UNIVERZITET U BEOGRADU
MATEMATIČKI FAKULTET

Aman Nadimpalli Raju

APPLICATION OF ATTENTIVE AND LATENT NEURAL PROCESSES TO QUASAR LIGHT CURVES IN THE CONTEXT OF THE LSST

Master's Thesis

Beograd, 2024.

Mentor:

dr Andjelka Kovačević
Univerzitet u Beogradu, Matematički fakultet

Thesis Comittee:

dr Dragana Ilić
Univerzitet u Beogradu, Matematički fakultet

dr Francesco Tombesi
University of Rome Tor Vergata

dr Luka Popović
Astronomical Observatory Belgrade

dr Eric Slezak
Observatoire de la Côte d'Azur

Date of Defense: 13.09.2024

"No man is an island,
Entire of itself;
Every man is a piece of the continent,
A part of the main."
- John Donne

I dedicate this thesis to everyone who has helped me grow. My parents, my brother, all my friends, my mentors, and especially everyone who has kept me sane over the last six years regardless of distance.

Scientific Acknowledgments

I would like to express my deep gratitude to the scientific communities and facilities that have been instrumental in the success of this research. I acknowledge the LSST AGN Data Challenge created by Weixiang Yu, Gordon Richards, and their team for their invaluable contribution. The SDSS survey and ZTF survey, have provided essential data that made this work possible. I also acknowledge the use of the SuperAST computational cluster of the University of Belgrade's Astronomy Department under the Faculty of Mathematics.

I am immensely grateful to my primary mentor, Anđelka Kovačević, for her consistent guidance and support, always steering me in the right direction. I extend my thanks to Marina Pavlović for her collaboration and assistance in the development of QNPy, and to Iva Ćvorović-Hajdinjak for her foundational work and help with the Neural Processes applied to quasar light curves. I appreciate Dragana Ilić's helpful feedback and Saša Simić's assistance with computation.

I would also like to thank my supervisors, Luka Popović, Francesco Tombesi, and Eric Slezak, for their valuable guidance and feedback throughout this project. Special thanks to Nicolás Guerra-Varas for his assistance in building and testing the transformer model, as well as to Marco Immanuel Rivera and Aurellio Deandra for creating and providing the datasets used in training the model.

Finally, I am grateful to everyone whose feedback has significantly enhanced the quality of this research. In particular, the constructive input from Maurizio Paolillo, Demetra De Cicco, Vincenzo Petrecca, Đorđe Savić, and Paula Sánchez-Sáez has greatly influenced the direction of the model. Thank you all for your invaluable contributions.

Master Thesis Title: Application of Attentive and Latent Neural Processes to Quasar Light Curves in the Context of the LSST

Abstract: With the advent of the Legacy Survey of Space and Time (LSST), traditional methods of modeling quasar light curves need to be replaced with data driven and non-parametric models. Neural Processes can learn to generate reliable representations of large datasets of stochastic processes from relatively few context points. In this thesis, we utilize Neural Processes upgraded with attentive mecha nisms, a latent space, and stratification via Self-Organizing Maps to model quasar light curves from prior large optical/UV surveys. We utilize simulated light curves to probe the hidden representations of the light curves to unlock information about key parameters driving quasar variability. Furthermore, we utilize Transformer models to detect hidden supermassive binary black hole mergers within light curves. With all of our models working in tandem, we provide a comprehensive analysis package for quasar light curves obtained from large surveys.

Keywords: astronomy, computation, quasars, time-series

This Master thesis is submitted in partial fulfillment of the requirements for the degree "MASTER ASTROFIZICAR" as part of a multiple degree awarded in the framework of the Erasmus Mundus Joint Master in Astrophysics and Space Science – MASS jointly delivered by a Consortium of four Universities: Rome “Tor Vergata”, Belgrade, Bremen, and Côte d’Azur regulated by the MASS Consortium Agreement and funded by the EU under the call ERASMUS-EDU-2021-PEX-EMJM-MOB.

Parts of this thesis have been presented as:

- “A Deep Learning Approach for Understanding Quasar Light Curves in the Legacy Survey of Space and Time" - Symposium Mathematics and Applica tion, Faculty of Mathematics, University of Belgrade, 2023
- “QNPy and QhX Workshop” - LSST AGN Science Collaboration Quarterly Report, March 2024
- “Attentive Latent Neural Processes for modeling Quasar Variability in the LSST” - LSST TVS Colloquium, 21st May 2024
- “Pay Attention: Neural Processes with Latent Spaces and Attention to model Quasar Light Curves for the LSST” - XIII SAW, Astronomical Society “Ruđer Bošković", 18th May 2024
- “UPGRADING QNPY: MODELLING QUASAR LIGHT CURVES IN LARGE SURVEYS” - VI Conference on Active Galactic Nuclei and Gravitational Lens ing, Zlatibor Mt., Serbia, June 02-06, 2024,
- “Connecting the Dots: Attentive Latent Neural Processes” - 2nd MASS Sum mer School, Nice, France, July 2024 (First Prize 180s Thesis Talk)
- “LSST SER-SAG-S1: upgrade of QNPy package” - Catching supermassive black holes with Rubin-LSST: Towards novel insights and discoveries into AGN science, Torino, Italy, July 22nd-25th 2024

Contents

Chapter 1

Quasar variability

1.1 Introduction

Studying quasar variability offers a chance to understand the processes that drive some of the most energetic phenomena in our universe. With the advent of the Legacy Survey of Space and Time (LSST), led by the Vera C. Rubin Observatory, an unprecedented amount of quasars will be observed, providing the means to conduct the most in-depth variability studies of quasars.

Through the use of novel machine learning methods, secrets hidden within stochastic quasar signals can be unlocked, whether purely for interpolation, for a more in-depth understanding of the parameters associated with supermassive black holes, or for detection of rare events such as impending supermassive binary black hole mergers. This thesis proposes a method to tackle these challenges through the use of a sophisticated technique that combines stratification, attentive mechanisms, and neural processes to generate reliable and smooth light curve models of quasars from large datasets with relatively few context points in a non-parametric manner. We will also discuss models that can be used in tandem, namely transformer mod els that can be used to detect hidden signals within the light curves. We will be using many quasar light curves

obtained from large optical/UV surveys, along with simulated light curves.

The thesis is structured as follows. In Chapter 2, we discuss the theory behind quasar variability and the models that drive our understanding of it. In Chapter 3, we discuss methods that are currently used to model quasar variability and the benefits and costs associated with these various models. In Chapter 4, we describe the different datasets that we will be using for this analysis with a mixture of real

and simulated quasar light curves. In Chapter 5, we discuss the models that we will be utilizing and the algorithm that we have developed that allows them to work in tandem. Finally, we will discuss our results in Chapter 6.

1.2 Active Galactic Nuclei

Quasars, also known as Quasi-Stellar Objects (QSOs), are powerful signals that have been observed throughout the electromagnetic spectrum ranging from high-energy gamma rays to radio wavelengths. This wide range of detection windows has lead to many different definitions of quasars. In order to understand the definition of quasars in the context of variability analysis, we will first discuss the commonly accepted unified model of Active Galactic Nuclei (AGN) phenomena [Antonucci, 1993].

AGNs are found in galaxies where the luminosity of the central region is com parable to or much brighter than the luminosity of the rest of the galaxy. These luminosities can extend to 10^{47} ergs/s making them among the most luminous ob jects in our universe. This enables the detection of AGNs at high redshifts (currently $z \sim 7$), which makes them important for understanding the evolution of our universe [Padovani et al., 2017].

AGNs exhibit other common characteristics. AGN are visible throughout the electromagnetic spectra from high-energy gamma-ray bursts to low-energy radio observations. The presence of broad emission lines and narrow forbidden and emis sion lines within their spectra is characteristic of AGN [Padovani et al., 2017].

It is widely accepted that all AGN phenomena are powered by a central super massive black hole (SMBH) surrounded by an accretion disk at the nucleus of the host galaxies. The matter accreting onto the SMBH provides the

energy to the disk to power the quasar. This accretion is among the most efficient processes and produces the characteristic bright signals. (See Shakura and Sunyaev [1973] and Rees [1984]).

The accretion disk is surrounded by many different regions including a dusty torus region that can absorb and remit optical and ultraviolet radiation into the infrared band. The broad line region (BLR) and narrow line region (NLR) are made of gaseous clouds that produce the characteristic lines observed in the spectra of different AGN types. The luminosity of this AGN region compared to the host galaxy determines whether the object is known as a quasar or a Seyfert galaxy.

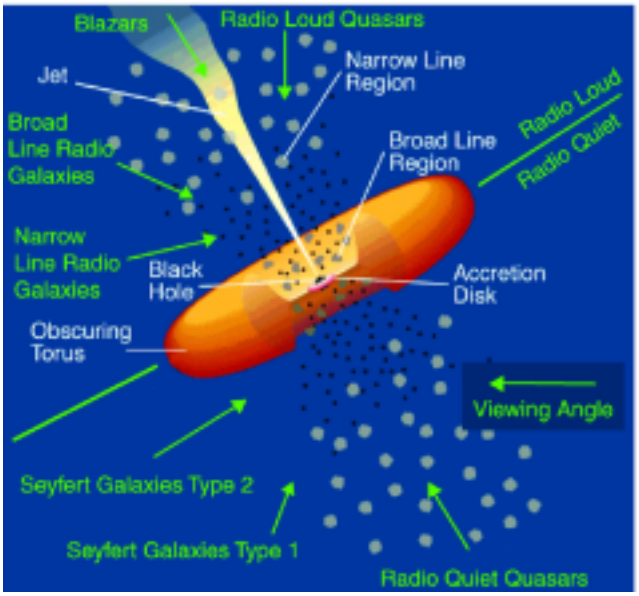


Figure 1.1: Unified Model of AGN(taken from Reynolds et al. [2014]). The different regions are marked in white and the object that is being observed is marked in green

Quasar luminosities are so bright that they outshine the host galaxy. Thus, we observe a point-like object instead of the characteristic disk-like structure of the

galaxy. With Seyfert galaxies, the AGN region is still bright but the structure of the host galaxy can be seen in the background.

AGNs can be classified into Type 1 and Type 2 objects depending on the presence of lines in their spectra. The Type 1 objects exhibit broader emission lines and narrow forbidden lines, while Type 2 objects only demonstrate narrow emission and forbidden lines [Khachikian and Weedman, 1974]. There are further classifications based on the presence of jets. Some AGNs exhibit extended jets that can be detected in radio frequencies. Furthermore, there are objects known as blazars that are characterized by rapid variabilities and enhanced γ-ray activity [Padovani et al., 2017].

The unified model of AGN is described in Figure 1.1. It proposes that these differences originate from the inclination of the object being observed. At low incli

nations from the black hole, the dusty torus region obscures the BLR, only enabling observations of the NLR. This explains the presence of Type 2 objects. At higher inclinations, the BLR can be seen and the object is a Type 1 object. The presence of radio jets also determines whether the object is radio-loud or radio-quiet. For radio-loud objects, when the object is viewed from the highest inclinations (i.e very close to the jets), the jets dominate the signal, giving rise to the high variability that characterizes blazars (see Antonucci [1993] for a full description of the unified model of AGNs).

Due to the dominance of the quasar core over the host galaxy, it is easier to de termine variability properties from quasars in large optical surveys. Thus, this thesis will focus on the variability associated with quasars, though the techniques and con clusions can be further applied to other classes of AGN with enough variability data and distinction from the effects of the host galaxy.

1.3 Variability of Quasars

Quasars exhibit variability across different electromagnetic wavelengths. The am plitude of the variability varies depending on the wavelength, from hours to days in the x-ray to months to years in the optical.

There are many proposed sources of variability. The proposed intrinsic source of variability comes from thermal fluctuations within the disk causing

flare-like or blob like events [Hawkins, 2007]. However, other phenomena around the AGN region have also been proposed to explain the observed variabilities. These include processes such as microlensing by objects along the line-of-sight [Hawkins, 1993] or other transient events such as supernovae occurring during a starburst event in the host galaxy [Aretxaga, 1997]. It is now commonly accepted that thermal fluctuations are the main driver of quasar variability and are used for modeling variability [Kelly et al., 2009]. We will discuss these models further in the next chapter.

Certain components of the central region can cause variability at distinct wave lengths. X-ray variability is thought to originate from the corona of near the central SMBH. The optical and UV variability occurs from the accretion disk, with an extra component in the optical occurring from the BLR and NLR regions. Even within a single wavelength regions, different wavelength bands can probe different regions of the accretion disk. These variation differences allow for characterizing the structure of the source of this variation [Ulrich et al., 1997].

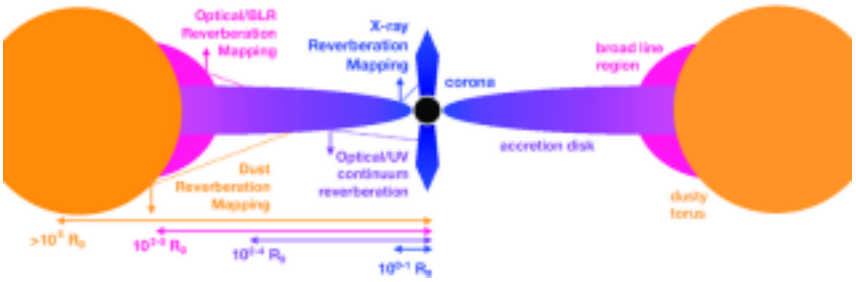


Figure 1.2: Reverbaration Mapping of different regions at different wavelengths. Figure from Cackett et al. [2021]

Quasar variability is difficult to model as the variations are aperiodic and stochas tic [Kelly et al., 2009]. Unlike periodic phenomena, stochastic variability requires statistical modeling techniques instead of a distinct set of periods that can be used to forecast the model into the distant future. However, unlike transient variability, the variability of quasars is not a temporary event and occurs continuously. It is also thought that the variability could be influenced by inherent features of the central SMBH. Extracting the effect of these features on the variability can be complex and non-linear. Thus, stochasticity is tougher to model and requires data-driven models.

1.4 Reverberation Mapping

The variability of quasars unlocks a powerful tool for understanding their underlying physics and structures known as reverberation mapping. Reverberation mapping utilizes the temporal variations in quasar signals to map out the physical structure of different regions, as well as estimate the mass of the central SMBH. This allows for finer resolutions of the central region than direct spatial measurements [Cackett et al., 2021].

The concept behind reverberation mapping lies in measuring the time lag be tween signals generated from the central luminous source and the signals that occur from the surrounding regions. The simplest model is known as the lampost model where the central luminous source is assumed to be the coronal emission of the central region which is reprocessed by the surrounding regions. The time lags char acterize a transfer function, which is convolved with the inherent variability in the quasar to generate the characteristic observed light curve. Thus, knowledge of the transfer function allows reconstruction of the region of the SMBH [Cackett et al., 2021]

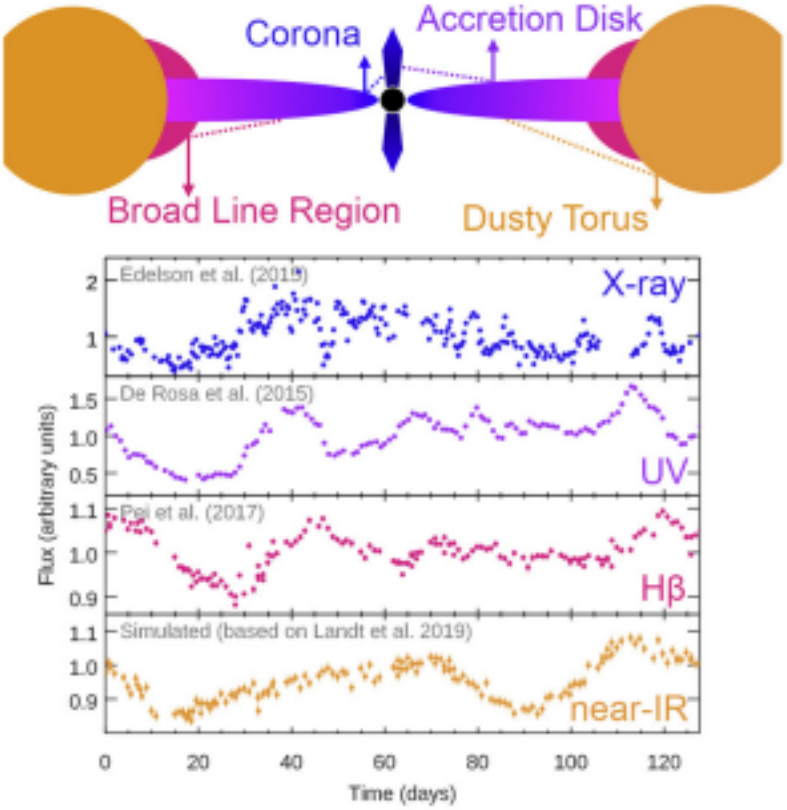
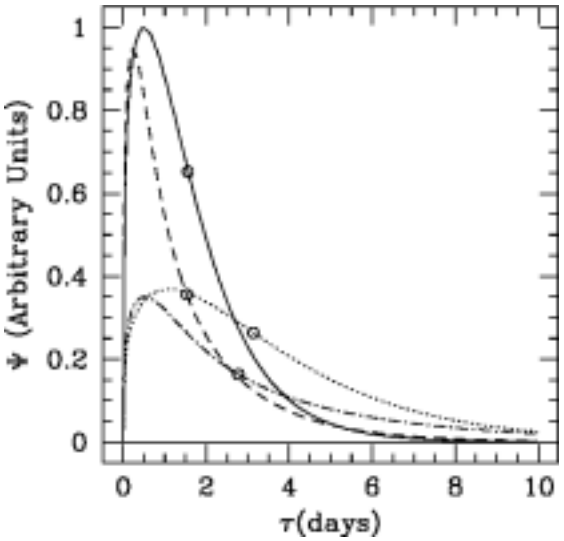


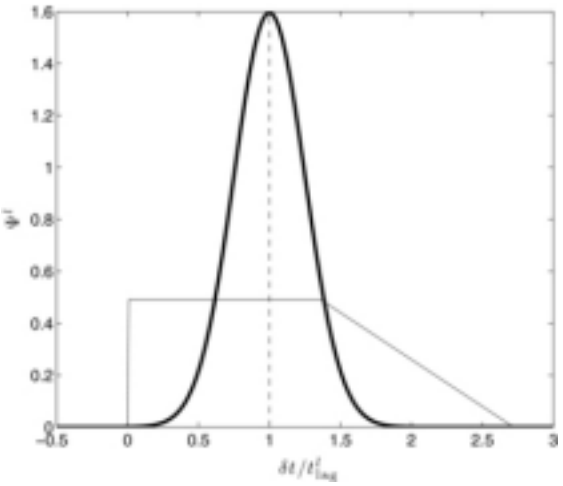
Figure 1.3: The effect of different regions on the light curve with the same implicit driving variability. Figure from Cackett et al. [2021]

The transfer function at different observed wavelengths capture information at different regions of the AGN. The X-Ray reverberation mapping allows for catego rizing structures very close to the SMBH, while UV/optical reverberation mapping can categorize the structure of the accretion disk. Further mapping in the optical region can also categorize the BLR of the AGN. A detailed visualization of the dif ferent regions of the accretion disc can be seen in Figure 1.2. Thus, reverberation mapping through temporal analysis can provide a great deal of information about the central regions of the quasar and how they reprocess signals generated from the central region [Cackett et al., 2021].

Reverberation mapping through spectral lines can provide narrow estimates of the black hole mass. However, large optical/UV surveys that capture many quasars typically provide photometric bands. These photometric bands can probe different regions of the central region. Photometric reverberation mapping involves measuring the time lag between the continuum emission (which is in the UV and optical bands)



(a) Thin disk transfer function from Cack ett et al. [2007] at different inclinations and temperatures.



(b) The transfer functions from Chelouche and Daniel [2012]. One is Gaussian (darker lines) and the other is a top-hat function modified to decay linearly instead of an abrupt end (lighter lines).

Figure 1.4: Different transfer functions from various sources

and the BLR emission (which can be seen in optical bands). This time lag can be used to estimate the size of the BLR as the signal travels through this region at light speeds. In Figure 1.3, the effect of these lags on the light curve can be seen. Then, the BLR radius can be used to estimate the SMBH mass assuming the viral motion of the clouds. The structure of the BLR at different wavelengths can be characterized by different transfer functions. Obtaining these transfer functions from the light curves helps determine the central structure.(see Cackett et al. [2021])

The actual shape of the transfer function depends on the assumed model. In Cackett et al. [2007], the transfer function is derived from a modified Planck function with the assumption of a thin accretion disk, while in Chelouche and Daniel [2012], the transfer function can be assumed to be either a gaussian or a modified top hat function with an assumption of a thick BLR. The particular choice of transfer functions can affect the spread of the time lags and their effect on the light curves.

While reverberation mapping is a powerful technique in theory, it is often con strained by large gaps within the observed light curves. These gaps can cause issues in determining the time lag from cross-correlation functions of the light

curve. Thus, it is necessary to utilize techniques to fill in these gaps and generate finer cadence light curves (conditioned on the observed data) to obtain smooth light curves nec essary for reverberation mapping [Cackett et al., 2021].

1.5 Quasar Variability in the LSST

Large-scale surveys such as the Sloan Digital Sky Survey (SDSS) and Zwicky Tran sient Facility (ZTF) have revolutionized the field of quasar variability by providing high-quality multi-year observations of spectroscopically confirmed quasars. For in stance, the SDSS Data Release 16 quasar catalog contains ~ 750*,* 000 quasars. [Lyke et al., 2020].

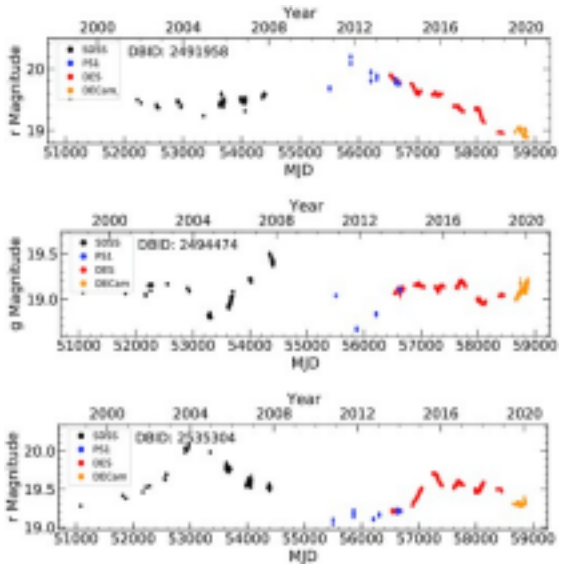
However, the advent of the LSST is set to increase these numbers by many orders of magnitude. The LSST is expected to detect ~ 15 million quasars through its operation time. Some of these quasars will be at higher redshifts than any prior surveys [LSST Science Collaboration, 2009].

