MSE) are larger when QNPy processes only Cluster 1. Both modeled curves (blue lines) follow the mean of the main gradient change, but neither of them captures details of the light curve. While the first modeled curve shows small gradient changes in more dense regions, the second one is completely smooth. This indicates that for clusters that contain highly variable curves, modeling results are close to the modeling of the whole set, which is still impressive given the significantly smaller number of curves for learning, making it much more efficient (considering the time it takes to model this cluster is several minutes, whereas for the whole set on the same machine takes a couple of hours).

Cluster 8 is an example of a Unique Cluster (C-U). This cluster consists of 5 curves with very distinctive structures. Fig. 10 shows the difference in modeling when IRAS09149m6206 is modeled as part of the whole data set (left) and as a part of Cluster 8 (right). Loss function and mean squared error (MSE) are higher when QNPy processes only Cluster 8. Both modeled curves follow the main gradient of the light curve, but the first case (modeling of the whole set) has more details and small gradient changes in the more dense regions. The second graph (modeling curves from Cluster 8) is a smooth representation of the main data trend. This indicates that for clusters that contain unique curves (some possibly misclassified), especially if they consist of very few examples (6 curves), modeling results are less accurate than those of modeling the whole set.

Cluster 16 consists of 8 curves and a very peculiar mean model. After further analysis of light curves that were classified in this cluster, it was apparent that some of them were misclassified. The most obvious example was the light curve 3C382 (Fig. 11), which should have been classified in Cluster 4. After running QNPy on Cluster 16, the results varied from model to model. Fig. 11 shows that modeling of the whole data set gives better results if clustering is not accurate.

The first model follows the main gradient change and overall data structure while the second one stays closer to the mean value following the gradient changes of the first part of the curve. It is also visible that loss and MSE could have lower values for a less accurate model. This could be caused by the fact that testing and validation sets are very small for clustered curves and there is a larger probability that they will randomly get the curves that fit the model better. This indicates that there is still a need for further improvement of the clustering algorithm.

5. CONCLUSION

Quasars are one of the most powerful sources in the Universe. Their radiation covers the whole electromagnetic spectrum in a very complex manner. The gathering and analysis of quasar light curves have been revealing the structure of these complex objects and continue to do so with the development of more powerful observing technologies. We need to perfect our programming tools to keep up with the immense amount of data that the most powerful telescopes can now harbor. Conditional Neural Process, which has shown to be a good modeling tool for capturing variability and complexity of quasar light curves, still struggles with modeling a large number of these complex data sets. To improve its performance, we examined the SOM clustering as a means of preprocessing and selecting batches of quasar light curves, which would give better and faster results.

SWIFT/BAT data was selected since it gives a good starting point for determining the applicability of SOM clustering on complex, stochastic, irregular, and very different sources. This data set consists of 139 very complex, variable, inhomogeneous light curves with many gaps and different structures and

Table 1: Clusters selected for analysis of the SOM clustering effect on QNPy modeling performance.

Cluster number	Number of curves	Selection reason
4 (C-LV type)	6	a small number of curves and a peculiar structure with one big gradient change
13 (C-MV type)	12	the largest number of variable curves, the mean model has a visible structure
1 (C-HV type)	11	highly variable curves, with different degrees of padding
8 (C-U type)	11	distinct from other clusters potentially indicating different variability types
16 (specific)	8	the mean model has a very peculiar flare-like feature within its mean curve