The LSST observation strategy provides great opportunities for quasar light curves. The LSST observation strategy is optimized for maximal sky coverage. Short cadence sampling of light curves (on the order of days) will be possible with the LSST, which allows for highly accurate time-domain analysis that can be used for reverberation mapping, as well as other interesting phenomena such as observing changing-look AGNs [Wang et al., 2024] or short-period SMBH binaries that can be detected in the gravitational wave domain by space-based observatories such as the future planned Laser Interferometer Space Antenna (LISA) [Xin and Haiman, 2021]. These events can be observed at much higher frequencies with the LSST than any such large-scale survey before.

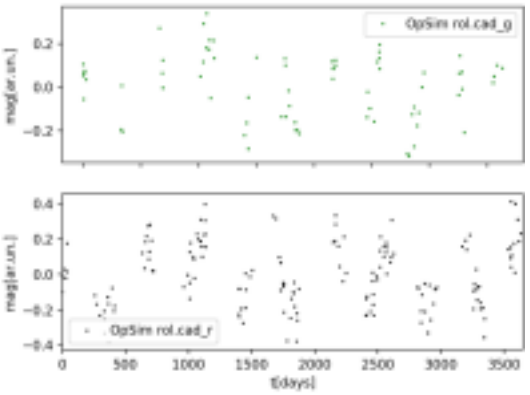
However, there will be gaps during different LSST ‘seasons’ (periods), where no observations will be taken. These gaps will interfere with these short-cadence events, necessitating the need for statistical modeling techniques that can smoothen these gaps [Kovačević et al., 2022a]. In Figure 1.5, we can see simulated light curves with cadences of the ‘rolling’ cadence strategy of the LSST from Kovačević et al. [2022b] as compared to observed light curves from other surveys.

Another important challenge during the LSST is modeling light curves in the initial few years of the survey. During this period, there would be fewer observations to model, necessitating the need for models that can learn quickly

from very few context points. Thus, it will be important to apply novel techniques to extract insights from quasar light curves within the LSST.



(a) 20 year quasars light curves from various surveys (including the SDSS, PS1, DES and DESCam) from Stone et al. [2022] [2022b]



(b) Light Curve simulated with LSST Rolling cadences from Kovačević et al.

Figure 1.5: Comparison of simulated LSST Cadences with other surveys 9

Chapter 2

Modelling of Quasar Variability

There are two main methods to model any light curves. It can be done in the frequency domain or time domain. For irregularly sampled stochastic light curves such as those of quasars from large surveys, frequency-derived features such as the periodogram are distorted. To deal with this, the best method is to utilize Monte Carlo sampling to simulate light curves from periodograms and choose the best fitting light curve. However, this method is computationally expensive, making it unsuitable to apply to large surveys [Kelly et al., 2014].

Thus, time-domain methods are required for modeling light curves in large sur veys. Traditionally, time-domain methods are modeled as realizations of Gaussian Processes. However, these methods have their challenges as well.

2.1 Gaussian Processes

Gaussian Processes are stochastic distributions where each point can be modeled by a multivariate Gaussian with a characteristic mean and covariance matrix. Thus, the entire process is a distribution over a distribution of the Gaussians. Gaussian Processes can be modeled by maximizing a likelihood

function through a Bayesian approach to reconstruct the stochastic process with a confidence interval associated with each point. The obtained covariance matrix can then be used to model the autocovariance function to construct a Fourier transform of the light curve. Thus, Gaussian Processes are a powerful tool to model light curves within the time domain. They also provide the benefit of providing a mean and standard deviation of the process, allowing for multiple samples to be realized from the processes [Kelly et al., 2014]. Gaussian Processes have also been shown to successfully model quasar light

curves Danilov et al. [2022].

However, this power comes with a price. Gaussian Processes require inversion of this covariance matrix which already contains a grid on the order of $n \times n$ where n is the number of points. Inverting this matrix requires an operation on the order of $O(n^3)$ [Kelly et al., 2014]. Furthermore, the initial covariance matrix is a prior assumption of a Gaussian Process model (known as the 'kernel'). For quasar light curves that will be observed within the LSST, we would like to be unbiased by prior kernels to facilitate the discovery of new objects and explain variability that differs from the assumptions that we have made prior.

In order to get around these obstacles, there are several different methods. The most popular method to model the variability is to make more assumptions about the basic processes. However, it is important to note that this imposes assumptions on the models once again.

2.2 Damped Random Walk

The most popular model for modelling quasar light curves is through the use of the Ornstein–Uhlenbeck process or the Damped Random Walk (DRW) process. These are specific realizations of processes that are known as First-Order Continous Autoregressive model or more popularly known as CAR(1) models. The differential equation governing the DRW process is given by:

$$dX(t) = -\frac{1}{\tau}X(t)dt + \sigma\sqrt{dt}\epsilon(t) + bdt \quad (2.1)$$

Where τ is know as the relaxation time of the quasar flux, characterized by $X(t)$. The ϵ is a white noise process that can be assumed to be gaussian. Thus,

there are two components that drive the variability forward. The red noise signals characterized by $X(t)$ are mixed with white noise signals that come from the ϵ term in the DRW process [Kelly et al., 2009].

This red noise is of special interest when studying quasars as it can contain valuable information about the structure of the object or signals indicating the presence of possible binary black holes.

The DRW is an important equation that shows up in another domains of physics as well. Most notably, it describes Brownian motion of particles. A process under going a DRW varies on the short term as per the σ and on the long term timescales

(comparable to τ), the τ relaxation time causes a disapation of the red noise, causing a domination of the white noise process [Kelly et al., 2009].

An advantage of the DRW model over a generic Gaussian Process is the ability to model in a computationally efficient manner [Kelly et al., 2009]. They can be modelled in linear time and with only a few parameters. Furthermore, studies that came out soon after the model confirmed agreement with the DRW model in real data [MacLeod et al., 2010; Kozłowski et al., 2010].

The DRW model is an effective tool to model light curves as well as generate simulated light curves. It can accurately describe several light curves that have been observed. However, the DRW comes with its own share of problems.

2.3 Challenges with the DRW and other Parametric Modelling Methods

Despite the initial success with DRW methods, subsequent results show observed deviations in some real datasets.

The Kepler satellite is able to observe with very short cadences and it was found that very short scale variabilities from Kepler data deviate from the DRW model [Mushotzky et al., 2011]. This was replicated in further studies as well [Kasliwal et al., 2015; Simm et al., 2016; Smith et al., 2018]. Furthermore, other studies have shown that constraining of DRW parameters could be a much tougher task than previously thought. It requires much longer surveys (on the order of 20 years) than any of the large optical surveys currently in operation

[Kozłowski, 2017].

Other methods have been proposed to replace the DRW method and match quasar light curves better. These include CARMA processes which are an extension of CAR(1) processes beyond the first order. The Damped Harmonic Oscillator model is a particular examples of CARMA models that has shown good results [Kelly et al., 2014; Kasliwal et al., 2017; Moreno et al., 2019; Yu et al., 2022].

Parametric-based models such as CAR(1) and CARMA have their advantages when it comes to well-defined datasets with a large number of observations and the ability to capture more physical insight into the driver of the processes behind them. However, they introduce more parameters to be assumed for modeling. The LSST will very likely produce quasar light curves that are different from the expected models [Kovačević et al., 2023]. This is especially important in the early years when

studying variability with fewer data points. This has led to the development of a different class of models that focus on learning variability in a data-driven method.

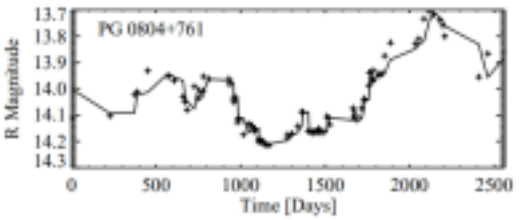
2.4 Deep Modelling Methods

The advent of machine learning and artificial intelligence has revolutionized different fields of science, as well as our society at large. As the amount of data generated and stored gets larger and larger each year, the ability of machines to learn underlying trends from data through complex algorithms improves as well. This same trend is seen in the field of astronomy as well. Large surveys increase the astronomical well of data exponentially. The ability of any one human being to analyze this amount of data is limited. Thus, we require machines to perform this analysis for us. As data science is developed more and more, there is less focus on preprocessing the data and more incentive to draw insights from raw data. However, as scientists, it is important to incentivize interpretability of insights gleaned from such models [Smith and Geach, 2023].

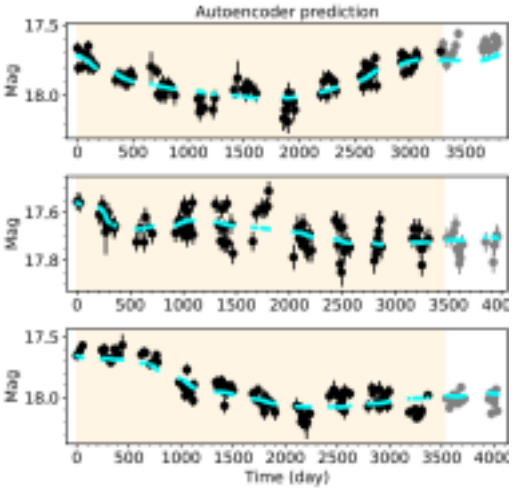
The field of quasar variability is a ripe one for the application of machine learn ing. Deep Neural Networks excel at many tasks that are required for

quasar studies. Classification of variable objects is a critical field for identification of AGN, as well as interesting subtypes (see Mechbal et al. [2024] and references herein). Unsuper vised clustering of light curves (which involves not knowing the underlying classes present) can unlock vital information about sources of variability and group to gether light curves that share similar structure [Kovačević et al., 2023; Čvorović Hajdinjak, 2024]. Models such as Recurrent Neural Networks and Attention-based Transformers are able to learn temporal features with minimal feature engineering, giving them the ability to unlock powerful insights about time series [Smith and Geach, 2023; Vaswani et al., 2017].

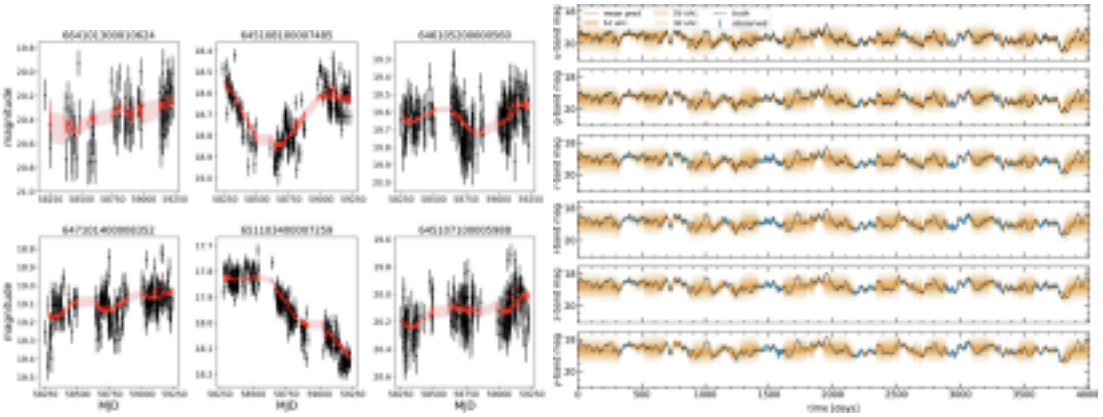
Beyond classification, neural networks perform well on regression and gener ational tasks, generating functions from stochastic datasets. This capability has allowed a plethora of different models to model quasar light curves. Auto-encoders have been utilized to generate representations of light curves and have been shown to correlate with properties driving the quasar such as black hole mass and lu minosity [Tachibana et al., 2020]. Variational autoencoders have been utilized to improve modelling through the introduction of a latent space to learn a wider range of representations as well as detect the presence of changing-look AGNs in the data



(a) Light Curve modelled by DRW from Kelly et al. [2009]



(b) Light Curve modelled by an auto encoder from Tachibana et al. [2020]



(c) Light Curve modelled by a VAE from Sánchez-Sáez et al. [2021]

(d) Light Curve modelled by latent SDEs from Fagin et al. [2024]

Figure 2.1: Different models of quasar light curves

[Sánchez-Sáez et al., 2021]. Modified Bayesian Attentive Neural Processes have been used for the dual purpose of reconstructing light curves, as well as retrieving physical parameters behind them [Park et al., 2021]. Latent Stochastic Differential Equations have been used to model multiband simulated light curves and estimate parameters and time-lags between all the curves [Fagin et al., 2024]. Stochastic Recurrent Neural Networks as well have been utilized to recover CARMA parame ters from light curves [Sheng et al., 2022]. In Figure 2.1, several examples of these models have been shown and compared to the DRW GP modelling.

Our model stands among these methods as a non-parametric way to reconstruct light curves. In addition to the modeling, as performed by the previous methods,

we aim to utilize the non-parametric modeling to generate smooth and interpolated light curves to aid in time series analysis. Initially tested on ASAS-SN light curves [Čvorović Hajdinjak, 2024; Čvorović-Hajdinjak et al., 2022], the conditional neural processes (CNPs) have been upgraded and tested on larger datasets, such as the LSST AGN Data Challenge [Kovačević et al., 2023]. Plain Conditional Neural Pro cesses perform well at learning stochastic representations over few different context points and are also able to produce confidence estimates and capture uncertainty in the modelling. Neural Processes stand at the intersection between Gaussian Pro cesses and Neural

Networks, with the promise to learn diverse stochastic processes with minimal parametric assumptions [Garnelo et al., 2018a,b]. However, CNPs are prone to underfitting and struggle to understand the full structure of temporal data [Garnelo et al., 2018b; Kim et al., 2019; Qin et al., 2019]. Thus, a wide range of upgrades are needed in order to fully capture temporal structure and generate reliable representations of real datasets.

We also aim to understand the insights that our model can learn. Through analysis of the internal workings of the model, we can derive insights into the very processes that drive variability within the quasar light curves. We utilize simple methods to derive quasar parameters out of the hidden layers of our model.

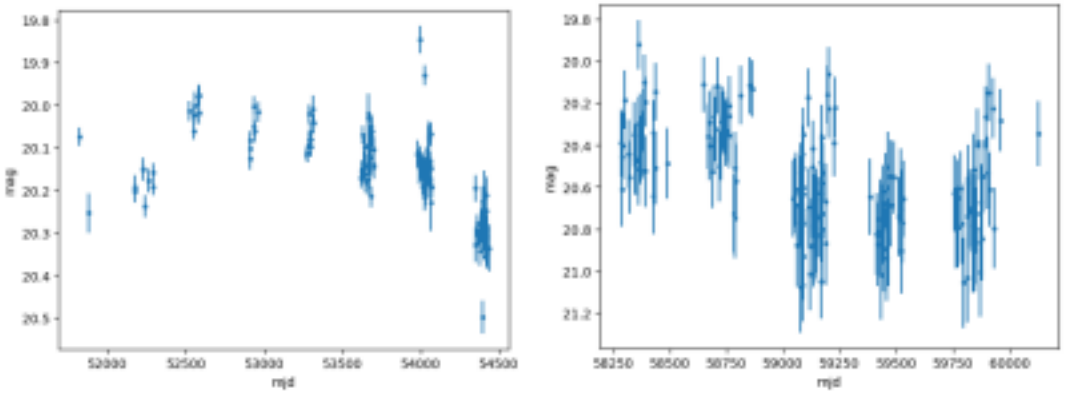
Chapter 3

Data

More and more large catalogs of quasar light curve data are becoming available

with the advent of large surveys. In this thesis, we will restrict our analysis to samples of optical/UV quasar light curves from the LSST AGN Data Challenge (Ultimately from SDSS) and ZTF. Examples of light curves from these surveys are seen in Figure 3.1.

In addition to real data, we use simulated quasar light curves to test if the model can reconstruct processes that contribute to the variability.



(a) Light Curve from the LSST AGN Data Challenge

(b) Light Curve from ZTF

Figure 3.1: Light Curves from different surveys with the corresponding photometric error bars.

3.1 LSST AGN Data Challenge

The LSST AGN Data Challenge is a sample of data from SDSS that provides an example of real observations of objects to mock the observations in the LSST. It was created from a combination of spectroscopically identified objects and variable objects from the Stripe82 region of the SDSS with x-ray-confirmed objects from the XMM-LSS x-ray region and taking the light curves of corresponding sources from the SDSS. While the challenge was mainly aimed at developing selection methods for AGNs, the data provided is also invaluable for the time-series analysis of AGNs [Yu et al., 2022; Savić et al., 2023].

We chose a sample of labeled ‘QSO’ objects from the LSST AGN DC that

contains at least 100 observations of the light curves. This is very similar to the sample chosen in our previous application of CNPs on the LSST AGN DC [Kovačević et al., 2023]. The key difference is that we choose light curves with at least 100 observations across different bands. This decreases the sample size by a few curves leaving us with 984 quasar light curves in the bands u, g, r, i, and z.

3.2 ZTF Light Curves

The Zwicky Transient Facility (ZTF) is a wide-camera all-sky survey that can detect variable objects. With 21 available data releases from 2018 to 2024, there are many short cadence observations available within the ZTF that would better inform the working of our model [The ZTF Collaboration, 2019].

We work with a small subsample of the ZTF. We first select $\tilde{}$1000 QSO observa tions from the dataset at random and then select objects that lie between the 40th and 60th percentile in number of observations. This is due to the large variability in the number of observations in the ZTF dataset as seen in Figure 3.2. This vari ability will induce bias during the padding processes as the high spread will lead to most light curves losing information and just becoming trivial zero variability lines. Thus, we finally obtain a sample of 188 light curves.

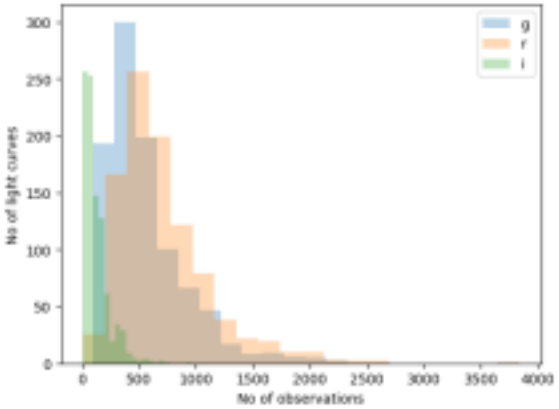


Figure 3.2: Number of observations per light curve in the subsample derived directly from the ZTF database before filtering. The bands g, r, and i are colored blue, orange and green respectively.

Table 3.1: Range of parameters for our simulated DRWs

Parameter	Range
Redshift	0.1-6
$\log(\tau_{DRW})$	0.6-3.5
SF_{∞}	0-0.5

3.3 Simulated Light Curves

3.3.1 Fiducial Light Curves

In order to test the performance of the model, it is useful to test on events generated from the same stochastic processes. Neural Processes literature usually evaluates performance on data generated from Gaussian Processes (see Garnelo et al. [2018a,b]; Kim et al. [2019]; Qin et al. [2019]; Foong et al. [2020]; Dubois et al. [2020]). Luckily, we have a gaussian process that can simulate similar characteristics of quasar variability, the DRW model.

We utilize the astroML implementation of the DRW and provide a random range of redshifts, taus and SF_{∞} parameters to simulate 1000 different DRW models. We vary these parameters as described in Table 3.1. We also degrade this data by assuming that the light curves are observed for a continuous 90 days yearly with gaps between each cycle. We compare the modeled smooth light curve to the real data without this degradation to visualize the performance

of the model as an interpolater for data for which it has no prior knowledge.

Table 3.2: Range of mean timelags for our simulated light curves with Gaussian transfer functions. The standard deviations of these gaussian is a quarter of the mean timelag.

Band	Mean Timelag(τ) (in lightdays)
u	0.1-0.3
g	0.6-0.9
r	1.3-1.7
i	2.3-2.7
z	3.5-3.9

3.3.2 Light Curves with Transfer Function

In order to determine if our model can capture physical information, we generate a sample of quasar light curves with several physical parameters associated with the central black-hole region and time-series features that drive the variability. For a more detailed description of how our light curves are simulated, refer to Kovačević et al. [2022b] Particularly of interest to us is whether the model can capture a convolutional kernel from the curve that can be used to reconstruct the transfer function, hence unlocking information on the central black hole region directly from data.

We simulate a sample of light curves, assuming a DRW driving variability. Then, to simulate the effect of different LSST bands on the light curves, we calculate a transfer function for the LSST band wavelengths. Thus, we are modelling the effects of different regions of the accretion disk on the driving variability.

We utilize two different types of transfer functions. We generate random mean time lags corresponding to each of the bands in the LSST as described in Table 3.2. Then, we generate a Gaussian functions based on the recommendations of Chelouche and Daniel [2012]. This Gaussian is centered at the mean time lag, with a standard deviation of a fourth of the mean time lag.

This transfer function corresponds to an assumption of a Gaussian shape of the BLR [Chelouche and Daniel, 2012].

The second transfer function assumes a thin disk model with the same lamp post model as described above. The transfer function from this set-up has been calculated in Cackett et al. [2007] and is varied for different wavelengths. We follow the same prescription of Kovačević et al. [2022b] to generate transfer functions. We will refer to this transfer function as the Cackett transfer function. This function contains information about parameters such as mass, inclination, Eddington lumi nosity ratios, and redshifts of the black hole. We generate random values in the

Table 3.3: Range of parameters for our simulated light curves with the Cackett transfer function

Parameter	Range
Redshift	0.1-6
$\log(\tau_{DRW})$	0.6-3.5
SF_{∞}	0-0.5
log(Black Hole Mass (in $M_{\odot}$))	8.5-9.5
Inclination (in radians)	$\frac{\pi}{6}-\frac{\pi}{3}$
Eddington Luminosity ratio	0.01 - 0.3

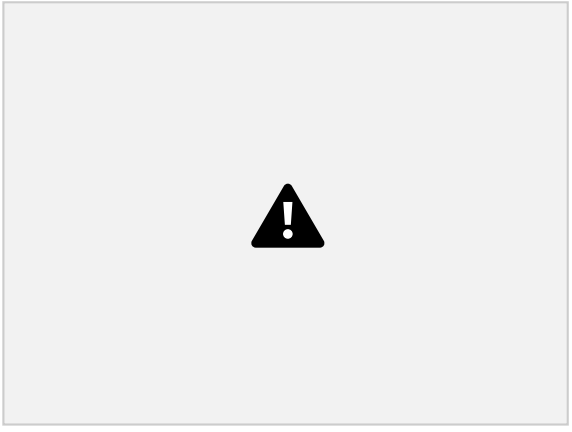


Figure 3.3: Light Curves simulated with various transfer functions in the u-band.

The light curve from the Cackett transfer function is plotted in orange, while the gaussian transfer function is plotted in blue. It should be noted that the Cackett transfer function smooths the light curve more than the Gaussian transfer function.

range inspired by Fagin et al. [2024] but modified to maintain a reasonable range of transfer functions for our model to recover. This modified range is shown in Table 3.3.

The transfer function is chosen randomly from one of these two shapes, normal ized, and then convolved with the DRW to generate the light curve. Thus, a light curve that contains information about a particular region of the accretion disk (via the transfer function at the observed band) and in the case of the Cackett tranfer function, the other parameters of the black hole, is generated. An example of the two types of light curves can be seen in Figure 3.3

3.3.3 Close-Binary Tick Tock Signals

One interesting subsection of quasar light curves is those that may contain embedded signals that hint at the possibility of an impending binary merger. These systems are particularly of interest as they are very rare events and serve as important tests of gravitational physics. Only one known candidate has been discovered [Jiang et al., 2022]. With the large amount of time-series data that we are receiving from the aforementioned surveys, we would like to train a model to correctly identify such events from the light curves.

We utilize the popular transformer models to identify the occurrence of these events in a time series [Vaswani et al., 2017]. We simulate these ‘tick-tock’ signals by adding a damped sinusoidal signal to both simulated light curves, as well as the aforementioned LSST AGN Data Challenge light curves. Then, we feed this to our transformer models to classify the light curves with tick-tocks.

Our simulated light curves are generated in a simpler manner. We simply gen erate random noise. Then, we take the sum of the noise and a damped

sinusoidal wave with an amplitude between 0.001 and 0.1 and a frequency between 0.001 and 0.1 days^{-1}as well. We generate 560 light curves with 280 containing tick-tocks and 280 without tick-tock signals.

Similarly, for the LSST AGN Dc, we add the tick-tock signal generated with an amplitude between 0.0001 and 0.0003 and a frequency between 0.0007 and 0.0009 days^{-1}. We generate 330 light curves with no tick-tock and 99 with tick-tock signals. Out of these 99, half of them are generated with a range of tick-tock signals applied to one light curve and half are generated with one tick-tock signal applied to a range of signals.

Chapter 4

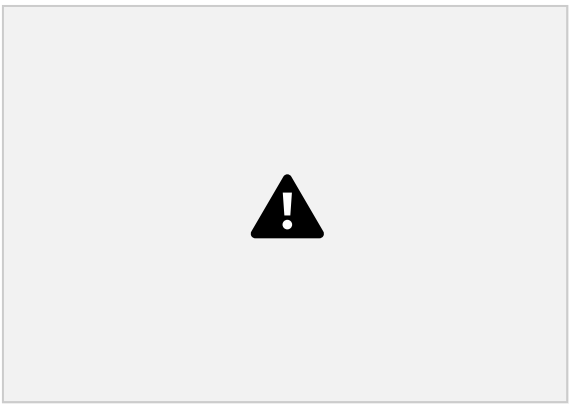
Methods

4.1 Neural Processes

The heart of our modelling algorithm is Neural Processes. Neural Processes combine the benefits of Gaussian Processes models with Neural Network models to generate representations of stochastic processes. Neural Processes can capture subtle patterns in data and scale up to large datasets when trained on a balanced dataset like neural networks. They also utilize Bayesian methods to update priors and generate a distribution at every point, allowing for a better understanding of how well the model is predicting the data at every point like a

Gaussian process [Garnelo et al., 2018b].

The aim of Neural Processes is to generate a representation of stochastic



pro

Figure 4.1: A generalized overview of the encoder-decoder structure of Neural Pro cesses. Context points are encoded into a global representation that is used in tan dem with target points to generate corresponding predictions. Figure from Dubois et al. [2020].

cesses conditioned on the different observations of the process. In our case, this is generating representations of each point on our light curve through an encoder. Then, these different representations are aggregated in some form to form a global representation of the stochastic process (for us, this is a global representation of the light curve). Finally, a decoder is used in combination with this representation to generate predictions on target points [Garnelo et al., 2018b]. This generalized overview of Neural Processes can be seen in Figure 4.1. The exact mechanism de pends on the particular choice of model from the Neural Processes family and we will discuss the differences through the rest of the chapter.

4.1.1 Conditional Neural Processes

Conditional Neural Processes (CNPs) are the simplest and first proposed forms of Neural Processes. They generate target predictions based on a probability distri bution that is factorized on the representation of the observation points [Garnelo et al., 2018a].

Let us assume that our observation set is known as O with a length of n. Every element in O consists of a pair of points (x,y) where x are the inputs and y are

the outputs. We also have a target set T that consists of just input points with a length $_{i=0}$ and $T = (x_i)^{i=n+m-1}$ of m. Thus, $O = \{(x_i, y_i)\}^{i=n-1}$ $_{i=n}$. For a light curve, our O is a set of observations and magnitudes pairs that we use as our context points, while T is time stamps that we would like to predict. In practice, we choose an O that is a subset of the entire light curve, while we choose a T that is larger than O while training. During test time, we choose T twice, once for the entire light curve and then again, we choose a large number of points to generate short cadence predictions of the light curve [Garnelo et al., 2018a].

We use our O to determine a probability for each point in T. This means we want to determine $p(F(T)|T; O)$. We assume that the processes are Gaussian and thus, this probability is a Gaussian probability density. We call this a Gaussianity assumption.

Another key assumption in Neural Processes is factorization. We assume that $p(F(T)|T; O)$ is Π^{m-1} $_{i=0}$ $p(F(x_i)|x_i; O)$. This factorization ensures that the process remains stochastic, as it implies consistency under permutation and marginalization (i.e any permutation of the time series is valid and the probability of any point stays the same integrating out other points in the distribution) [Garnelo et al., 2018a].

With these assumptions in place, Neural Processes utilize Neural Networks to generate these encodings. Thus the points are encoded with a multi-layer perceptron as:

$$R_i = MLP((x_i, y_i)) \quad (4.1)$$

where x and y come from O.

These representations are then aggregated to generate a global representation. In our case, this is a representation of our light curve with the dimensionality of the last layer of the multilayer perception (MLP). The initial mechanism for this aggregation was simply taking the mean of all of the representations [Garnelo et al., 2018a,b].

$$R = \frac{1}{N} \sum_{i=0}^{N-1} R_i \quad (4.2)$$

This representation is now combined with the target points in order to generate both a mean and a standard deviation prediction for each target point drawn from T. This is the decoder and it is represented as:

$$\mu_i, \sigma_i = MLP(R, x_i) \quad (4.3)$$

where x comes from T.

This process is far more computationally efficient than the Gaussian Processes because it just has to act once on the context set and then predict each target point. Thus, it is only of the computational complexity of O(C+T) where C is the number of contexts and T is the targets [Garnelo et al., 2018a].

Now the question is, whether this sort of encoding can actually approximate any function. Indeed, as shown that any function on a set (such as times and magnitudes for a light curve) can be expressed as a function acting on a sum of products of functions on the set [Zaheer et al., 2017]. Assuming S to be a set, f to be a function on the set and ρ and ϕ to be functions on the elements of the set:

$$f(S) = \rho \left(\sum_{s \in S} \phi \right) \quad (4.4)$$

Thus, we can use the MLPs to construct a full representation of the set of light curves.

With this basic model in mind, the next step is to train and optimize this model. During training, the CNP is shown a diverse set of realizations of the stochastic

process. As the model trains, it can learn how to learn across different datasets. This suits our goal of modeling a large variety of light curves very well. As we saw earlier, the final representation is a Gaussian. For conditional neural processes, optimizing the model is rather straightforward. We utilize the predicted Gaussian distribution and calculate the logarithm of the probability (LogProb) that the observed target points belong to this distribution. For a Gaussian, the PDF is
defined as:

where $y_t$is the target point. $\sqrt{2\pi\sigma}e^{-\frac{1}{2}(\frac{y_t-\mu}{2\sigma})^2}$(4.5)

Prob =^{1}

Thus, the log probability can be calculated at every point as

$$\text{LogProb} = -\frac{1}{2}\log(2\pi) - \log(\sigma) - \frac{1}{2}\left(\frac{y_t-\mu}{\sigma}\right)^2 \quad (4.6)$$

However, there is another consideration that needs to be made. The higher the loss, the better the distribution captures the value. However, most algorithms in machine learning are used to minimizing a loss. Thus, we add a negative sign to the loss. This ensures that minimizing the value will maximize the log-likelihood. We also draw attention to the use of a mean error calculation. In prior works, we have utilized a summed probability instead. The usage of a mean shrinks the gradients, allowing for faster training of the model and comparison of training across light curves of different lengths.

$$L = \frac{1}{N}\sum_{i=0}^{N} \log\left(p(y_t|x_t; C)\right) \quad (4.7)$$

We will also refer to this loss as the reconstruction loss of the light curve. Alter natively, we will refer to it as the LogProbLoss or LogLikelihood Loss Garnelo et al. [2018b].

This framework is the heart of the Neural Processes modelling algorithm. We will refer to this basic mean-pooling model conditioned on context points as a Basic CNP. In theory, any function can be modeled by BasicCNPs [Garnelo et al., 2018b]. However, CNPs are very prone to underfitting in practice. Thus, extra modifications are needed to scale this framework for the generation of more reliable predictions of realistic processes. The next three sections describe modifications that are essential for working with real quasar light curves.

4.1.2 Attention

One of the reasons that BasicCNPs end up underfitting data is the mean pooling mechanism. This mean pooling implies that every point on the light curve con tributes equally to the target point prediction. This is a flawed assumption for time series [Kim et al., 2019]. For example, consider a simple light curve that has a single flare event with the rest of the light curve remaining flat. A more accurate approach to modeling the top point of the flare would be to utilize the points that lead to the rise and fall of the event. However, mean pooling can cause the contribution of these points to be dwarfed by the flat nature of the remaining light curve. Thus, we would like a method to inform the model which points to prioritize when modeling.

Attentive mechanisms are revolutionary machine learning algorithms that have revolutionized the ability of models to understand sequential data. Attention was initially developed for text processing and improved the ability of models to under stand intertextual relations, without the need to process data sequentially [Vaswani et al., 2017].

This ability also makes attention suited to the understanding of time series. Attention can provide the needed mechanisms to improve the temporal structure of the model predictions. The use of attention to Neural Processes was first proposed by . They proposed that the mean pooling step was acting as a bottle-neck and causing underfitting of the context set by the neural processes. To alleviate this, they utilized attentive mechanisms, transforming the BasicCNP into an AttentiveCNP, which we will refer to as an AttnCNP [Kim et al., 2019].

The attentive mechanism works as such. We have a set of points keys and values (k_i, v_i). We utilize queries q in order to determine a weight to each key and then get a value corresponding to each query point. We can assume that the keys, values, and queries are ordered in matrices of K, V and Q with dimensions ($n \times d_{\text{keys}}$),($n \times d_{\text{values}}$) and ($m \times d_{\text{queries}}$). In this thesis, we will utilize two types of attention that are the most popular. These are dot product attention and multi-head attention [Kim et al., 2019].

For dot product attention, we utilize a simple dot product of these matrices.

$$\text{DotProductAttention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (4.8)$$

Multihead attention is an extension of dot product attention where each of the matrices is first linearly transformed by utilizing key, query, and value weights for

a specific head. This process is repeated for each head. Then, the heads are con catenated and linearly transformed back into the dimensionality of the original data [Kim et al., 2019].

These two attentive mechanisms were found to have the best results with Neural Processes. Thus, we mainly provide the option to utilize these two. In our models, we have found that dot product attention is needed to capture the high degree of variance associated with quasars. However, we utilize multihead attention in our transformer models.

Now, the attentive mechanisms can be utilized in the model. There are two types of attention depending on the keys and queries. If they are the same, the attention is known as self-attention. Otherwise, it is known as cross-attention. We insert self-attention into the encoder. This generates a local representation for each point with the MLP representation as the values, while the keys and queries are both the context input points (for us, this is the context time steps).

The next modification is to utilize the attention to inform the target predictions. This is done by using a cross-attentive mechanism with the local representations as the values, and the keys as the context input points again. This time, the queries are the target input points (which are the time steps that we would like to predict). Thus, we have a representation that is generated for each target point. Note that this representation still utilizes the entire light curve. The difference from mean pooling is that it can understand the importance of different context steps to the overall structure.

The rest of the structure remains the same for AttnCNPs. We still use the same loss and train in the same way. The difference is that the attentive mechanism increases the time complexity of the model. Since self-attention relates each context point to every other context point, the complexity is now $O(C^2)$. Similarly, the cross-attention complexity is $O(CT)$ because it relates each context point to each target point. Thus, the entire complexity is $O(C(C + T))$ as opposed to $O(C + T)$ for the BasicCNP. However, this cost is potentially offset by the quicker convergence and better reconstructions by AttnCNP over BasicCNP [Kim et al., 2019].

4.1.3 Latent Neural Processes

One problem of the Conditional Neural Processes lies in the factorization assumption that was made. This factorization assumption assumes that each point is indepen dent of the distribution at the other points. This means that the model predicts

each point independently of each other. While attention helps alleviate this, the model would be better if it could understand the global uncertainty and predict similar values at similar target points. Furthermore, Conditional Neural Processes are not able to generate coherent samples. Coherent sample predictions can help the model capture a wider range of possible predictions. This is helpful, especially in light curves, where it can catch likely flare events. Gaussian Processes account for these problems through a covariance matrix, but utilizing similar mechanisms in Neural Processes will increase complexity [Garnelo et al., 2018b].

To address this concern, the latent path was added to conditional neural pro cesses. We will call these models LNPs to distinguish them from CNPs. However, it is important to note that in literature, NPs usually indicate neural processes with the latent path as we will describe in this section.

The latent path utilizes an encoder similar to the CNP encoder described above. Thus, latent path keeps the MLP and self-attention from earlier. Then, the latent path mean pools the representations similar to the BasicCNP. However, the differ ence is that this representation is now passed through another MLP layer in order to generate a multidimensional mean and standard deviation that characterize a Gaussian latent space associated with the global representation of the curve. This latent space is sampled (this is done by taking the sum of the mean with the product of a stochastic factor (between 0 and 1) and the standard deviation). This sam ple is then combined with the target points to produce the predicted target points [Garnelo et al., 2018b].

The deterministic path from the previous section with cross-attention can be used as well. This means we use the same cross-attention generated representation, the latent space sample, and the target points to generate the predicted target points. Figure 4.2 shows the model's flow-chart and the utilization of the different representations generated by the model [Kim et al., 2019].

However, this latent path changes the calculation of the loss and

optimization. We now have to change the way we calculate the error. From the marginalization assumption we have for Neural Processes, we can write the conditional probability
as:

$$\log p(y_t|x_t; C) = \log \int (p(z_t|C)p(y|x, z)) \quad (4.9)$$

We use the shorthand $p(y|x, z) = \Pi_t p(y_t|x_t, z_t)$

The second term here can be thought of as factorized and Gaussian. However, the entire probability is not, as $p(z_t|C)$ is not necessarily Gaussian. This means that the

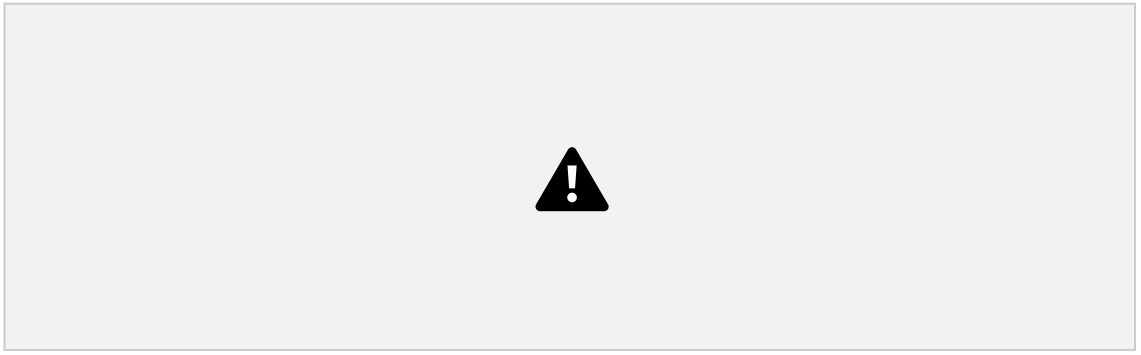


Figure 4.2: A detailed description of the mechanism behind both LNP and AttnL NPs. Figure from Kim et al. [2019]. Note the addition of the self attention and cross attention mechanisms in the AttnLNP.

scope of modeling increases and the samples need not be Gaussian either. However, this does complicate the log-likelihood as it is no longer necessarily Gaussian and is integrated over z.

There are two methods of getting around this and estimating the loss through another method. The first is the ELBO method which is the standard method in literature, while the second utilizes a Monte-Carlo approximation of the integral. These are known as Neural Processes Variational Inference (NPVI) and Neural Processes Maximum Likelihood (NPML) respectively [Garnelo et al., 2018b; Foong et al., 2020].

NPVI was the first method developed to address this issue. It borrows heavily from principles utilized by Variational Autoencoders known as amortized variational inference [K. and W., 2013]. The main idea is to utilize sampling from the posterior to reduce the variance of samples. It is intractable to estimate the latent variable from both the context and target sets as it involves a complicated

Bayesian estimate across z. Instead, we replace this by passing both the context and target sets through the model. Then, using Jensen's inequality to find an evidence-based lower bound (ELBO) on our probability [Garnelo et al., 2018b; Kim et al., 2019]. This is found to be:

$$\log p(y_t|x_t; C) \geq \mathrm{E}_{z\sim p(z|D)}[\log p(y_t|x_t; z) - KL(p(z|D)p(z|C))] \quad (4.10)$$

Here, D is the input set of both the context and target. Since the z is Gaussian for both D and C, we can calculate the Kullback-Liebler divergence (KL divergence) between them. Thus, we can now try to minimize this quantity. This can be shown to reduce to minimizing the same previous loss while subtracting the KL loss term

from the loss. Thus, NPVI ensures that we can generate coherent samples while training the model on just a single sample. This makes it efficient in terms of computation time and resources. This also causes the training of the network to be different from the testing of the network, as the model has access to both context and target at train time but not at test time [Garnelo et al., 2018b; Foong et al., 2020].

However, NPVI has its disadvantages. It focuses on creating the best latent distribution. Also, the encoder has the dual task of creating a posterior distribution, as well as encoding the data. This could lead to detrimental effects during the training of the model. Thus, there are NPML methods that are usually found to have better performance but are more expensive in time and resources.

NPML utilizes a simple Monte Carlo approximation on the intractable loss.

$$\log p(y_t|x_t; C) = \log \frac{1}{L} \sum_{i=1}^{L} p(z_t|C)p(y|x, z) \quad (4.11)$$

This tells us that we can take multiple samples of our data and try to maximize the average loss of each of the samples. However, this is not an unbiased estimate. Instead, there is a negative bias and the probability here is a conservative estimate of the true probability. Furthermore, it was found that the number of samples has to be on the order of 20 to achieve a good performance [Foong et al., 2020; Dubois et al., 2020].

In practice, it is found that NPML performs better than NPVI, but requires

more computational time and resources [Foong et al., 2020; Dubois et al., 2020]. For a large dataset like the ones we would expect from the LSST, we would recommend the use of the NPVI method to utilize computational resources better and thus only present results utilizing NPVI in this thesis.

At this point, the Neural Processes module sounds very similar to a Variational Autoencoder [K. and W., 2013]. Indeed, Latent Neural Processes utilize concepts of Variational Autoencoders heavily. However, an important difference is that Varia tional Autoencoders generate samples from learning from a given input. This means that there are no input and output points in a VAE. Neural Processes learn from input and output points and generate representations at the queried target input points [Garnelo et al., 2018b]. This is more helpful for the task we wish to accom plish with Neural Processes, which is to interpolate quasar light curves. We see a comparison of VAEs and Neural Processes, along with another model for generating samples from data known as a Neural Statistician in Figure 4.3.

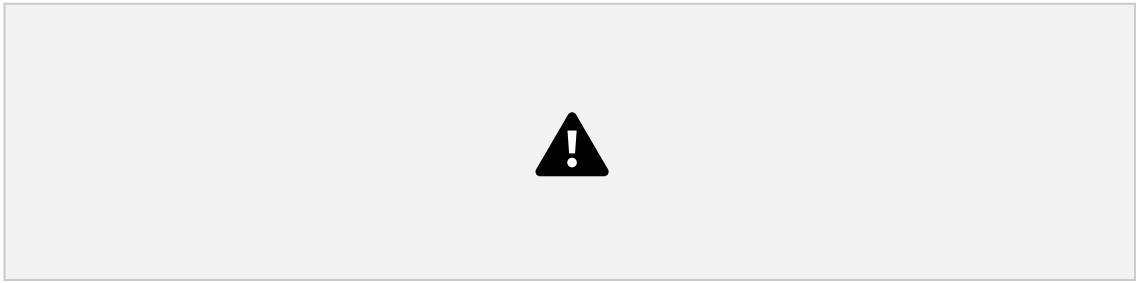


Figure 4.3: The main difference between NPs and VAEs is the use of the context points to inform the latent representation in Neural Process models. Figure from Garnelo et al. [2018b].