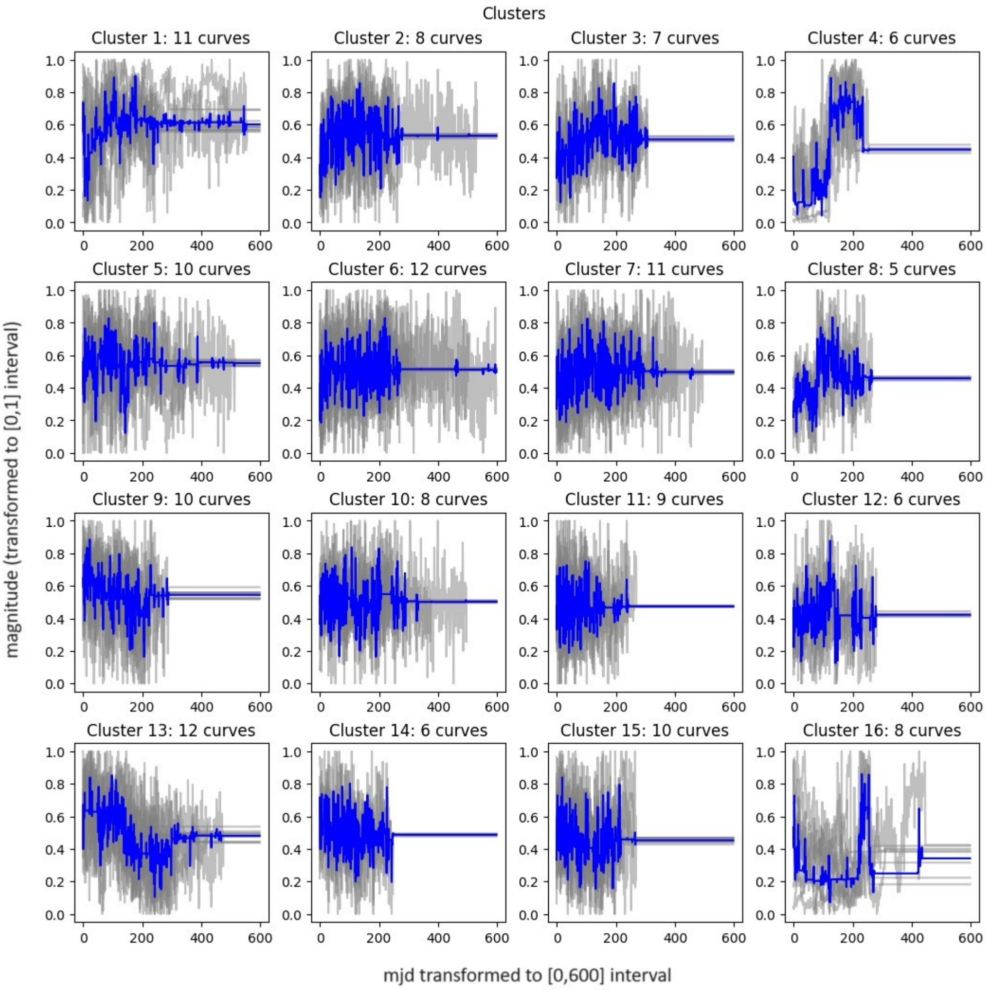

Fig. 6: Visualisation of each of 16 clusters, with average curve structure in blue. The SOM method requires all curves to be the same length (length of maximum curve). Curves are padded with mean magnitude value. After padding, curves are mapped to [0,1]x[0,600] intervals and clustered.

lengths. The same data set was used for training and testing the Conditional Neural Process on which QNPy has been built. Testing Neural Processes on light curves with large uncertainties helps assess the robustness and adaptability of the QNPy model.

The SOM algorithm was selected for this task since it is not a very complex machine-learning algorithm for unsupervised learning and it works fast, which is essential for preprocessing purposes. It is trained through a competitive neural network (neurons in the output layer compete among themselves to be activated). The competitive process is to search for the most similar neuron (BMU) with the input pattern. The BMU and its neighbors adjust their weights. We have used the MiniSom implementation of the Self Organizing Maps (SOM) and attempted to

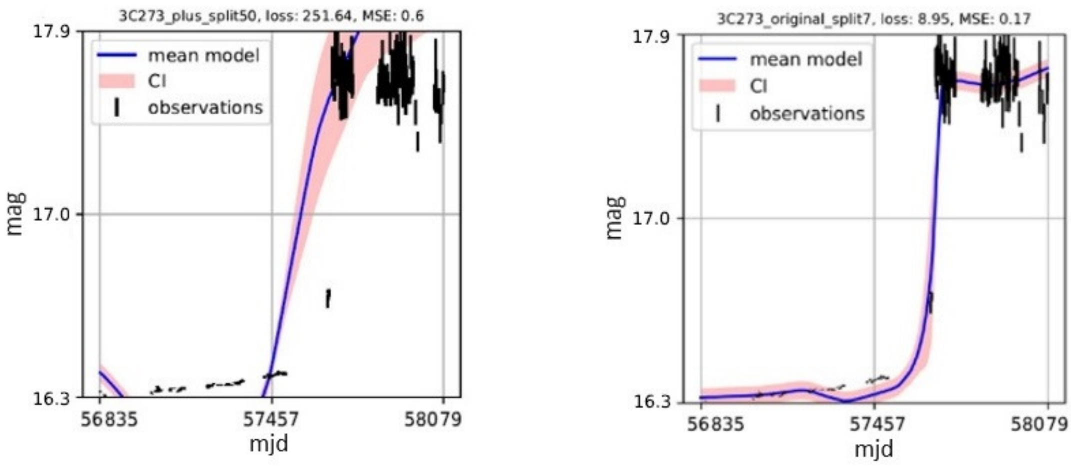

Fig. 7: The QNPy modeling of the 3C273 light curve using 3000 iterations. Data is transformed to the [-2,2][-2,2] interval during the QNPy preprocessing phase. The light red area is the one-sigma confidence interval. The left graph shows the results of the modeling of 3C273 during the processing of the whole data set, while the right side graph shows the modeling of 3C273 during the processing of Cluster 4. Cluster 4 is an example of a cluster with low variability (C-LV). The mean model is shown as a blue line.

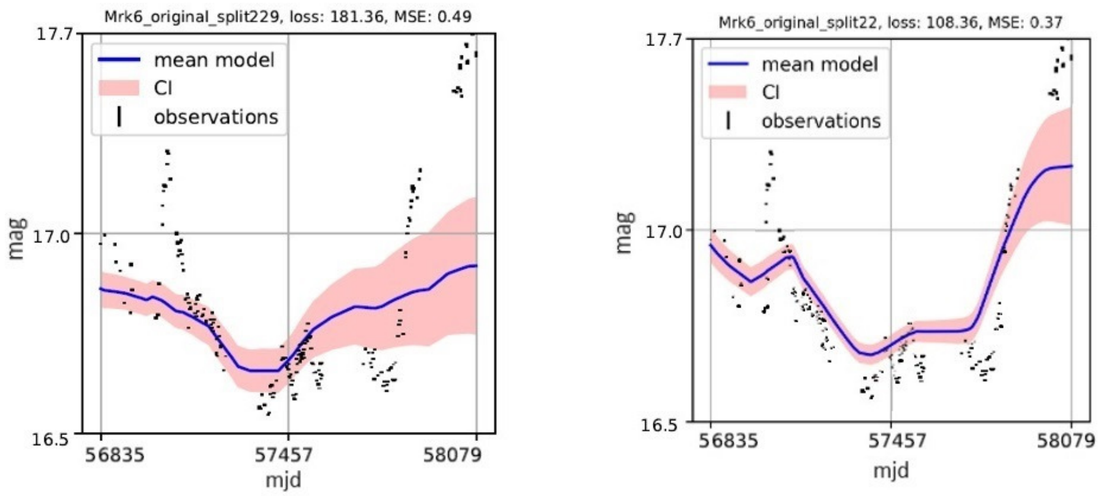

Fig. 8: The QNPy modeling of the Mrk6 light curve using 3000 iterations. Data is transformed to the [-2,2][-2,2] interval during the QNPy preprocessing phase. The light red area is the one-sigma confidence interval. The left graph shows the results of the modeling of Mrk6 during the processing of the whole data set, while the right side graph shows the modeling of Mrk6 during the processing of Cluster 13. Cluster 13 is an example of a cluster with moderate variability (C-MV). The mean model is shown as a blue line.

make the best clustering of the SWIFT/BAT curves. Curves from several selected clusters were modeled via QNPy and results were compared with the original clustering of the whole set.