Following the literature, we will drop the term latent from the description of our model and refer to it as an ANP (Attentive Neural Process) model.

4.1.4 Parameter Estimation

Finally, we would like to see how well the modeling of our light curves can help in recovering physical parameters. Specifically of interest to us is the ability of the model to capture the underlying convolutional kernel of the transfer function. Estimating this kernel from the data would allow for probing the structure of the quasar directly from variations in the light curve without the need to decouple continuum contribution to quasar lines and their reverberations.

In order to do this, we propose two methods. The first is similar to the Baysian Attentive Neural Processes for parameter estimation proposed by Park et al. [2021]. We add an extra MLP into the model that can encode the learned information in the deterministic and latent samples to provide estimates of the transfer function. We do not utilize the LSTM layer as we aim for faster models and the LSTM did not add significant improvements in our case. We also proceed using a Bayesian approach of generating a mean and standard deviation and maximizing the likelihood of the target transfer function drawn from this distribution. Thus, we modify the loss term by adding an additional loss from the parameters

$$\text{Loss}_{\text{param}} = \frac{1}{N} \sum_{i=0}^{N} \log\left(p(\text{param}|z, R; C)\right) \quad (4.12)$$

Where the parameter is reconstructed from the latent space sample z and the attention-weighted representation R which are both conditioned on the context set [Park et al., 2021].

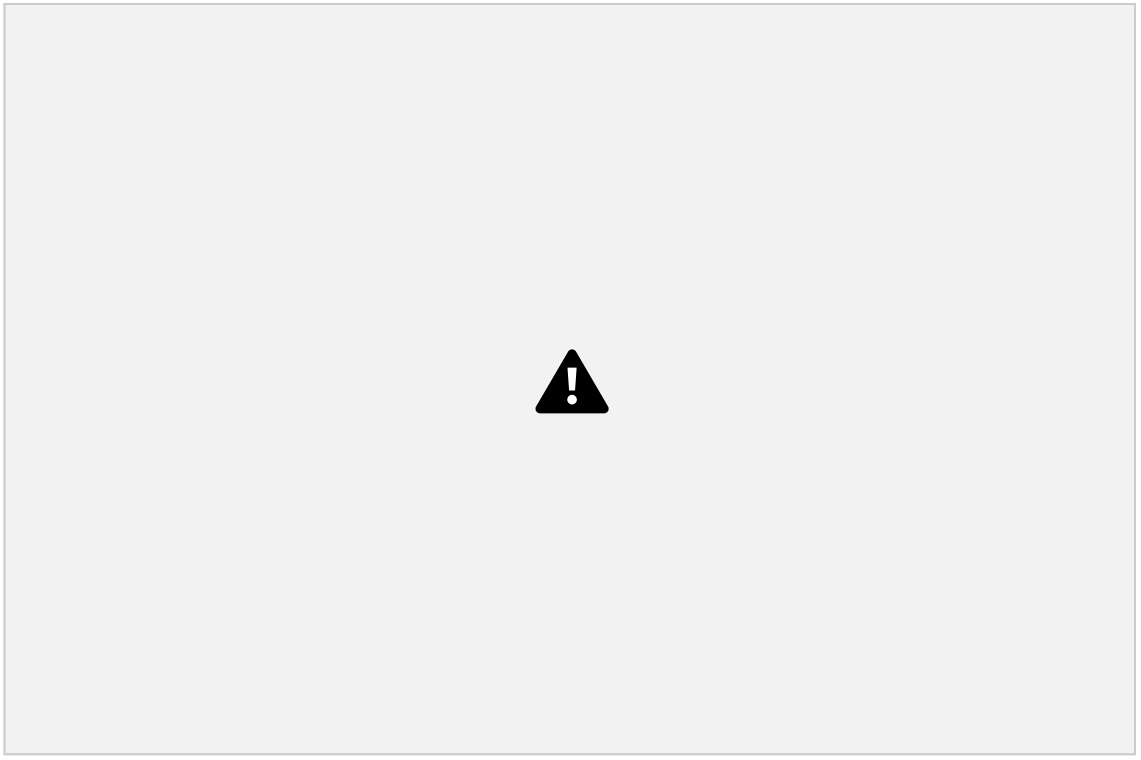


Figure 4.4: The Neural Processes model with the addition of a parametric MLP. Figure modified from Park et al. [2021]. We utilize this MLP both in tandem with the model during training or only after being trained.

We include a hyperparameter β_{param} as well to control the parametric loss's effect on the training of the model. We typically set β_{param} as 0.01 as this does not impede the training of the model. This model can be seen in Figure 4.4.

The second method is to examine the trained model for hints of correlation with parameters. This was done by Tachibana et al. [2020], with the autoencoder network. Thus, instead of adding the MLP during training time, we examine the hidden representation of the light curve directly instead. We average the attention generated representation across the curve, utilize an MLP, and train it to generate the parameter or transfer function of choice with the same loss metric as described above.

4.2 Self Organizing Maps

We have discussed Neural Processes and the modifications that we have added to them to upgrade from Conditional Neural Processes to Attentive Latent Neural

Processes. However, it would be impractical to train a large model on the plethora of data that we have coming from the LSST. Instead, we would like to partition our dataset before training with the ANP. This partitioning will provide a two-fold advantage to the modeling process.

The first advantage is that the dataset will be smaller. Thus, we can train many models in parallel that will all be examining smaller subsets of the model. This will speed up the training process. However, a good data-driven model will be able to generate balanced datasets. Through careful clustering of light curves, the model can generate different insights on different clusters.

We utilize Self Organizing Maps (SOMs). SOMs can quickly learn patterns of high-dimensional data and provide clustering representations. Furthermore, their ability to process light curve observations as features allows new data points to be added easily without complicating the clustering algorithm [Kohonen, 1990].

A typical SOM is two-dimensional and consists of a grid of nodes. Each of the initialized nodes is a mapping from a weight vector to a position in the grid. We will use a 1-dimensional index for the weights for simplicity here. We will

also represent the weight vector with w. When the SOM is training, it updates based on one input at a time. It identifies the node that is closest to the chosen input point (based on the Euclidean distance between the input and the weight) and this is deemed the winner node or the Best Matching Unit (BMU).

$$d_{min} = \text{argmin}||x - w_i|| \quad (4.13)$$

where x is the input point. The BMUs weight is now updated based on the

formula. $w_{i+1} = w_i + \eta(t)h(i)(x - w_i)$ (4.14)

Here, h is the neighborhood function that controls how much each node affects each of its neighbors. We choose a Gaussian neighborhood function for simplicity. This Gaussian is centered on the winning node and is parameterized by a standard deviation σ. We also have η which is the learning rate of the model. σ and η are both hyperparameters that can be tuned for different performances for the model. We decrease both parameters as the training proceeds with a simple decay function:

$$x_{i+1} = x\ 1 + \frac{i}{(N/2)} \quad (4.15)$$

Where i is the current iteration and N is the maximum number of epochs of the model. The number of epochs and the grid size of the SOM are also important hyperparameters for the SOM training [Kohonen, 1990].

When the SOM has been fully trained, each of the input vectors are then assigned to the BMU's group. While there are many methods to interpret the clusters from this configuration, two are of the most interest for an unsupervised clustering. The first is to interpret each nodal group as a cluster. The second is to group together nodes in order to identify clusters. The easiest way to do so without imposing order on the data is to view the weights associated with each node. Each node is paired with the neighboring node that has the minimum euclidean distance from it. This is repeated for every node that has an even lower euclidean distance from the previous distance until the pattern ends at one node [Hamel and Brown, 2012]. Thus, the gradients are identified among the nodes based on those that are closer to each other and can define a grouping of nodes, which we will call 'gradient based clustering'. This method has the capability of identifying clusters in the dataset without the number of

nodes setting the number of clusters, while still allowing the nodes to learn subtle differences while training.

Thus, our SOM can learn topological differences in the light curves and assign each light curve to a representation that matches it best. We provide each light curve as a data point to the SOM.

4.3 Complete Model Setup

In this section, we describe how the entire model setup works. When dealing with real light curves, we first apply a cleaning procedure to deal with outliers. From the recommendation of Sánchez-Sáez et al. [2021], we first remove all points that have an error of more than a magnitude. Then, following the recommendation of Graham et al. [2015]., we apply a three-point median filter on the light curve. Then, we perform a 5th-degree polynomial fit to the curve and remove all points that significantly deviate from the fit. Initially, we start with 0.25 mag deviation. If too many points are removed, we increase the threshold until no more than 10% of the points are removed. Finally, our light curves are ready to go through the pipeline. Our process is very similar to the previous pipeline for the CNPs [Kovačević et al., 2023; Čvorović Hajdinjak, 2024].

The very first step is to cluster our light curves. We pad our light curves to the length of the longest light curve. Then, we scale our light curves. From ex perimentation, we have found that the best result is to utilize a min-max scalar in the range of [-2,2] to achieve the best results. These scaled light curves are

passed through the SOM and assigned clusters through either the BMU node or the gradient-based grouped node clusters. We include many metrics to understand the effectiveness of the clusters. These include Quantization Error and Topographic Error in addition to standard unsupervised clustering metrics such as Silhouette Score, Calinski-Harabasz Index, and Davies-Bouldin Index. The Quantization error captures how well each cluster representation captures the inter-cluster light curves. The lower the QE, the better the clusters represent the data [Kohonen, 1990].

$$QE = \frac{1}{N} \sum_{i=1}^{N} (x_i - w_{BMUi}) \quad (4.16)$$

The Topographic Error captures the topological preservation of the data by determining the average distance between every curve's BMU and the second BMU. The lower the topographic error, the more distinct each node is Bauer et al. [1999]; K. [1996].

$$TE = \frac{1}{N} \sum_{i=1}^{N} (w_{BMUi} - w_{SBMUi}) \quad (4.17)$$

The final three scores are standard metrics in unsupervised clustering. The Silhouette score determines the separation of different clusters by measuring the distance of points from the center of the cluster to the center of the nearest cluster. A higher score indicates more separated clusters. The Davies-Bouldin Index measures the *similarity* of the points to their cluster center as compared to the nearest cluster. Similarly to the silhouette score, it is higher for more well-defined clusters. Finally, the Calinski and Harabasz Index measures the variance of the clusters as compared with the total variance of the dataset.

Through the interpretation of metrics, as well as hyperparametric tuning of the SOM and choice of clustering mechanisms, one can determine the optimum clustering for the dataset. Then, we utilize an ANP for each cluster. Our ANP is built completely in PyTorch.

Once again, for the ANP we pad the light curves to the length of the longest light curve. Then, we scale the light curves in the same way as before. This scaling ensures that we capture variances in a scale-free manner, treating the variations within each light curve as equal. Finally, we split our data into sets used for training, validation, and training. We follow the standard recommendation of around 80% of the data being training data, with 10% each for validation and testing. We also augment our data by adding and subtracting the measurement errors from each point. We have

seen that this method has improved our model's capability to model error, has led to quicker convergence, and prevented overfitting (see Kovačević et al. [2023] for a detailed discussion).

Now, we train our ANP on the training data. In every epoch of training, we select a random number of points between 60-80% of the light curve as context points and then 80-100% of the light curve as target points. We batch our training set and utilize average statistics across the batch in training.

We also evaluate the performance of the models on the validation set. We choose the model that performs best on our validation set to avoid overfitting to the test set. We also include an early-stopping mechanism if the validation loss no longer decreases.

Our training is optimized for the reconstruction loss as explained in the Neural Processes section. However, we also evaluate and report metrics that are more standard for regression in our package, namely the Mean Standard Error and the Mean Absolute Error of the data.

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N} w_i(y_i - \hat{y}_i)^2 \quad (4.18)$$

$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{N} |y_i - \hat{y}_i| \quad (4.19)$$

The MSE metric places more emphasis on outliers as it utilizes the square of the error, while the MAE utilizes the absolute error so is less sensitive to outliers. Our implementation of the MSE error also differs with the inclusion of a weight that corresponds to the error of the observation. It is important to note that these metrics do not influence the training of the model, however, they provide a good idea of how well the model is working and training.

Finally, when the model is trained, we provide the model with the entire dataset and produce modeled light curves from the train, validation, and test sets. We evaluate metrics based on the entire curve and also produce short cadence repre sentations using the model to interpolate the light curve. We can also utilize the model to provide estimations of the parameters.

We also have a range of hyperparameters typical of machine learning models. We typically utilize an MLP that first encodes our observed magnitudes and times into a 128-dimensional representation. Then, we pass it through 3 layers of linear encodings spaced by activation functions. We find that the Leaky ReLu activation

function performs best, as regular ReLu encoding produces too many zeros for our encoding. We also utilize a latent space representation of equal size as our encoding size (i.e 128 dimensions) and then pass our representation concatenated with our target points through a 4-layer MLP decoder that produces a characteristic mean and standard deviation of a normal distribution.

We also utilize an Adam optimizer with a learning rate of 10^{-3}[Kingma and Ba, 2014]. We use a batch size of 8.

The standard deviation of the distribution (both to generate a latent distribution and the final prediction) is modified to remain positive. The model outputs a logarithmic standard deviation. We perform both a softmax and sum function to add together the logarithmic outputs and generate the standard deviation of the distributions. Then, we add a small constant of 0.01 to ensure that the standard deviation is non-zero. (Refer to Dubois et al. [2020] for a more detailed explanation)

4.4 Transformers

As a special application of time-series analysis, we also provide a discussion of the uses of attentive mechanisms in extracting hidden features from quasar light curves. We utilize a Gated Transformer Network to detect hidden close binary signals in the dataset [Liu et al., 2021].

A traditional transformer extends from the Multi-Headed Attention model to im plement an encoder-decoder model to process sequential data. The encoder consists of stacked attention layers that contain a multi-headed self-attention mechanism and a simple feedforward neural network. The decoder contains similar stacked lay ers with the addition of a cross-attention layer that calculates previous inputs. It also utilizes positional encoding to allow for sequential knowledge of the data and masking to prevent the transformer from ’cheating’ and predicting the latter part of the sequence [Vaswani et al., 2017].

The gated transformer network (GTN) makes a few modifications to the trans former mechanism as seen in Figure 4.5. It adds a non-linear embedding before the stacked layers. Next, it splits the encoder into two towers to analyze multivariate time series data. The first tower utilizes positional encoding and attentive mech anisms to learn step-wise correlations across the different channels that comprise the multivariate time series. The second tower looks at each ‘channel’ (one variable of the multi-variate time series) and utilizes attention to learn features across the channel [Liu et al., 2021].

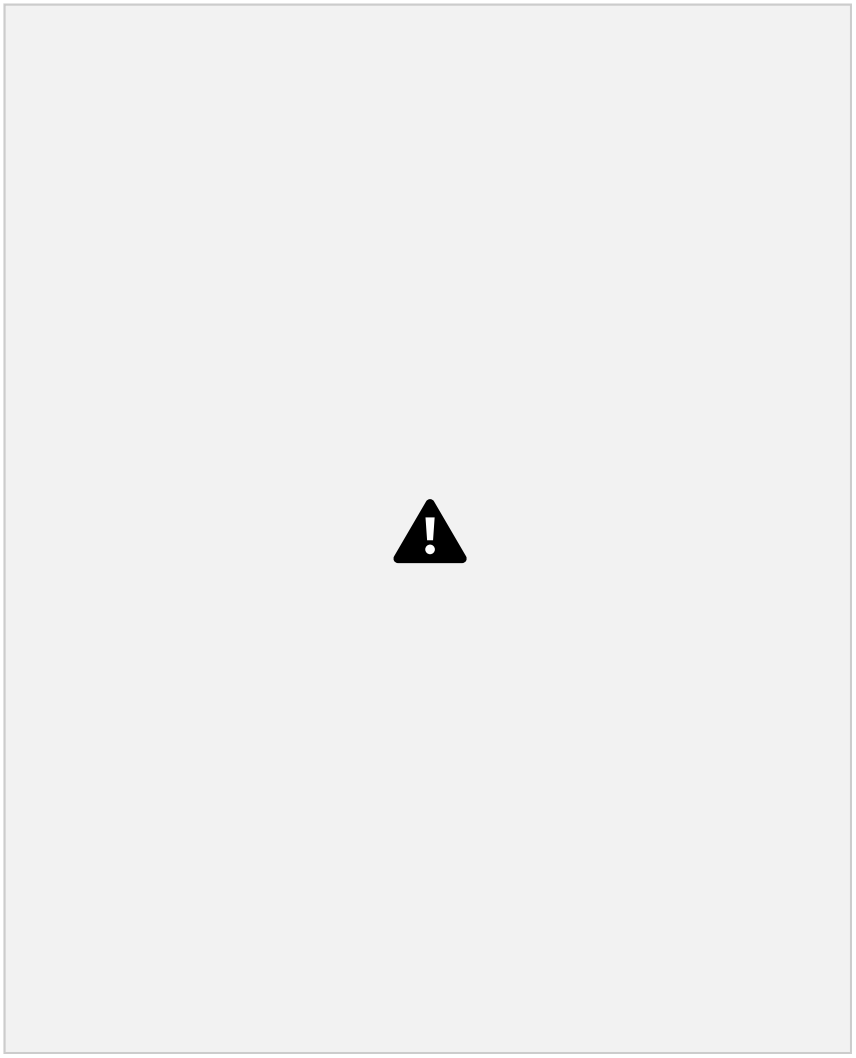


Figure 4.5: Model Architecture of a Gated Transformer Network. Figure from Liu et al. [2021].

Finally is the namesake gating mechanism. The gating mechanism utilizes non linear activation transforms of each of the tower outputs to determine how much weight to assign to each of the outputs. This improves the model instead of just concatenating both the outputs together [Liu et al., 2021].

We utilize the GTN as a classifier to determine which light curves contain em bedded close binary signals. We work both with simple simulated light curves and the LSST AGN DC light curves with artificially injected close binary signals. We choose the hyperparameters as a batch size of 16, Adam optimizer, and a learning rate of 10^{-4}. Furthermore, we utilize an 8-headed model, with an encoding size of 64 and a hidden dimension of 1024 on the feed-forward network. We also utilize a

relu activation function and a dropout layer of 0.2 on our feedforward network. 39

Chapter 5

Results and Discussion

5.1 Upgrades to the model: Tests on Fiducial Dataset

5.1.1 Old vs New: The improvements with Attention and Latent Space

To test the upgrades to the model, we utilize 100 simulated DRW light curves and our ANP model. When trained on the degraded light curves, we find that the upgrades have indeed improved the performance of the model. From Figure 5.1, it can be seen that the models train very similarly on the light curves. Indeed, it is found that in practice, the neural process models are very powerful and able to model training data to very high precision. It should also be noted that this is a highly challenging test for the models, with most of the data obscured.

The improvements can be seen in Figure 5.2, where the upgraded AttnLNP model can perform better on the validation data better than the CNP. Though the difference might seem small, it should be noted that this is in log space, and the metric measures the average performance of the model throughout the entire light curve. Thus, we can see the addition of attention and a latent space has improved the performance of the model. Furthermore, the addition of a latent space allows for the possibility of generating multiple samples of the light curve predictions.

In Figure 5.3, we plot a test light curve (i.e one never seen by the model during training) to show the power of having a model that can produce different samples. With the addition of multiple samples, the model performs better and the additional

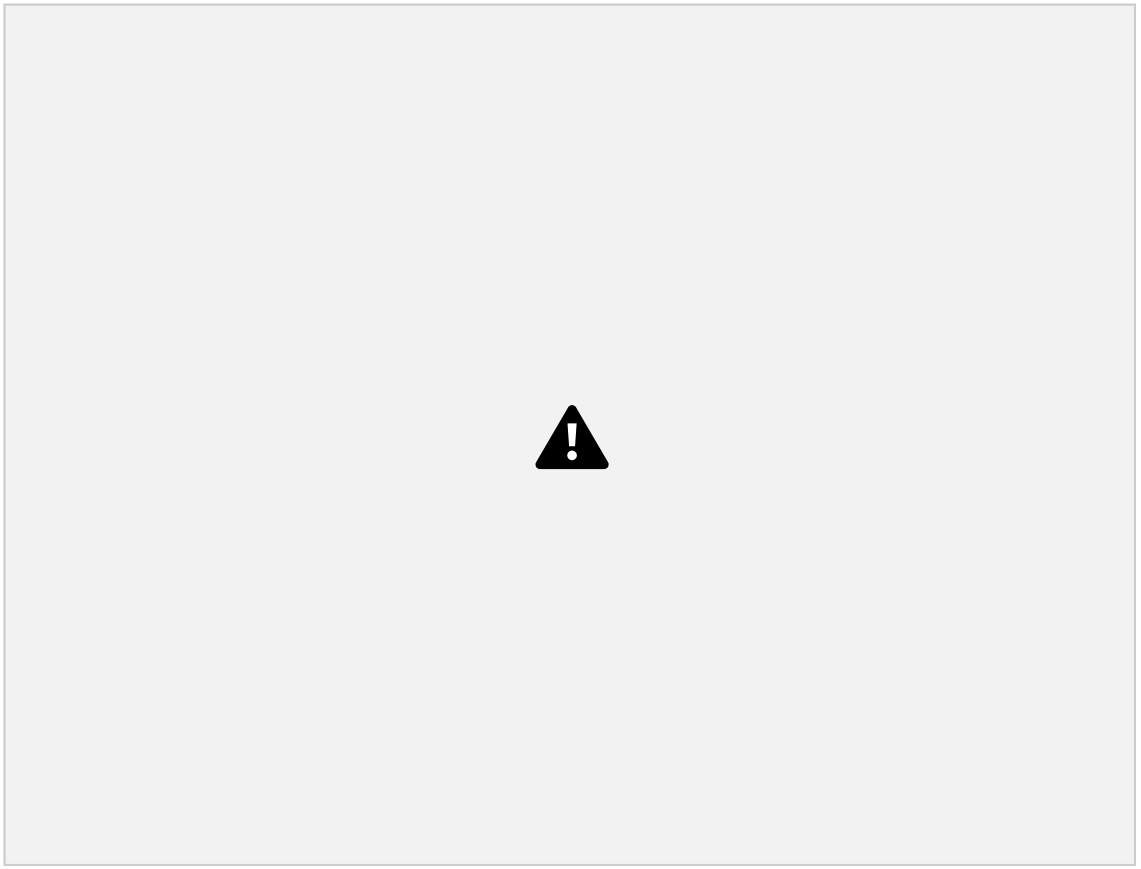


Figure 5.1: Training Curves of the (old) CNP model and the (upgraded) AttnLNP models on fiducial DRW light curves. The CNP model loss is plotted in blue, while the AttnLNP model loss is plotted in orange.

samples can be used for further analysis to analyze better the light curves for periodic signals or reverberation mapping.

5.1.2 Recovery of hidden data

In this section, we look at the performance of the upgraded model on a larger dataset and the ability of the model to recover information that was hidden from it. We utilize 1000 simulated DRWs. Due to the large sample size, we utilize a larger ANP network with 256 nodes in the encoding for both the deterministic and latent paths.

With a larger sample size and training set, the model is able to perform better. The mean negative log probable loss is now 0.58 on the test data with the light curve data with gaps.

When evaluated on the entire data, the model can capture some light curves better than others. We evaluate this performance by calculating the mean negative

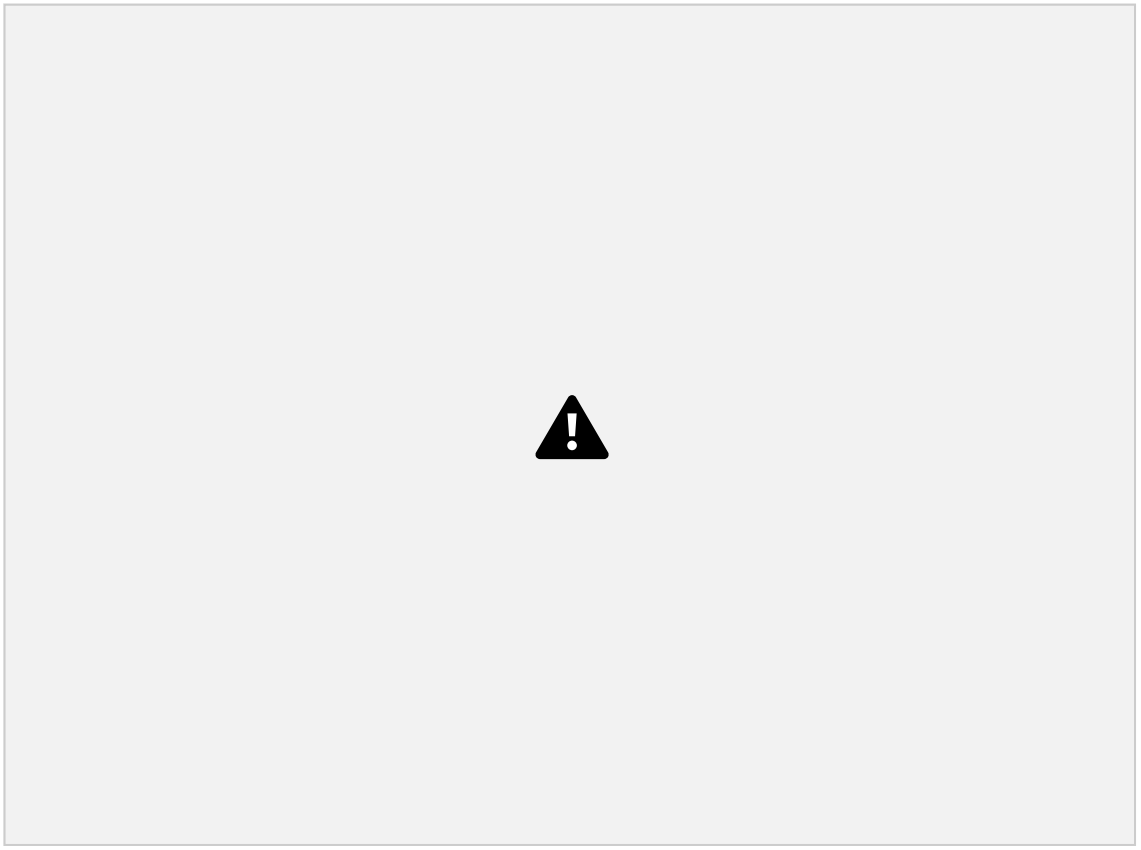


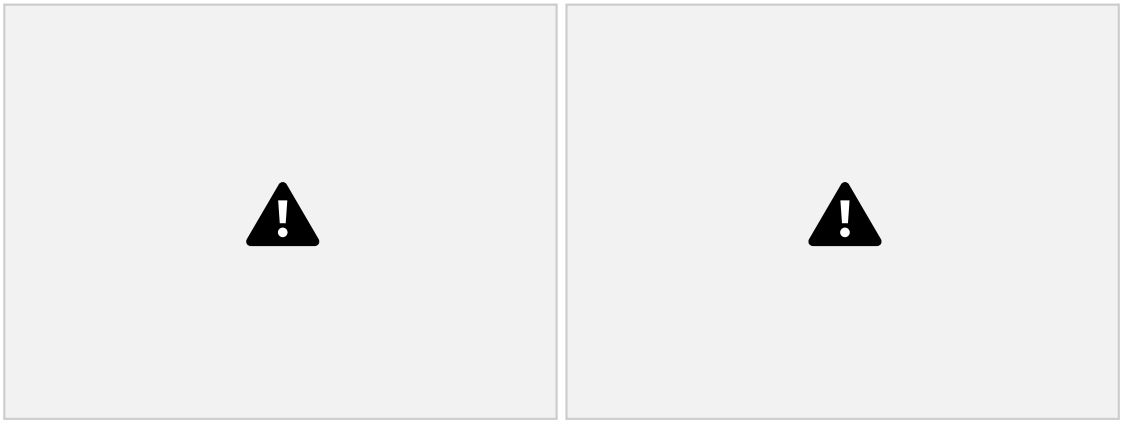
Figure 5.2: Validation Curves of the (old) CNP model and the (upgraded) AttnLNP models on fiducial DRW light curves. Layout same as Figure 5.1

log probable loss of the modeled light curve with the real light curve. A higher value indicated a better match. In Figure 5.4, we plot the 4 best modelled light curves. The model can fit the data well and most of the light curve falls within the confidence interval.

However, there are cases when the model fails to predict the entire light curve. In Figure 5.5, we plot the 4 worst modeled curves. The model fits the data with the gaps well but fails to understand the underlying structure in the areas that it hasn't seen. However, even in these cases, the model still doesn't make extraneous predictions and the predictions are still reasonable without access to the full light curve.

The average loss for the curves is 0.37. Thus, most of them fit well to the data. In Figure 5.6, we plot a light curve that has a loss close to the average value. We see that the model is successful in capturing the entire light curve within the confidence interval of the measurement. Thus, we can conclude that

the model is successful at



(a) CNP modeled Fiducial DRW Light Curve

(b) AttnLNP modeled Fiducial DRW Light Curve with 10 samples of the light curve

Figure 5.3: Representative test Light Curves modeled under the old CNP model (a) and the new AttnLNP model(b). Black dots indicate observations with error bars. The mean modeled light curve is shown with the blue solid line with the shaded confidence intervals of 1,2 and 3σ

reconstructing light curves even from limited context points. In more realistic data with larger amounts of light curves, as well as more observations in the gaps, the model should perform very well in reconstructing observations.

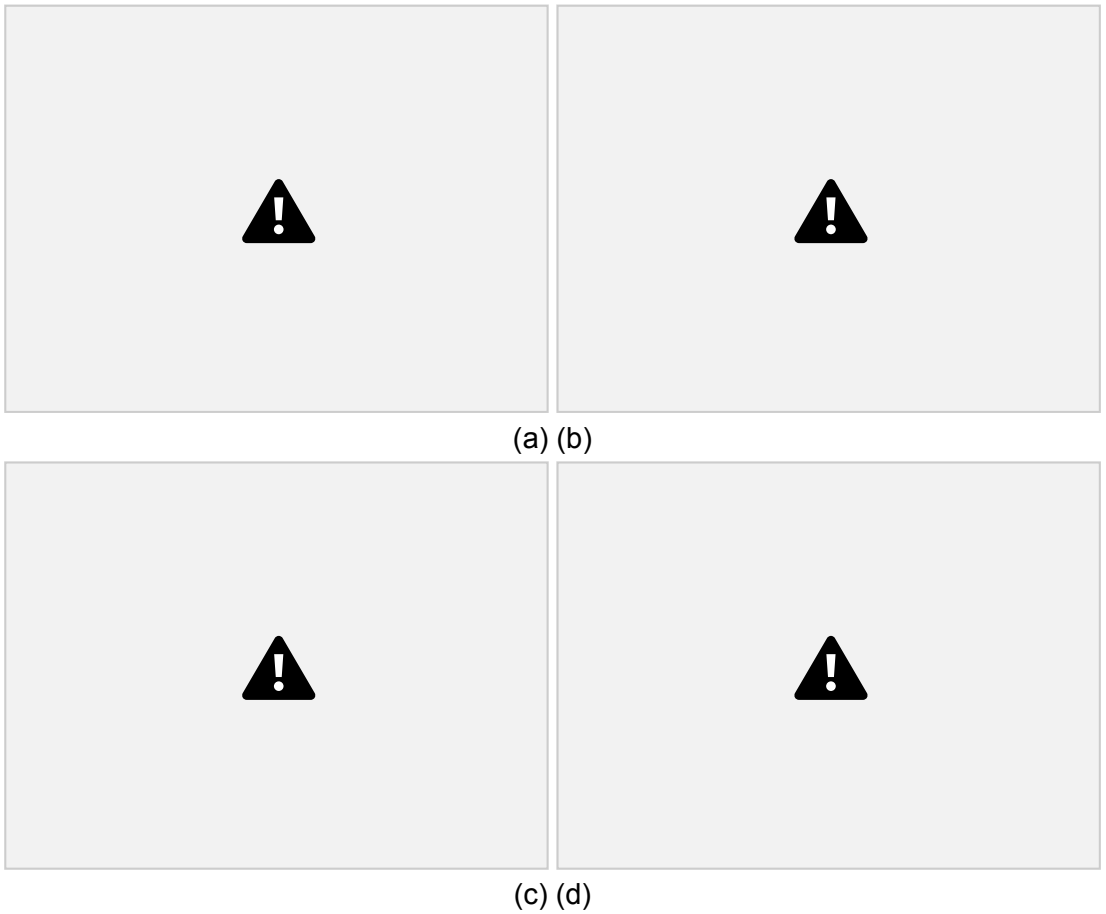
5.2 Recovery of Parameters from Simulated Light Curves

Next, we use the simulated light curves with transfer functions and test the model's effectiveness in the recovery of parameters. Specifically, we would like to recover the transfer function.

5.2.1 Reconstruction of Gaussian Transfer Function

For the Gaussian transfer function light curves, besides the transfer function, the varied parameters are the DRW τ and SF_∞ and the redshift. When training the model, we include the extra MLP to predict the transfer functions, as well as other parameters.

The reconstructed transfer functions for the test light curves can be seen in Appendix A. We include the mean predicted transfer function, along with the actual mean transfer function for each band in Figure 5.7. It can be seen that the model



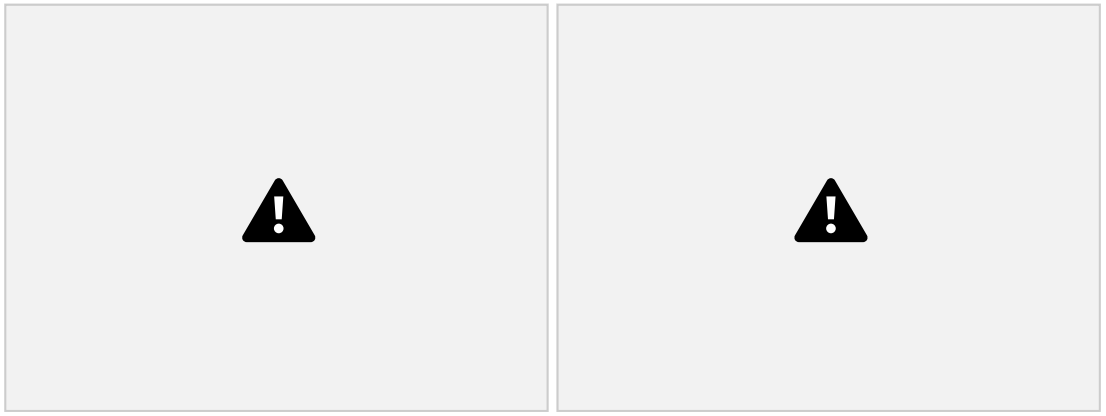
(a) (b) (c) (d)

Figure 5.4: Examples where the model can replicate a fiducial curve well despite only training on data with large gaps. Note the different scales of the y-axis. The full light curve is plotted with blue dots, while the degraded observations that the model sees are marked with orange dots. The modelled prediction of 10 samples are plotted with blue lines with a 2σ confidence interval shaded in blue.

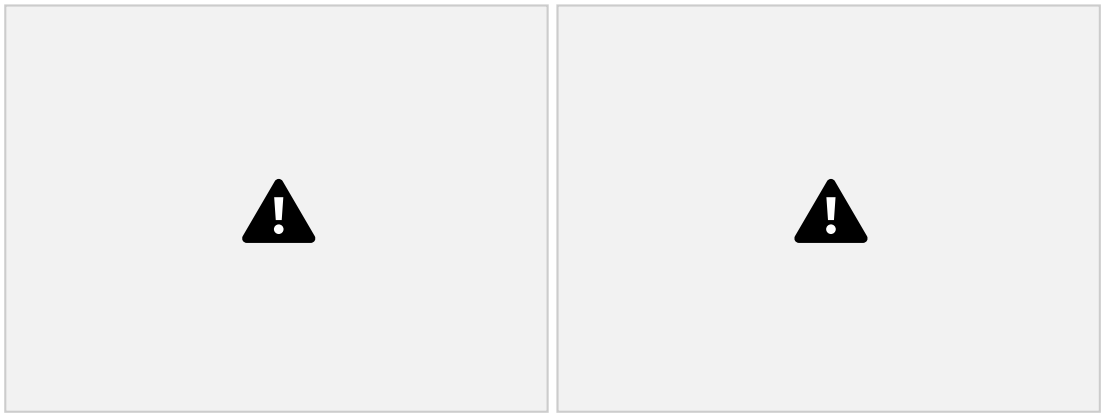
performs exceedingly well as the size of the time lags increases. We include the negative log probable loss and see that it decreases (i.e performs better) with an ascent in the bands. The u-band transfer functions are particularly difficult to model as the lags peak over a small range and vary more over a smaller region than the other bands. However, the model is still able to recover the overall

shape and peak for each of the bands.

We also can see the recovery of parameters in Figures 5.8 - 5.12. From the distribution of mean recovered parameters compared with the true underlying pa rameters, we see that in all bands, the predicted parameters are tightly centered around the center of the distribution. Furthermore, in a few cases (for example,



(a) (b)



(c) (d)

Figure 5.5: Examples where the model encounters problems in replicating the fidu cial curve. Notation same as Fig 5.4

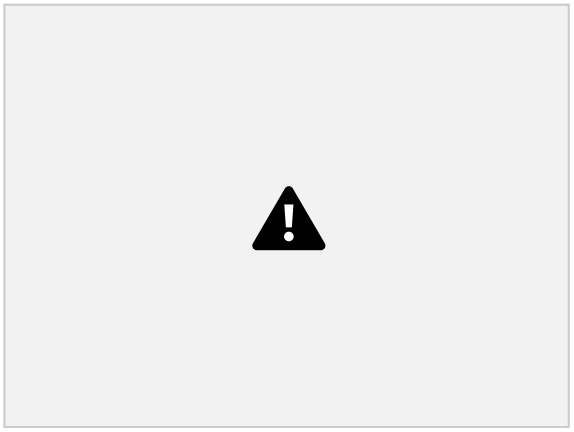
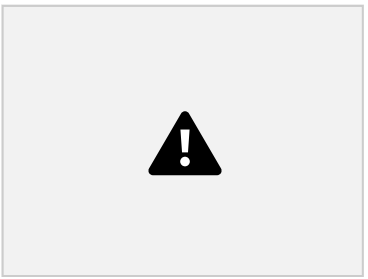


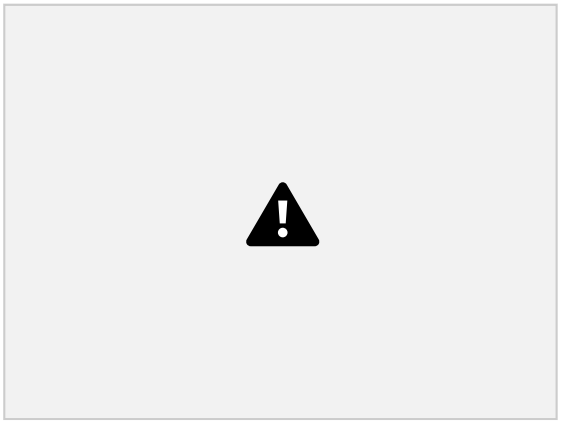
Figure 5.6: Example close to the average goodness of fit of the predicted curve to the actual light curve. Notation same as Fig 5.4

(a) Recovery of the mean transfer function from u band light curves

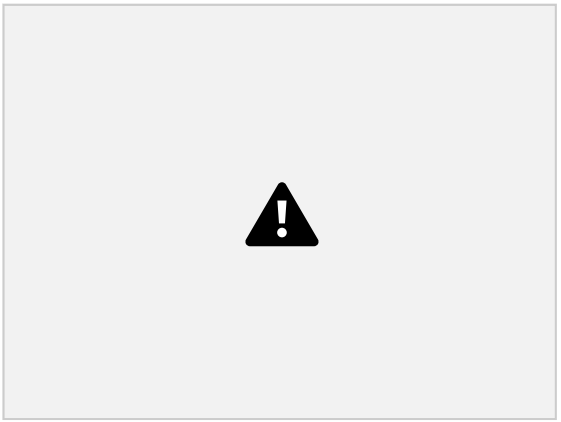
(b) Recovery of the mean transfer function from g band light curves



(c) Recovery of the mean transfer function from r band light curves



(d) Recovery of the mean transfer function from i-band light curves



(e) Recovery of the mean transfer function from z-band light curves

Figure 5.7: Recovery of mean transfer functions from light curves of different bands simulated with a Gaussian transfer function. The real light curve is plotted

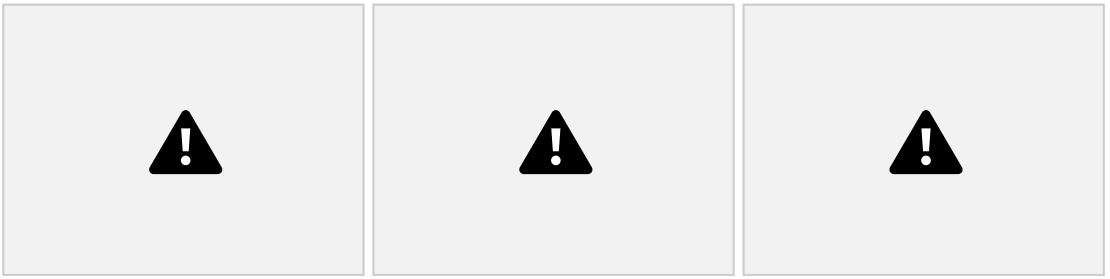
in orange, while the modelled transfer function is plotted in blue with a 2σ confidence interval shaded in blue.

Figure 5.8a and Figure 5.9b), the distribution can model peaks in the parametric distribution. This matches the findings of Park et al. [2021], where the parameters associated with the black holes were recovered tightly centered around the original distribution.

Thus, we can conclude from this particular transfer function that the model is able to parse out the transfer function well when there is a high degree of similarity among transfer functions. Our SOM stratifies light curves by topographical similari ties, which could include similarities from transfer functions. Thus, the stratification of light curves could group light curves with similar transfer functions in real large datasets like the LSST catalog, allowing for easy detection of transfer functions.

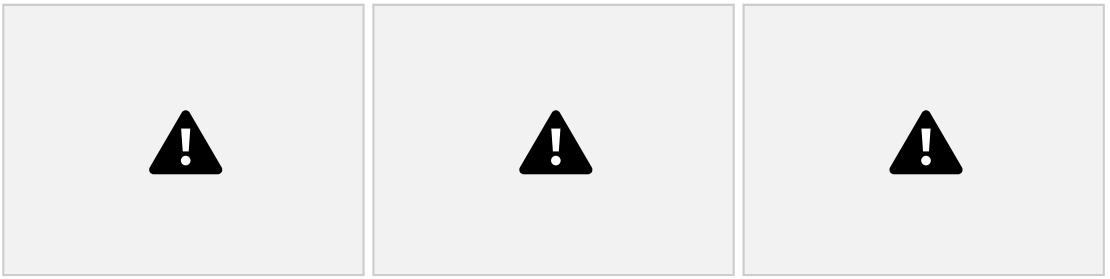
(a) Recovery of the DRW τ parameter (b) Recovery of the DRW SF_∞ parameter (c) Recovery of the redshift

Figure 5.8: Recovery of parameters from the u band light curves simulated with a Gaussian transfer function. The actual distribution of parameters is plotted in orange, while the distribution of predicted mean for each light curve is plotted in blue.



(a) Recovery of the DRW τ parameter. (b) Recovery of the DRW SF_∞ parameter (c) Recovery of the redshift

Figure 5.9: Recovery of parameters from the g band light curves simulated with a gaussian transfer function. Notation same as Figure 5.8



(a) Recovery of the DRW τ parameter (b) Recovery of the DRW SF_∞ parameter (c) Recovery of the redshift

Figure 5.10: Recovery of parameters from the r band light curves simulated with a gaussian transfer function. Notation same as Figure 5.8

(a) Recovery of the DRW τ parameter (b) Recovery of the DRW SF_∞ parameter (c) Recovery of the redshift

Figure 5.11: Recovery of parameters from the i band light curves simulated with a gaussian transfer function. Notation same as Figure 5.8

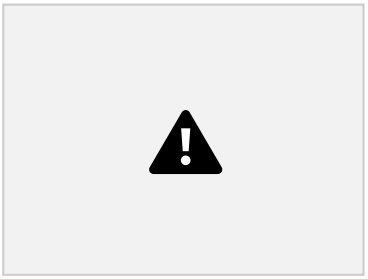
(a) Recovery of the DRW τ parameter (b) Recovery of the DRW SF_∞ parameter (c) Recovery of the redshift

Figure 5.12: Recovery of parameters from the z band light curves simulated with a gaussian transfer function. Notation same as Figure 5.8

5.2.2 Reconstruction of the Cackett Transfer Function

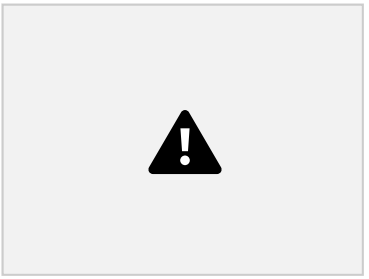
With the Cackett transfer function, we can recover three additional parameters related to the black hole. These are the black hole mass, the Eddington luminosity ratio, and the inclination of the black hole to the observer. These parameters also induce variability in the transfer functions, making this an additional challenge for the model. We train the model similar to the previous section with these additional parameters.