It was noted that the SOM parameters (neighborhood function, topology, scaler, etc) contribute significantly to the clustering process and results. Additionally, the SOM method required the reshaping of light curves via normalization (minmax) and transformation to the same interval. Removing outliers in light curves is beneficial for better classification and a more even distribution of light curves among clusters.

The results indicate that curves from clusters whose mean value has distinctive features, with low and medium variability (C-LV, C-MV), show more improvement in modeling via QNPy than those with highly variable mean functions (C-HV) or unique structures(C-U), especially if such cluster contains a small number of curves. Comparing the results of modeled light curves before and after clustering, showed that good clustering could contribute to greater extant performance of the QNPy modeling

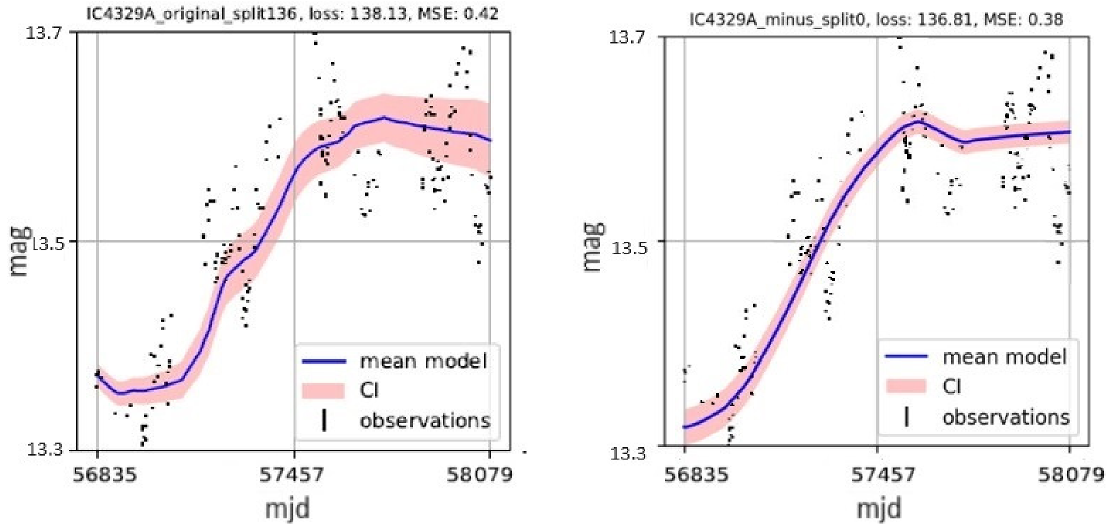

Fig. 9: The QNPy modeling of IC4329A light curve using 3000 iterations. Data is transformed to the [-2,2][-2,2] interval during the QNPy preprocessing phase. The light red area is the one-sigma confidence interval. The left graph shows the results of the modeling of IC4329A during the processing of the whole data set, while the right side graph shows the modeling of IC4329A during the processing of Cluster 1. Cluster 1 is an example of a cluster with moderate variability (C-HV). The mean model is shown as a blue line.