The mean transfer functions for each band can be seen in Figure 5.13 and the reconstructed transfer function for each curve is seen in Appendix B. The model can recover the transfer function better for lower bands in this case. This is because the lower bands have a lower variation of time lags. The higher bands also extend into much higher time lag values, smearing out the convoluted light curve more, causing recovery of parameters from light curves with gaps to be tougher and a longer baseline sampling needed to identify variability trends. However, the model

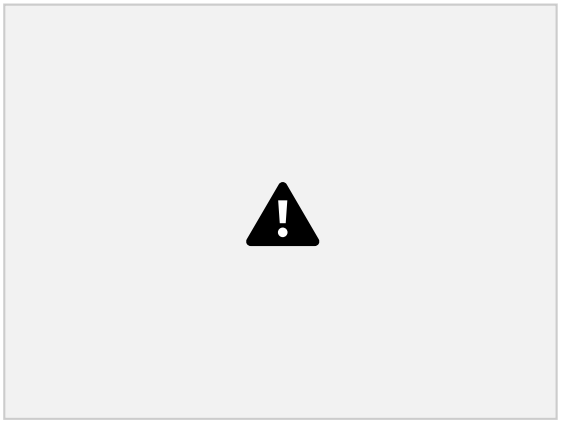


(a) Recovery of the mean transfer function from u band light curves

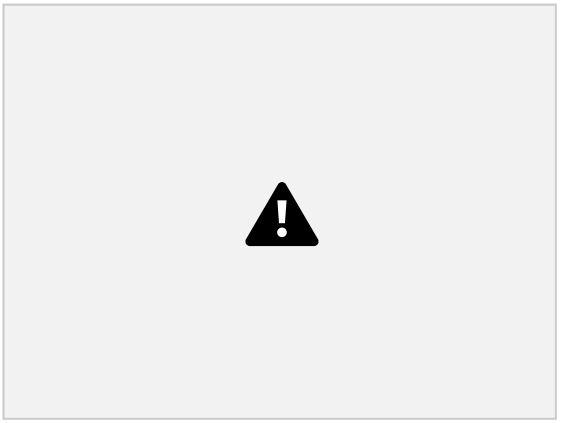
(b) Recovery of the mean transfer function from g band light curves



(c) Recovery of the mean transfer function from r band light curves



(d) Recovery of the mean transfer function from i-band light curves

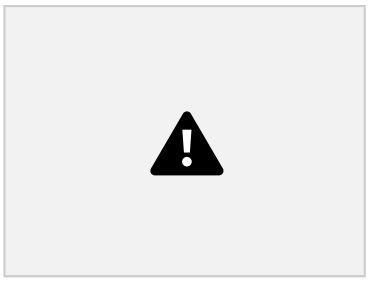


(e) Recovery of the mean transfer function from z-band light curves

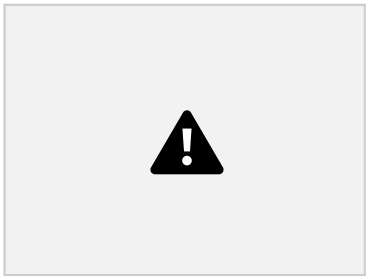
Figure 5.13: Recovery of mean transfer functions from light curves of different bands simulated with a Cackett (Thin-Disk approximation) transfer function. Notation same as Fig 5.7

is still able to capture the inflection point for even the most poorly fit transfer functions well, thus capturing the peak of the transfer function very well. In Figures 5.14 - 5.18, the parameter recovery distribution for the Cackett trans fer function light curves is shown. Once again, the predicted distribution tends to be centered around the peak of the actual distribution. However, distinct peaks can still be recovered as in Figures 5.16d and 5.15f where the predicted distribution has multiple peaks.

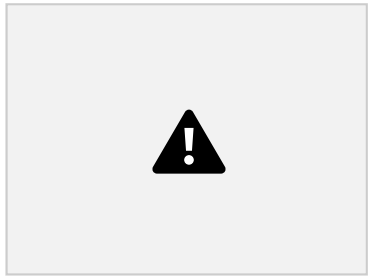
With this, we have provided a means of estimating a transfer function and other black hole parameters from limited data points. We can train different models on real transfer functions recovered through photometric reverberation mapping and produce different estimates for transfer functions for light curves in the LSST.



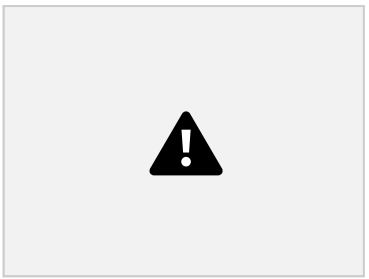
(a) Recovery of the DRW τ parameter



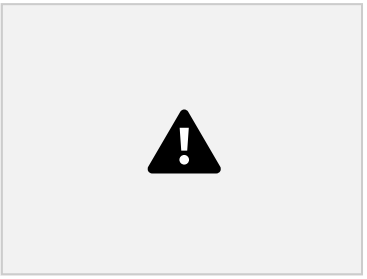
(b) Recovery of the DRW SF_∞ parameter



(d) Recovery of the (log) Black Hole Mass (in $M_\odot$)



(c) Recovery of the redshift



(f) Recovery of the inclina tion of the black hole (in ra dians)

(e) Recovery of the Edding ton luminosity ratio

Figure 5.14: Recovery of parameters from the u band light curves simulated with the Cackett (Thin Disk approx) transfer function. Notation same as Figure 5.8

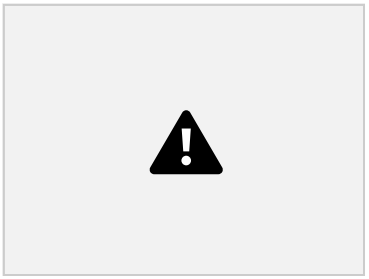
5.2.3 Recovery of the Parameters after training

We have demonstrated that our model can recover parameters with an MLP added during training. However, we can also investigate if the model is capable of learning these parameters without interference during training. To achieve this, we simply detach the MLP that predicts the parameters and allow the model to train with a focus on solely reconstructing the light curve. Once the model has been trained, we utilize a method similar to Tachibana et al. [2020]. We obtain hidden representations of the curve during test time by averaging the representation obtained through cross attention and one latent space sample across all target points. Then, we pass this hidden representation through an MLP with one hidden layer that outputs either a single parameter or a function (which we compare to the transfer function).

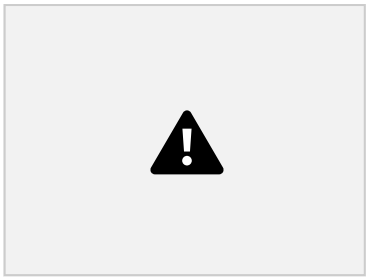
Unlike Tachibana et al. [2020], we utilize the Bayesian nature of NPs to train this MLP. We output a mean and standard deviation of the parameter and optimize the model to minimize the negative log probable loss of this distribution with respect to the parameter of interest. For this task, we utilize 100 simulated u-band light curves with the Cackett Transfer function and try to reconstruct both the transfer

(a) Recovery of the DRW τ parameter

(b) Recovery of the DRW



(f) Recovery of the inclina



SF_∞ parameter

(c) Recovery of the redshift

Black Hole Mass (in $M_\odot$)

(e) Recovery of the Edding ton luminosity ratio

tion of the black hole (in ra dians)

(d) Recovery of the (log)

Figure 5.15: Recovery of parameters from the g band light curves simulated with the Cackett (Thin Disk approx) transfer function. Notation same as Figure 5.8

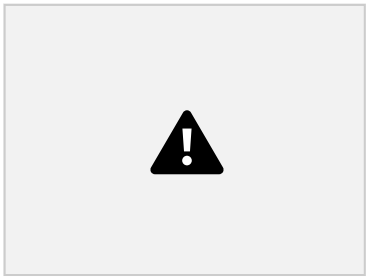
function and the aforementioned parameters. We ran the MLP for 2000 epochs with an early stopping patience criteria of 200 epochs. We use a hidden layer halfway between the size of the transfer function and the latent space size, thus with 314 nodes. We choose a learning rate of 10^{-5}.

In Figure 5.19, we see that the model trains extremely quickly and can achieve a very low loss on the transfer function. Surprisingly, the model performs better on validation data than training data. This is perhaps because of the greater diversity of the training data as well as the fact that the model can capture the trends very well.

The model is able to perform well on test data as well and with a mean negative logprobloss of -1.09. The great performance of the model in recovering the transfer function suggests that the transfer function is indeed hidden within the the model's representation of the light curve. We plot the recovered transfer functions from the training curves in the Appendix C.

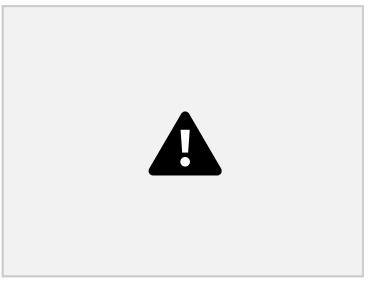
We also recover other physical parameters. For parameters that have a higher range like the mass and tau, we utilize a learning rate of 10^{-4}, while we utilize 10^{-5}

(a) Recovery of the DRW τ parameter



(b) Recovery of the DRW SF_∞ parameter

(d) Recovery of the (log) Black Hole Mass (in $M_\odot$)



(f) Recovery of the inclination of the black hole (in radians)

(c) Recovery of the redshift

(e) Recovery of the Eddington luminosity ratio

Figure 5.16: Recovery of parameters from the r band light curves simulated with the Cackett (Thin Disk approx) transfer function. Notation same as Figure 5.8

for other parameters like inclination, SF_∞, and redshift that lie in a smaller range of values. For the eddington luminosity ratio, we find that the learning rate does

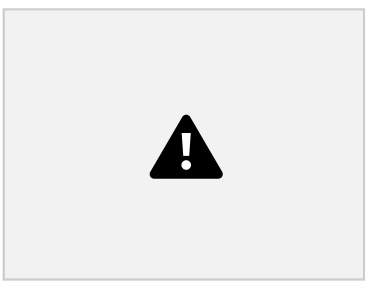
not make a difference. We also run the model for 1000 epochs with a early stopping patience of 100.

We plot the parametric distribution of parameters in the Appendix C and find we recover the same behavior of recovering parameters tightly centered around the actual distribution's center.

Thus, we find an alternative method for recovering parameters and the transfer function without biasing the training of the Neural Processes. The model's success in recovering these parameters from the hidden state lends credence to the assumption that the model can learn driving factors of variability. This method also allows for different models of transfer functions to be tested on real data after reconstruction. Furthermore, the parameters can be recovered much more flexibly without altering the structure and loss of the NP with this method.

(a) Recovery of the DRW τ parameter

(b) Recovery of the DRW SF_∞ parameter



(c) Recovery of the redshift

(d) Recovery of the (log) Black Hole Mass (in $M_{\odot}$)

(e) Recovery of the Eddington luminosity ratio

(f) Recovery of the inclination of the black hole (in radians)

Figure 5.17: Recovery of parameters from the i band light curves simulated with the Cackett (Thin Disk approx) transfer function. Notation same as Figure 5.8

5.3 LSST AGN DC

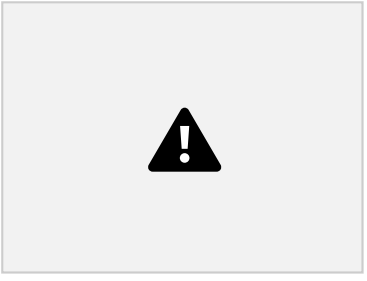
In this section, we present our results on the LSST AGN Data Challenge.

5.3.1 Clustering with SOM

In Kovačević et al. [2023], the SOM was tested on the LSST AGN DC. We present an upgraded version of the SOM from the previous study and results from different clustering styles.

Utilizing cleaned u-band curves, we obtain different clustering configurations for different hyperparameters of the SOM. We use u-band curves as the variability is thought to be closer to the driving variability of the quasar (Refer back to the convolution of different transfer functions with the light curve).

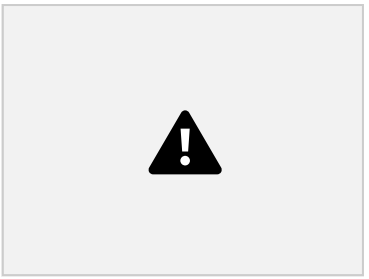
Running a 6 × 6 SOM for 10,000 epochs, we found that the hyperparameters σ = 1.5 and a learning rate of 0.05 produced the best combination of quantization and topographic errors. In Figure 5.20, we can see that the learning rates of 0.05 and 0.1 both produce similar Quantization errors but 0.05 has a lower topographic error.



(a) Recovery of the DRW τ parameter

(b) Recovery of the DRW SF_∞ parameter

(d) Recovery of the (log) Black Hole Mass (in $M_\odot$)



(f) Recovery of the inclination of the black hole (in radians)

(c) Recovery of the redshift

(e) Recovery of the Eddington luminosity ratio

Figure 5.18: Recovery of parameters from the z band light curves simulated with the Cackett (Thin Disk approx) transfer function. Notation same as Figure 5.8

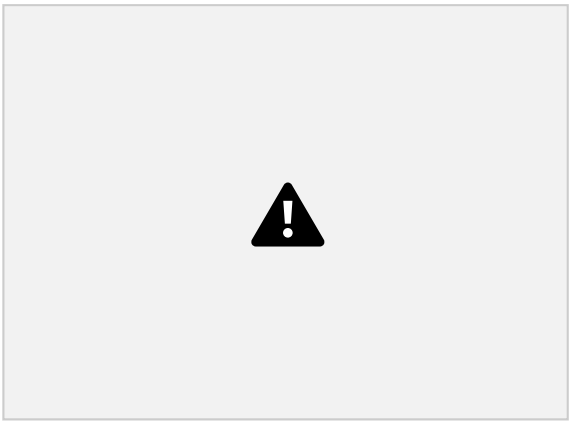
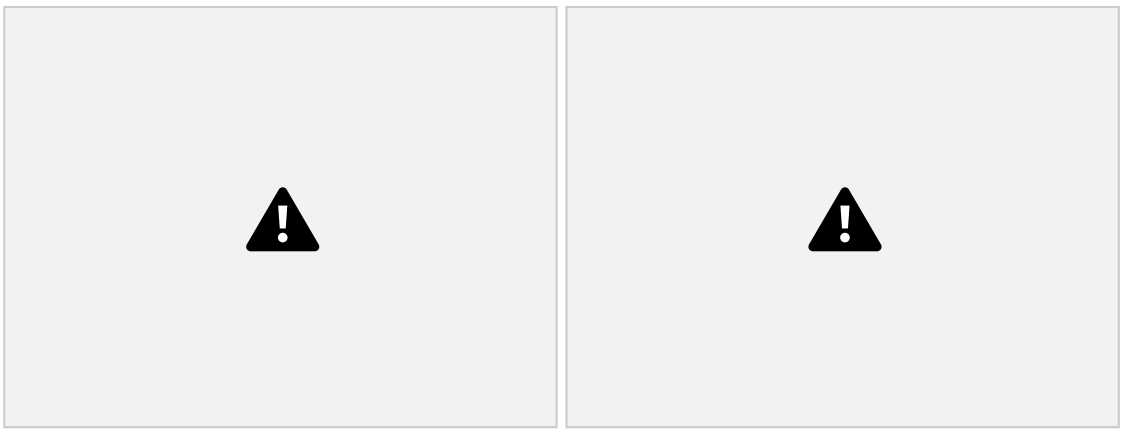
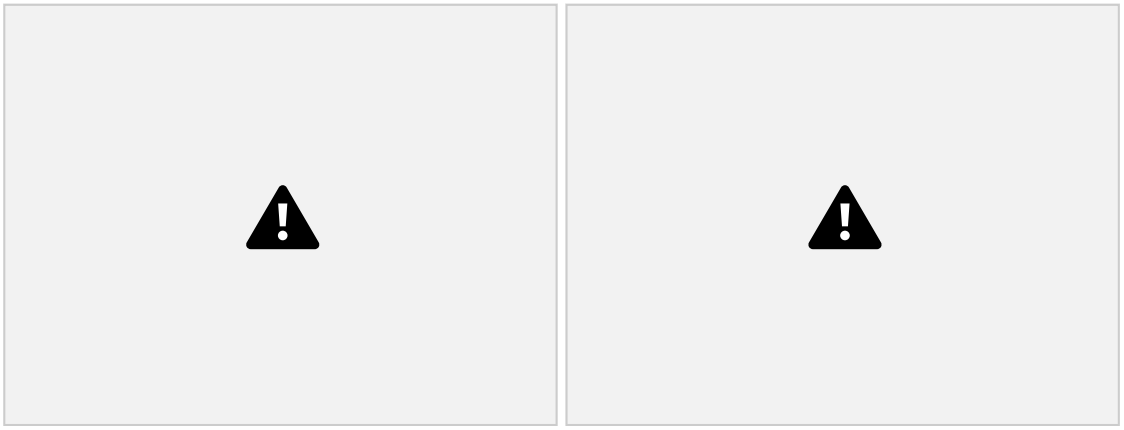


Figure 5.19: The training and validation curves for the MLP recovering the transfer function from a hidden representation of simulated cackett light curves in the u band. The training loss is plotted in blue, while the validation loss is plotted in orange.



(a) Quantization Error (b) Topographic Error

Figure 5.20: Learning Curves for LSST AGN DC light curves for different learning rates at constant σ = 1.5. The learning rate is varied between 0.01, 0.05, 0.1 and 0.5 and the error is plotted with lines in blue, orange, green and red, respectively.



(a) Quantization Error (b) Topographic Error

Figure 5.21: Learning Curves for LSST AGN DC light curves for different sigma values but constant learning rate of 0.01. The sigma is varied between 0.5, 1.0, 1.5 and 2.0 and the error is plotted with lines in blue, orange, green and red, respectively.

In Figure 5.21, we see that 1.5, 1.0, and 0.5 produce similar values quantization errors for when fully trained, but 1.5 also produces a much lower topographic error. Now, we choose the size of the SOM grid. The accepted convention is to set the grid size equivalent to five times the square root of the dimensionality of the data. We found this to produce too many clusters with too few curves for a dataset like the LSST AGN Data Challenge, so we set our default to simply the square root of the dimensionality. However, we would also like to empirically select the optimum grid size. For simplicity, we require the grid to be square and test various square

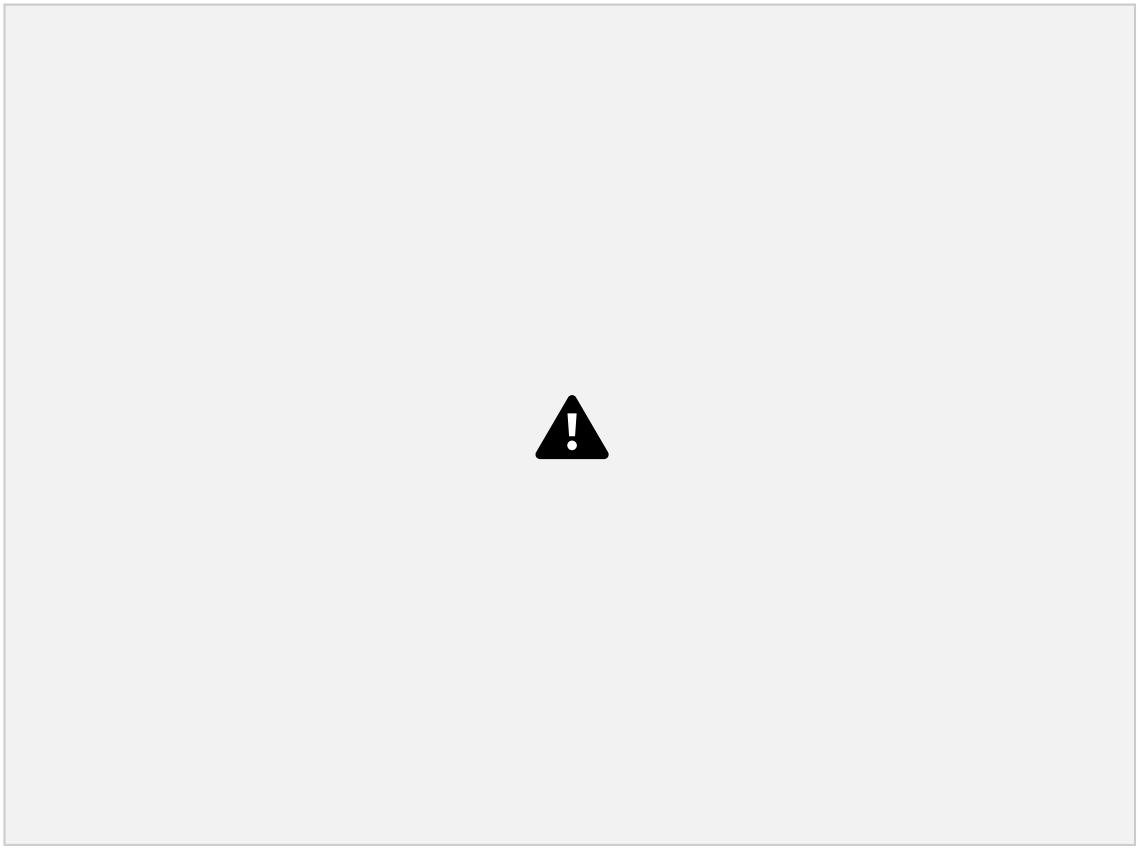


Figure 5.22: Quantization and Topographic Errors for different SOM grid sizes for the LSST AGN DC. The final quantization error for different grid sizes is plotted in blue, while the corresponding topographic error is plotted in orange.

grid lengths to find the one with the best quantization error at the end of 10,000 epochs with the aforementioned hyperparameters for our dataset. In Figure 5.22, we see that the quantization error decreases for larger grid sizes, but the topographic error continues to increase for larger grid sizes. A bigger topographic error indicates more random clustering, so it is advantageous to optimize both the quantization and topographic error. Thus, we choose a SOM grid size of 12 × 12.

When we cluster with a 12×12 SOM, we produce 144 distinct clusters. However, we end up with too many separate clusters. The maximum number of light curves within a cluster is only 25. Thus, the SOM has found intricate differences but has stratified too much. Thus, we use gradient cluster mapping to create meaningful clusters from the SOM nodes. In Figure 5.23, we have plotted the distance map of the SOM. The darker the node, the further away the node is from its neigh bors. Creating gradients on the distance map, we can apply our gradient clustering algorithm to group together nodes.

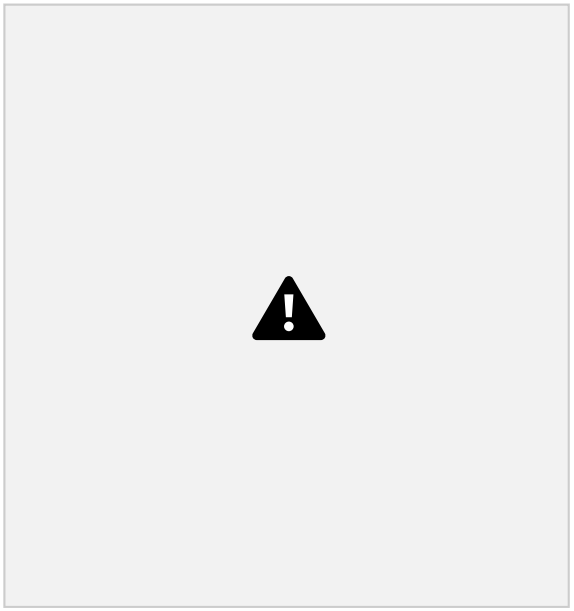


Figure 5.23: The Distance Map for a 12 × 12 SOM on the LSST AGN DC light curves. The darker the grid, the further away the SOM node from the neighboring nodes.

In Figure 5.24, we see the gradients. Each cluster center is chosen at the center of each of the gradients. Every node that lies along one of the lines originating from the cluster centers is grouped into that cluster. Thus, we reduce the number of clusters from 144 to a much more manageable 11 clusters. We can see the distribution of light curves in each cluster in Figure 5.25.

We also plot the light curves that make up each cluster in Figure 5.26. It should be noted that the averaged light curves are created through barymetric time warping, which accounts for the fact that each of the light curves could be shifted in time [Petitjean et al., 2011, 2014; Forestier et al., 2017]. Thus, we can detect the presence of characteristic flares and dips within each of the clusters through the average light curve.

Thus, we demonstrate that the SOM can effectively cluster on large datasets, while providing a range of hyperparameters and metrics that can be used to adapt to the particular dataset of interest. Now, similar to Kovačević et al. [2023], we focus on a single cluster and model the light curves within it.

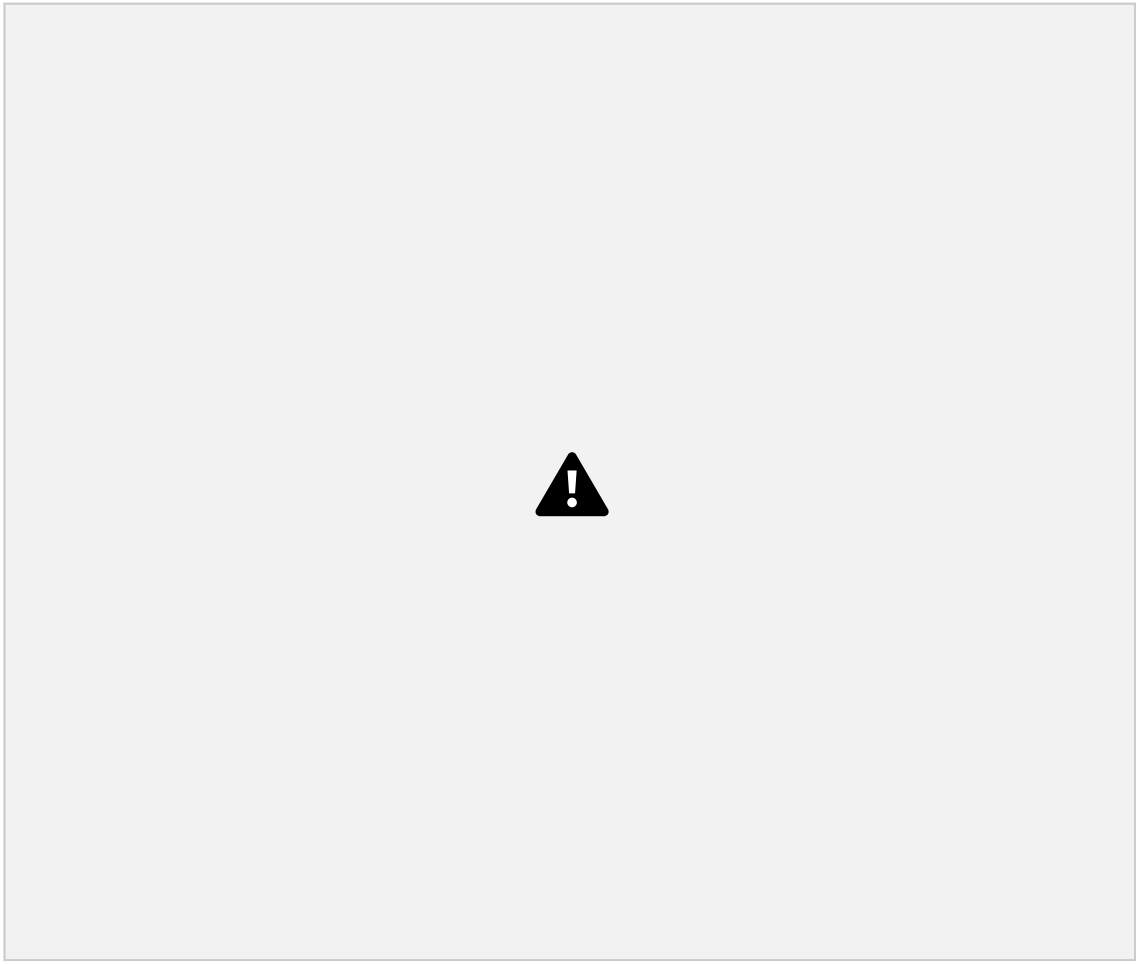


Figure 5.24: The Starburst gradient clusters from a 12×12 SOM on the LSST AGN DC light curves. Nodes that are grouped together lie along black lines that lead to the same central node. The distance map is shaded, with darker nodes being closer to their neighbours.

5.3.2 Analysis of One Cluster

For analysis, we choose Cluster 7 as it is the largest cluster. We run the model for 2000 epochs. However, we allow the model to stop early if the validation data does not improve for 500 epochs.

Our results can be seen in the appendix in Figure D.1. We see that in most cases, the model can stay within the observed light curve, providing a good

reconstruction of the light curve. The removal of outliers as compared to Kovačević et al. [2023] has changed the modeling. While the loss associated with the light curves is slightly larger, the absence of large outliers allows the models to learn finer variations in the light curve and produce reconstructions with more variability. This cleaning

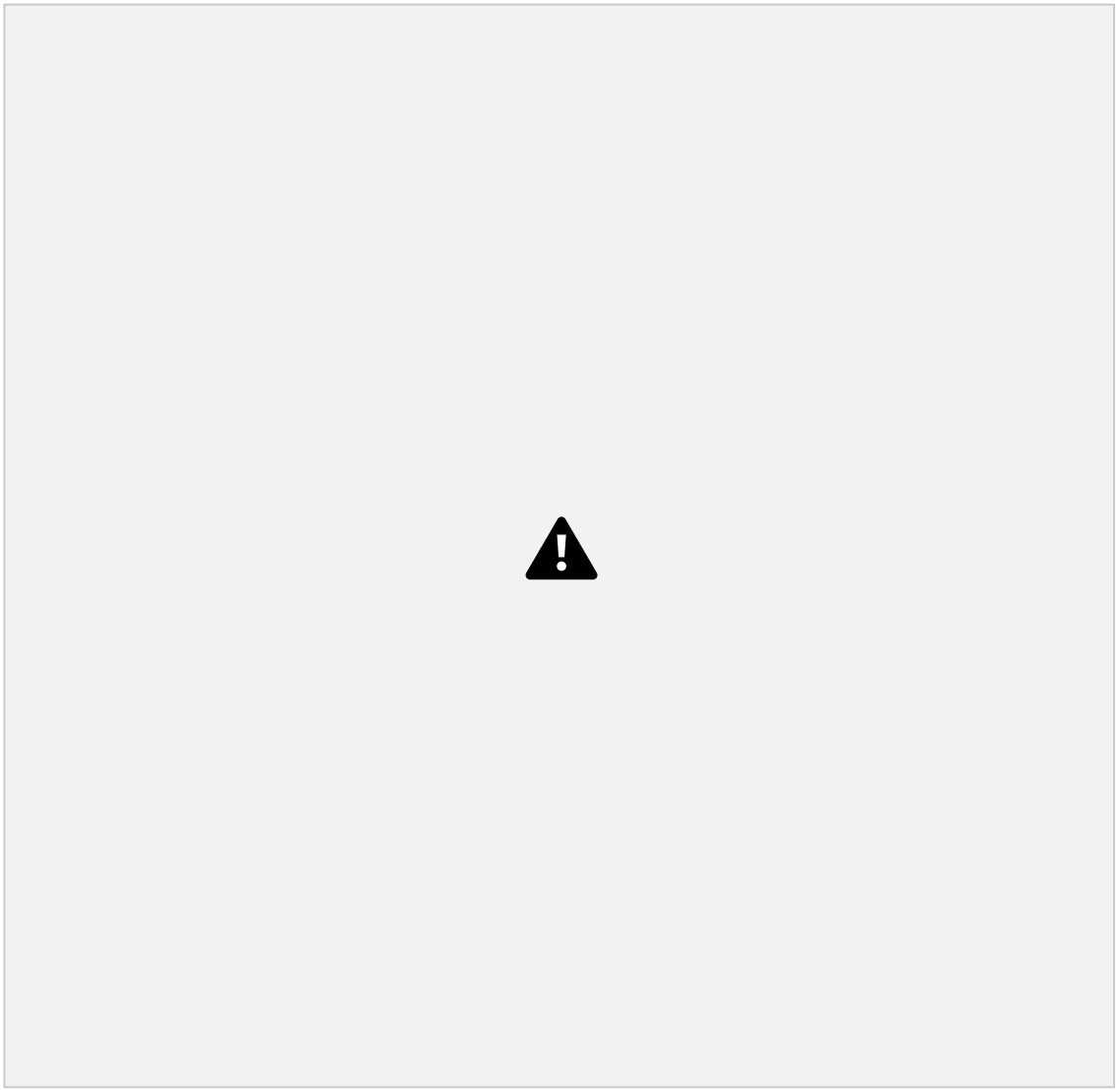


Figure 5.25: The distribution of light curves from the LSST AGN DC Clusters

of outliers is important as it impacts the scaling of the light curves between the maximum and minimum of the light curve.

In Figure 5.27, we see the training and validation curves. We see that the model is able to learn the best performance on the validation loss very quickly

and then starts overfitting to the test data. Thus, the model stops early at epoch 675 (though we plot the loss until epoch 1175.

We also average the mean representation of the light curves within the cluster and feed it through a decoder to predict the average behavior of the cluster. We plot this average behavior in Figure 5.28. We see that there are two characteristic bumps within the light curve, along with a generally decreasing trend. Through this

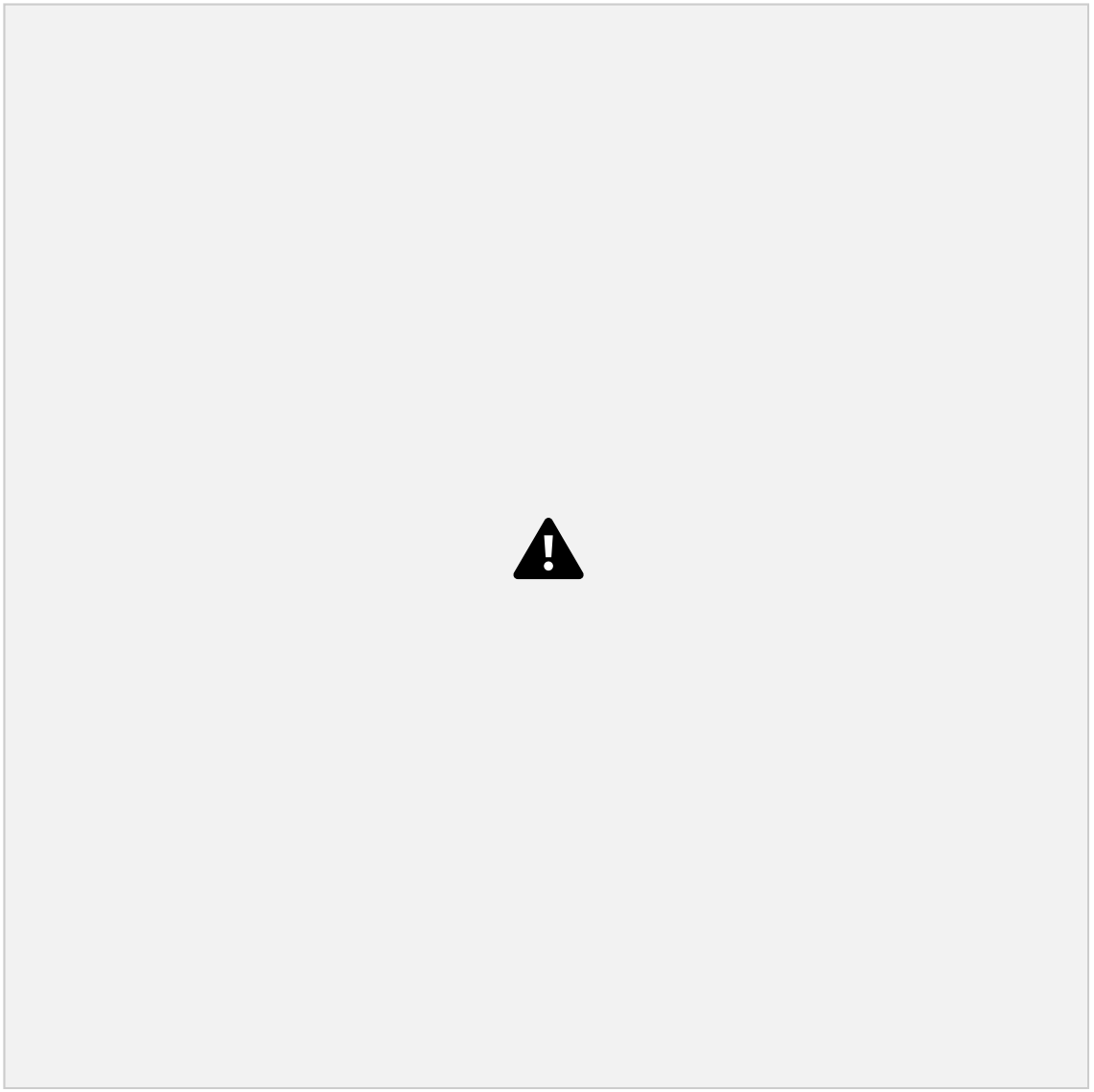


Figure 5.26: The distribution of light curve within each cluster from the LSST AGN DC. The blue curve is the time-warped average while each of the grey curves is an actual light curve

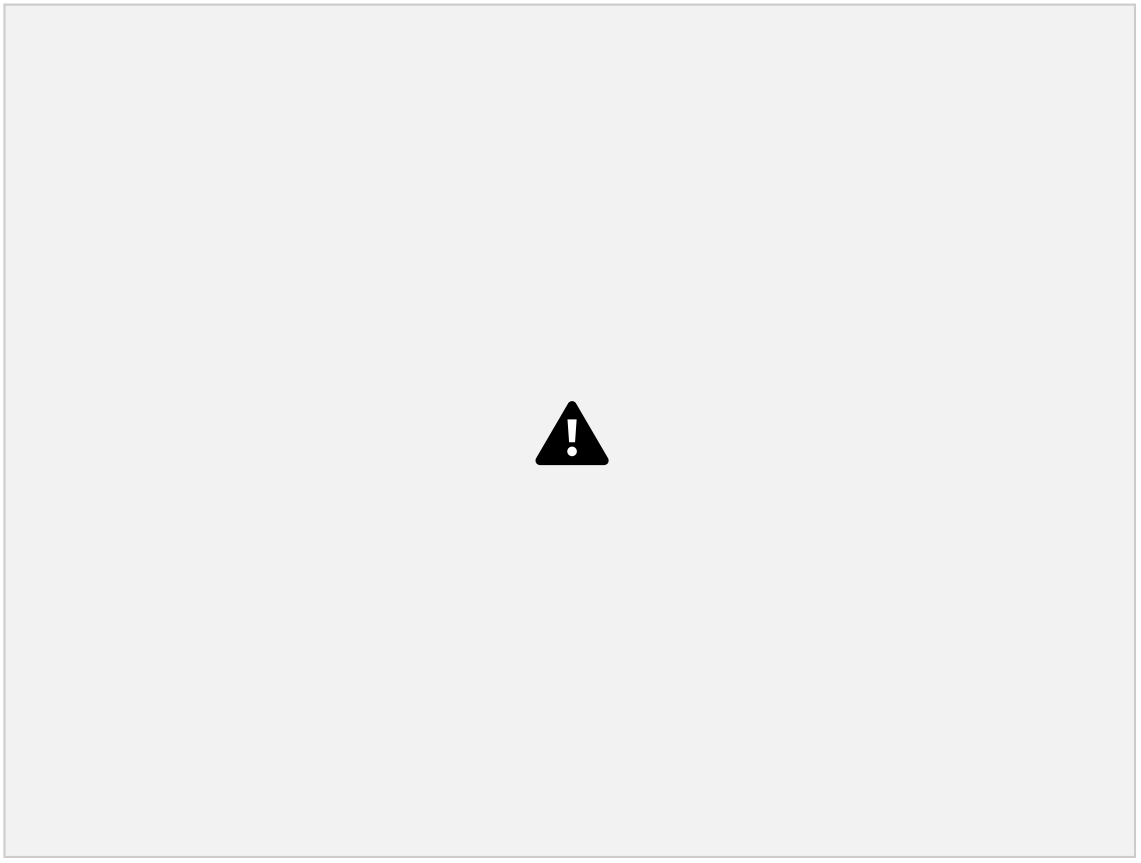


Figure 5.27: The training (plotted in blue) and validation losses (plotted in orange) while training on one cluster from the LSST AGN DC.

method, we can analyze the insights that the model discovers within each cluster.

5.3.3 Multi-Band Modelling

With data from 5 different bands available within the LSST AGN DC, we test the possibility of modeling across different bands. We choose cluster 5 as it contains fewer bands, allowing for easier modeling of 5 separate bands together. We model this cluster in the u-band twice, once when trained on only the u-band data and again when trained on all of the bands in parallel. We use the same model conditions as the last section.

In Figure 5.29, we see that training on multiple bands performs much better with the loss. The multi-band training and validation curves both perform far better on multi-band data than single-band data.

In Figure 5.30, we see the test light curve reconstructions obtained from training on just the u band data. In Figure 5.31, we see the reconstructions from

all the

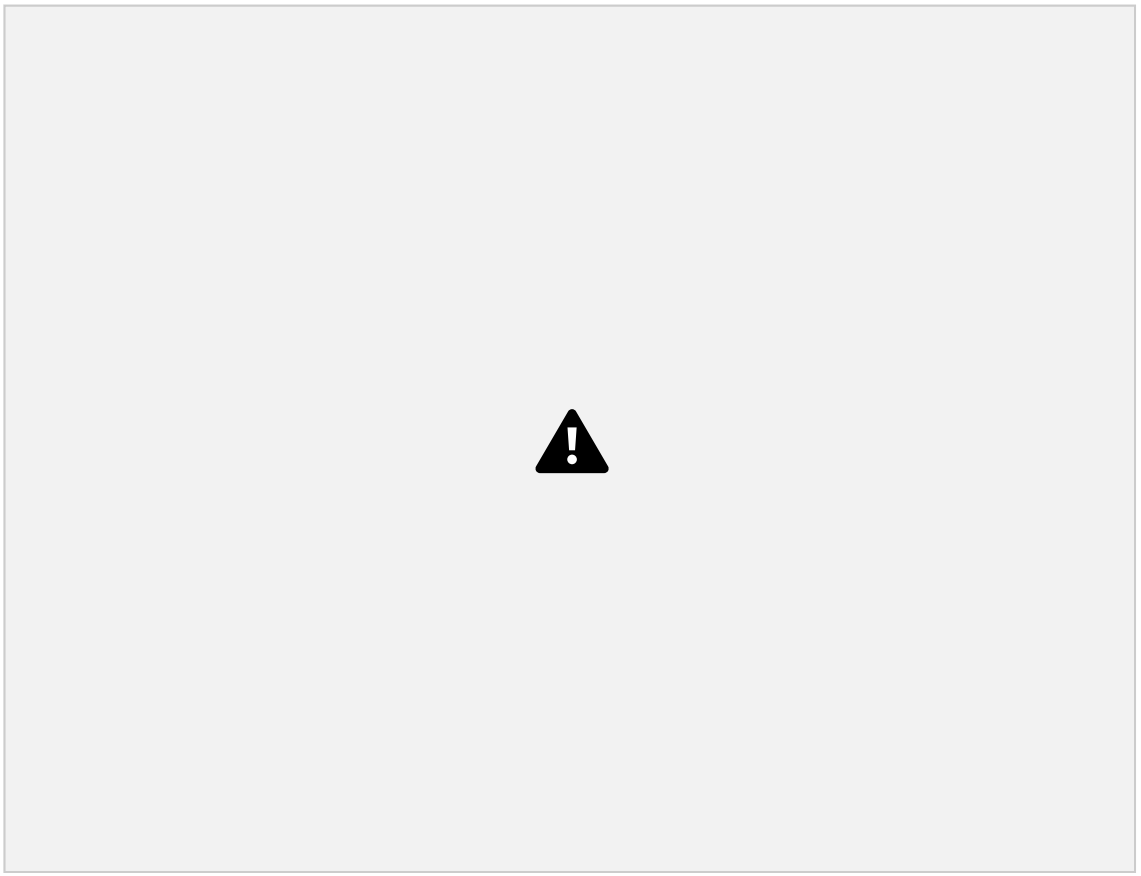


Figure 5.28: The decoded average representation of the light curves within the cho sen LSST AGN DC cluster is plotted with a blue line (Cluster 7). The different shaded bands are the confidence intervals (1,2 and 3 σ respecively) of the represen tation

light curves in Cluster 5 across bands. We see that the multi-band light curves can make better inferences between data points with more variability. Thus, the model is able to learn across different bands to model single-band data.