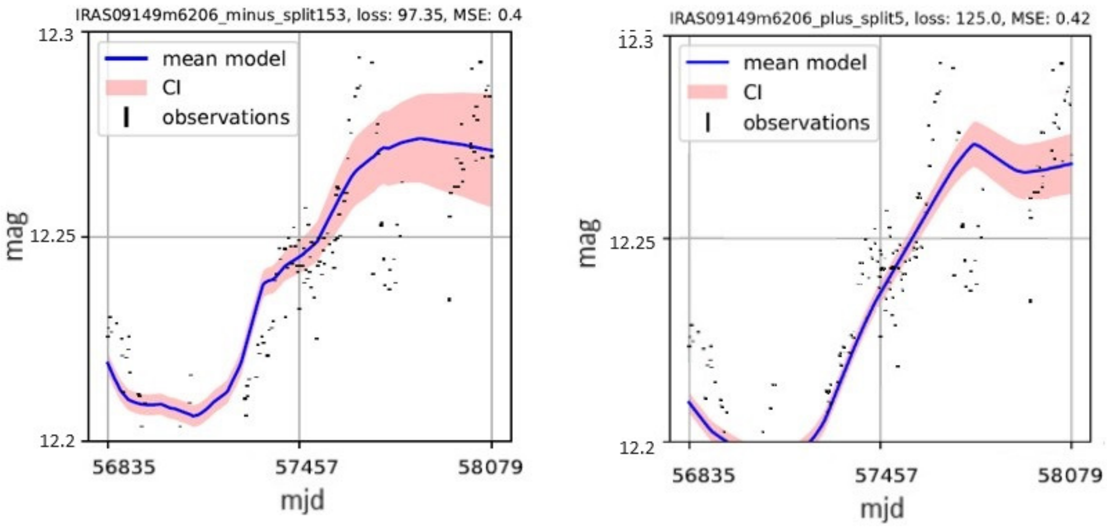

Fig. 10: The QNPy modeling of IRAS09149m6206 light curve using 3000 iterations. Data is transformed to [-2,2][-2,2] interval during the QNPy preprocessing phase. The light red area is the one-sigma confidence interval. The left graph shows the results of the modeling of IRAS09149m6206 during the processing of the whole data set, while the right side graph shows the modeling of IRAS09149m6206 during the processing of Cluster 8. Cluster 8 is an example of a Unique Cluster (C-U). The mean model is shown as a blue line.

process. However, the SOM has shown to be very sensitive to some data features which could lead to wrongly clustered curves. In the case of wrongly clustered light curves, the QNPy modeling shows far less reliable results than the modeling of the full set of data. Not only because there is less data to "learn" from, but because most data have certain features that are not present in misclassified light curves.

The clustering of quasar light curves as a means of preprocessing could significantly improve the modeling via QNPy, but clustering itself could be improved to achieve the most optimal results. The SOM could be used for this task, but it needs further analysis and testing on various datasets, with careful tuning of its parameters and good preparation of input data.

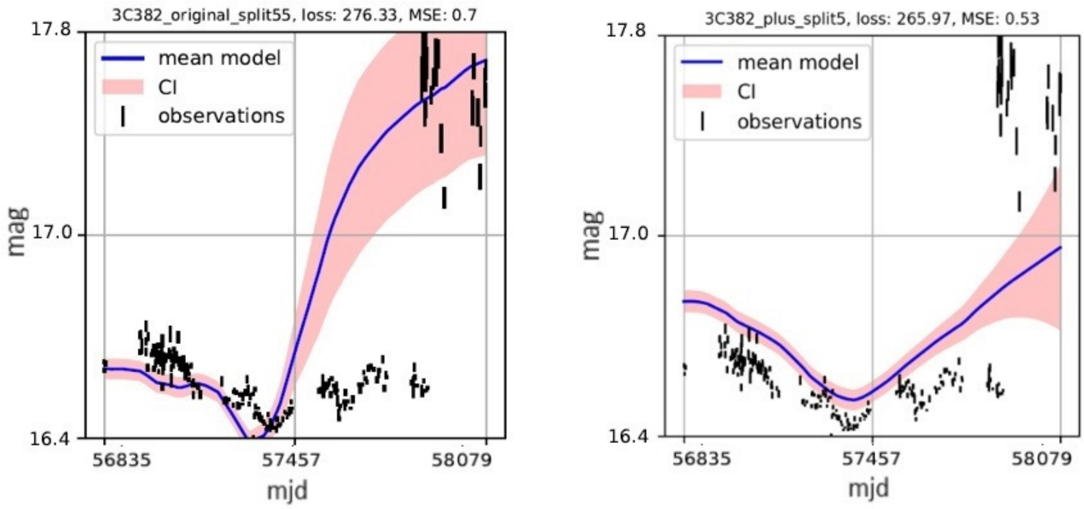

Fig. 11: The QNPy modeling of 3C382 light curve using 3000 iterations. Data is transformed to the [-2,2][-2,2] interval during the QNPy preprocessing phase. The light red area is the one-sigma confidence interval. The left graph shows the results of the modeling of 3C382 during the processing of the whole data set, while the right side graph shows the modeling of 3C382 during the processing of Cluster 16. The mean model is shown as a blue line.

6. FUTURE WORK

The SOM algorithm could be tested on a set of light curves that have a similar number of data points in the same period so that reshaping and padding don't change the structure of light curves before clustering. The ZTF (Zwicky Transient Facility) data would be a good choice for this analysis since they have fewer uncertainties, which makes them a good sample for further exploring the effectiveness of QNPy and creating a comprehensive analysis, offering deeper insights into light curve behaviors.

Padding could be done via interpolation, reflective padding, or replication. The selection of the number of clusters could be explored further. Clustering could be done via other methods to determine the one that would make the best clustering results and hence contribute more to the improvement of the QNPy process.

Acknowledgements – This paper is greatly improved by the contribution, guidance, and consultation from Dr. A. Kovačević and Dr. M. Pavlović, along with the highly appreciated comments from referees. The author acknowledges the funding provided by the University of Belgrade - Faculty of Mathematics (the contract 451-03-66/2024-03/200104) through the grants by the Ministry of Science, Technological Development and Innovation of the Republic of Serbia.

REFERENCES

Antonucci, R. 1993, ARA&A, 31, 473
Bank, D., Koenigstein, N. and Giryes, R. 2021, arXiv:2003.05991
Bianco, F. B., Ivezić, Ž., Jones, R. L., et al. 2022, ApJS, 258, 1
Bonfield, D. G., Sun, Y., Davey, N., et al. 2010, MNRAS, 405, 987
Brandt, W. N., Ni, Q., Yang, G., et al. 2018, arXiv:1811.06542
Buchs, R., Davis, C., Gruen, D., et al. 2019, MNRAS, 489, 820
Čvorović-Hajdinjak, I., Kovačević, A. B., Ilić, D., et al. 2022, AN, 343, e210103
Fagin, J., Park, J. W., Best, H., et al. 2024, ApJ, 965, 104
Hemmati, S., Capak, P., Masters, D., et al. 2019, ApJ, 877, 117
Holoien, T. W. -S., Brown, J. S., Vallely, P. J., et al. 2019, MNRAS, 484, 1899
Ivezić, Ž., Kahn, S. M., Tyson, J. A., et al. 2019, ApJ, 873, 111
Jankov, I., Kovačević, A. B., Ilić, D., et al. 2022, AN, 343, e210090
Kelly, B. C., Bechtold, J. and Siemiginowska, A. 2009, ApJ, 698, 895
Kochanek, C. S., Shappee, B. J., Stanek, K. Z., et al. 2017, PASP, 129, 104502
Kohonen, T. 1990, Proceedings of the IEEE, 78, 1464
Kohonen, T. 2013, Neural Networks, 37, 52
Kohonen, T. 2014, MATLAB Implementations and Applications of the Self-Organizing Map (Unigrafia Oy)
Kovačević, A. B., Popović, L. Č., Simić, S. and Ilić, D. 2019, ApJ, 871, 32
Kovačević, A. B., Ilić, D., Popović, L. Č., et al. 2023, Univ, 9, 287
Kozłowski, S. 2017, ApJ, 835, 250
LaMassa, S. M., Cales, S., Moran, E. C., et al. 2015, ApJ, 800, 144
Liu, G. 2023, arXiv:2212.03853

LSST Science Collaboration, Abell, P. A., Allison, J., et al. 2009, arXiv0912.0201
Ng, A., Jordan, M. and Weiss, Y. 2001, in Advances in Neural Information Processing Systems, Vol. 14 (Cambridge: MIT Press)
Ordovás-Pascual, I. and Sánchez Almeida, J. 2014, A&A, 565, A53
Pavlović, M., Raju, A., Kovačević, A., et al. 2024, Journal of Open Source Software, in preparation
Rejeb, S., Duveau, C. and Rebafka, T. 2022, Chemometrics and Intelligent Laboratory Systems, 231, 104653
Richards, G. T., Lacy, M., Storrie-Lombardi, L. J., et al. 2006, ApJS, 166, 470
Sánchez-Sáez, P., Lira, P., Mejía-Restrepo, J., et al. 2018, ApJ, 864, 87
Sánchez-Sáez, P., Lira, H., Martí, L., et al. 2021, AJ, 162, 206
Shang, Z., Brotherton, M. S., Wills, B. J., et al. 2011, ApJS, 196, 2
Shapovalova, A. I., Popović, L. Č., Chavushyan, V. H., et al. 2017, MNRAS, 466, 4759
Shapovalova, A. I., Popović, L. Č., Afanasiev, V. L., et al. 2019, MNRAS, 485, 4790
Shappee, B. J., Prieto, J. L., Grupe, D., et al. 2014, ApJ, 788, 48
Tachibana, Y., Graham, M. J., Kawai, N., et al. 2020, ApJ, 903, 54
Tóth, B. G., Rácz, I. I. and Horváth, I. 2019, MNRAS, 486, 4823
Tueller, J., Mushotzky, R. F., Barthelmy, S., et al. 2008, ApJ, 681, 113
Ulrich, M.-H., Maraschi, L. and Urry, C. M. 1997, ARA&A, 35, 445
Urry, C. M. and Padovani, P. 1995, PASP, 107, 803
Wagner, S. J. and Witzel, A. 1995, ARA&A, 33, 163
Yu, H. and Hou, X. 2022, A&C, 41, 100662
Zhang, X., Yang, F., Guo, Y., et al. 2023, Mathematics, 11, 330

МОДЕЛОВАЊЕ ПРОМЕНЉИВОСТИ КВАЗАРА УПОТРЕБОМ НЕУРОНСКИХ ПРОЦЕСА БАЗИРАНИХ НА САМООРГАНИЗУЈУЋИМ МАПАМА

И. Чворовић - Хајдињак

Катедра за астрономију, Математички факултет, Универзитет у Београду
Студентски трг 16, 11000 Београд, Србија

E–mail: *iva.cvorovic@gmail.com*

УДК 524.7
Оригинални научни рад

Условни неуронски процеси су се показали као моћни алат за моделовање кривих сјаја квазара. Имајући у виду сложену природу ових објеката, а самим тим и података који нам од њих стижу, моделовање кривих сјаја може бити временски веома захтевно. Резултати моделовања некада нису довољно прецизни да би се на њима вршила даља анализа. У циљу унапређења моделовања кривих сјаја, у овом раду је испитан ефекат груписања великог броја кривих сјаја квазара коришћењем Самоорганизујућих мапа (СОМ), у процесу који би претходио неуронском моделовању. Резултати примене СОМ методе на скупу кривих сјаја квазара, прикупљених у програму ASAS-SN, и њихово касније моделовање, показују да условни неуронски процеси раде боље након претходне обраде података СОМ методом. Закључак овог рада је да груписање кривих сјаја квазара СОМ методом, пре моделовања условним неуронским процесима, унапређује резултате моделовања.