In Figure 5.32, we identify two light curves that were used in training across both models and compare the reconstruction in both cases. We find that multi band data can induce more variability into the curve again as seen in light curve 1418215. However, it can cause the reconstruction to be less smooth as seen with the 146636. Furthermore, it is important to note that the multiband data increases the uncertainty of the modelling.

Our model's treatment of multi-band data sets it apart from other models. In other works such as Fagin et al. [2024] and Park et al. [2021], all of the bands

are provided as features of the data to the model. This is advantageous as simultaneous

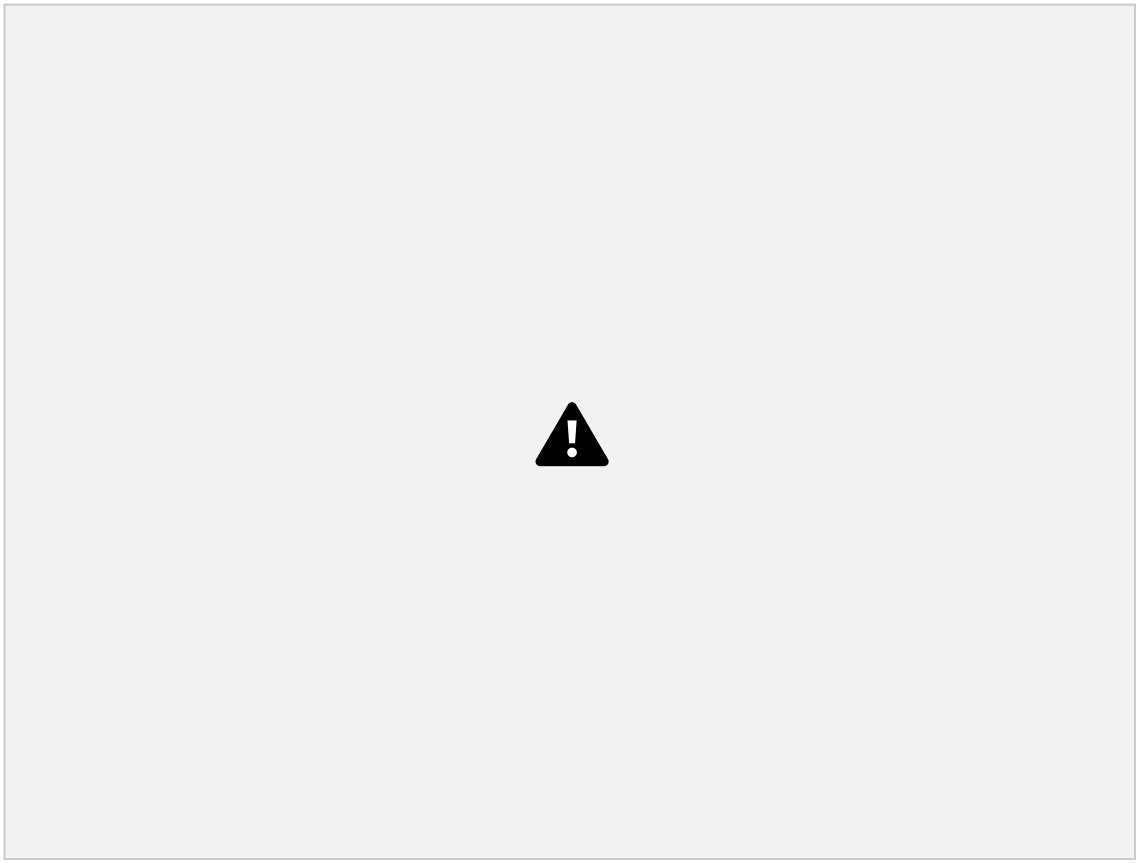


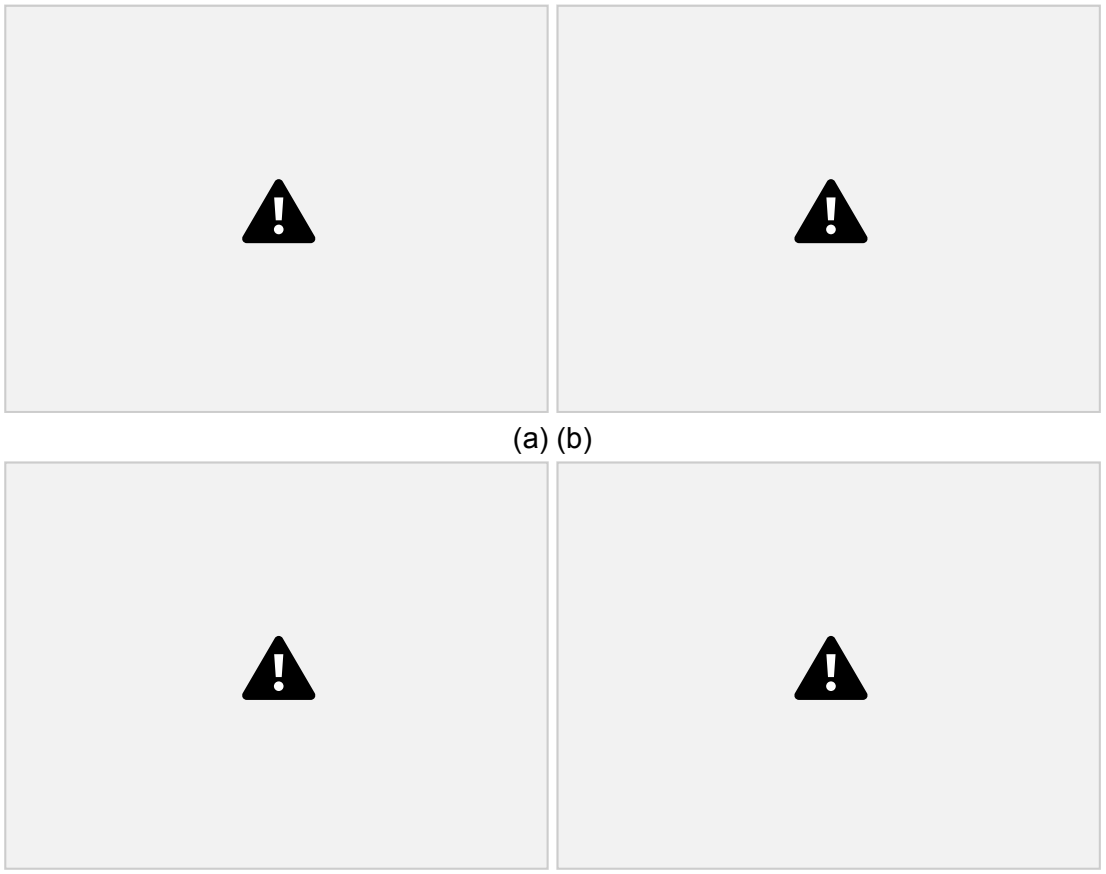
Figure 5.29: Training Curves for just u band as compared to multi-band modelling. The training and validation curves are plotted in blue and orange respectively for the model trained exclusively on u band light curves, while they are plotted in green and red respectively for the model trained on light curves from all bands.

modelling of the different bands can be achieved faster and the model can parse out different correlations between bands. However, in our model, we decouple the different bands and provide each band as a separate light curve. This allows for flexible reconstruction of different bands and the ability to learn subtle trends within each band without bias. Thus, each approach has its own merits.

We include the modelled test light curves across bands in the appendix in Figure D.2.

5.4 ZTF Light Curves

We repeat the same modelling performed above on one cluster of g-band ZTF light curves. We utilize the g-band as it contains the median number of observations



(a) (b)

(c) (d)

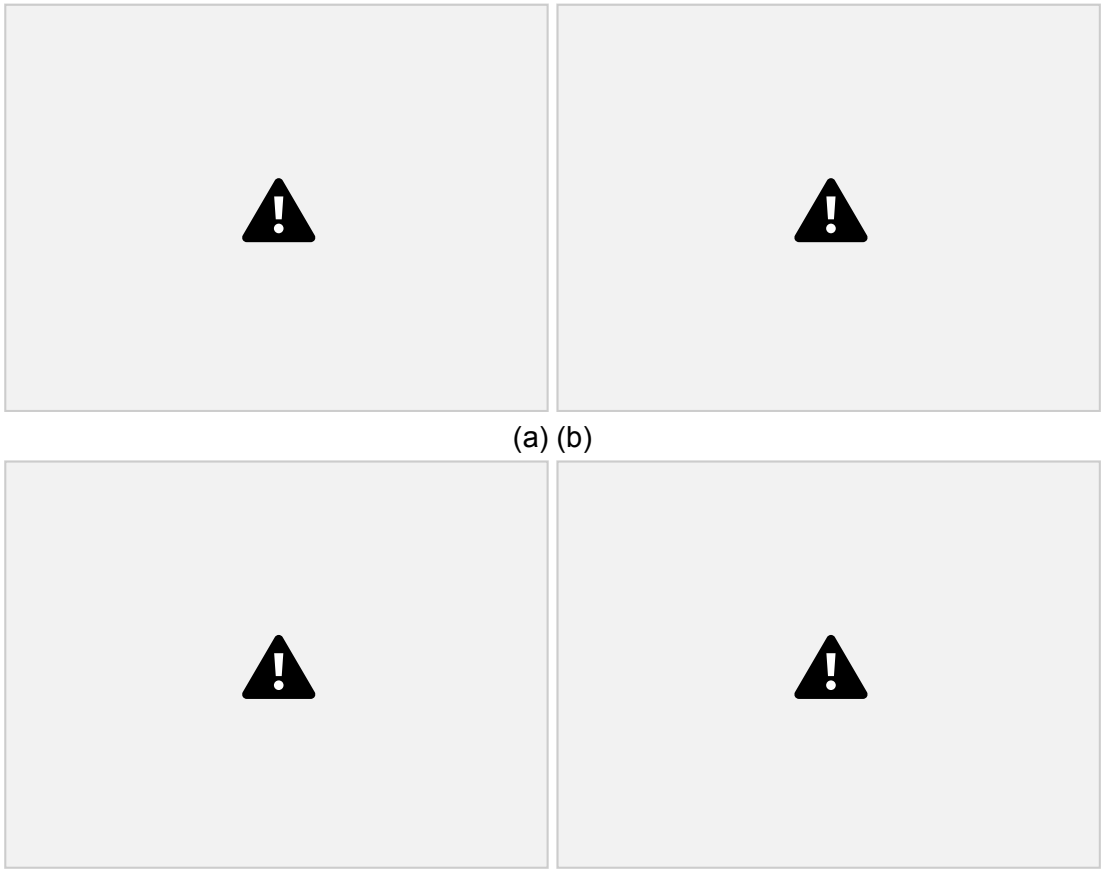
Figure 5.30: Modelled light curves from Cluster 5 with training only on the u band data. Notation same as Figure 5.3

across filters and the variability originates closer to the center of the AGN.

5.4.1 Clustering with SOM

From our sample, we evaluate the optimal hyperparameters of the SOM. The choices of parameters can be seen in Appendix E. We find a grid size of 7 with

a learning rate of 0.05 and σ of 0.75 produces the best balance between quantization error and topographic error. Once again, we utilize gradient-based clustering and obtain 7 clusters as seen in Figures 5.33 and 5.34 . We choose Cluster 4 as it is the cluster with the largest number of light curves.



(a) (b)

(c) (d)

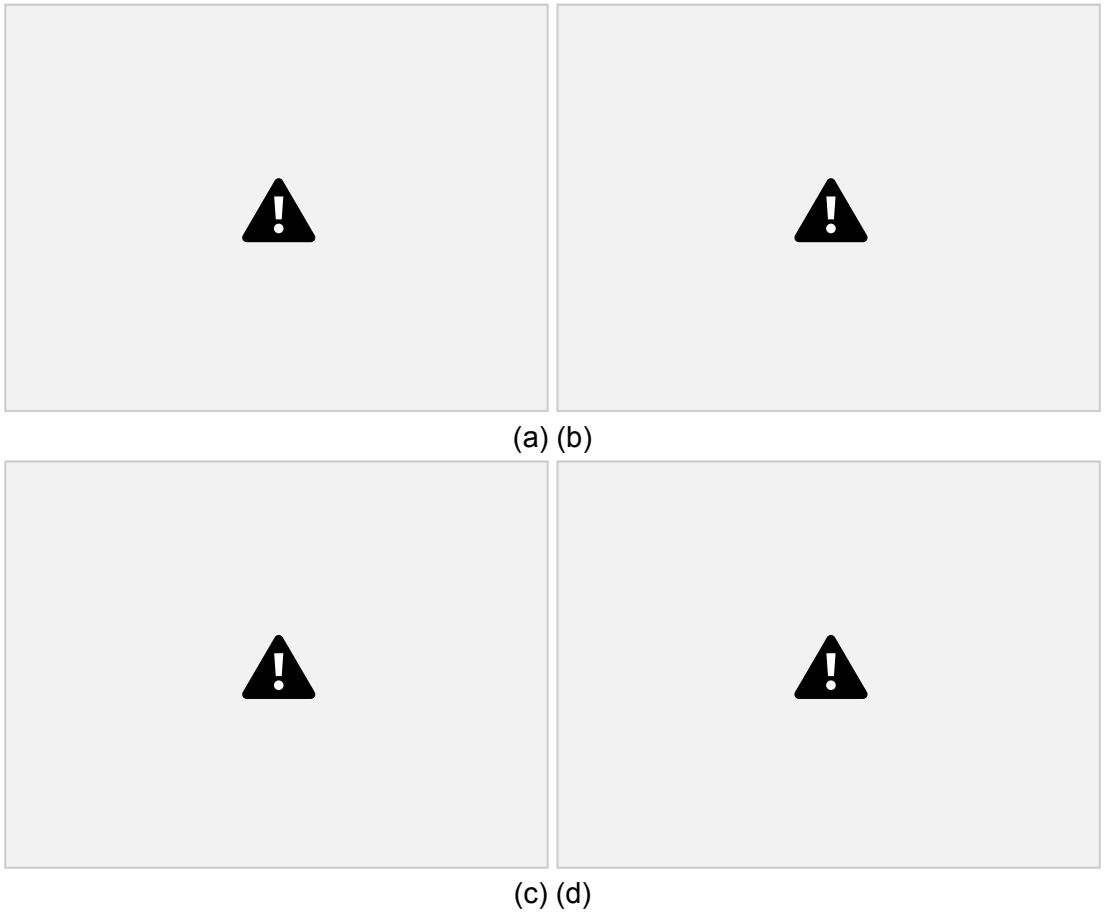
Figure 5.31: Modelled light curves from Cluster 5 with training on all bands of the LSST AGN DC. Notation same as Figure 5.3

5.4.2 Analysis of One Cluster

We utilize the same standard configuration for the model and run it for 2000 epochs on Cluster 4, with the option for early stopping. The model stops early at epoch 879.

We can see the training curves of the model in Figure 5.35. We see once

again that the model achieves a rapid convergence towards the optimum validation loss. The reconstructed test light curves are plotted in Figure 5.36. We see that the model can recover densely populated areas of the light curve very well. However, the model faces issues on less densely sampled regions of the light curve. All in all, the model doesn't stray from the trend of the light curves and can capture most of each light curve within the confidence intervals of the model.



(a) (b)

(c) (d)

Figure 5.32: Comparison of light curves from Cluster 5 reconstructed with training on only u-band data vs the same light curves trained on all band data. The light curves that contain u in their name have been trained on multiband data. Notation same as Figure 5.3

5.5 Transformers and Ticktocks

5.5.1 Performance on Simulated Curves

Before we utilize the transformer, there is a clear distinction between the light curves that contain tick-tock signals as compared to the light curves that do not. In Figure 5.37, we see that light curves containing tick-tock signals have a non-zero mean magnitude, while the light curves without them have a near-zero magnitude. This shows that the task should be an easy one for the transformer and it should be able to learn that the mean is an important feature to distinguish between light curves.

In Figure 5.38, it can be seen that the model performs very well on the simulated curves. The model can achieve a very low loss on both the training and validation

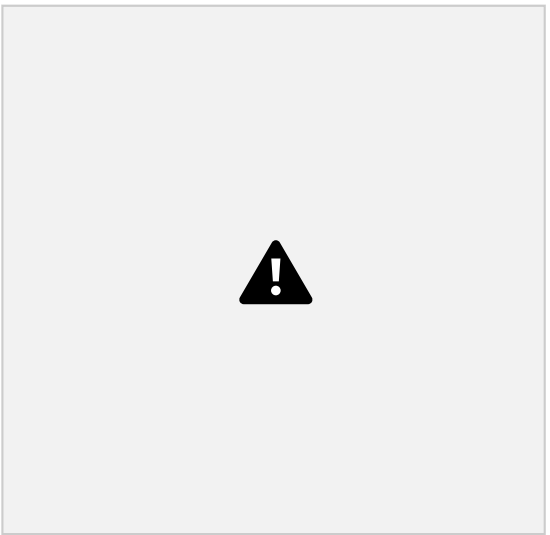


Figure 5.33: The distribution of light curves in the clusters obtained from the ZTF sample.

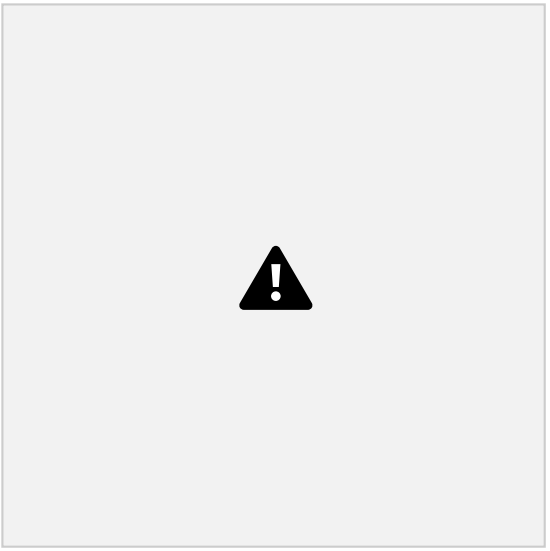


Figure 5.34: Visualization of light curves in the clusters obtained from the ZTF sample. Notation same as Figure 5.26

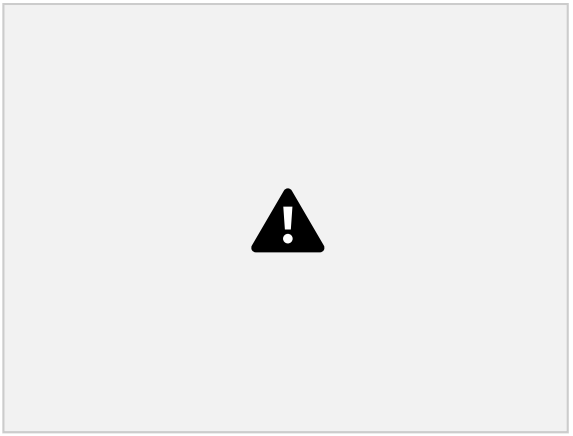
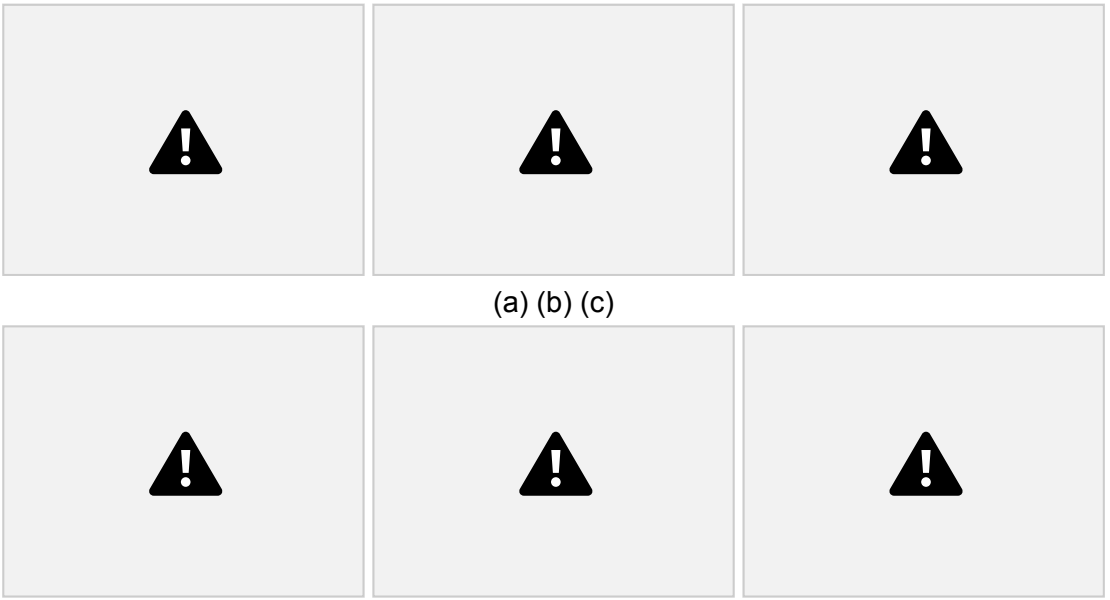


Figure 5.35: Training (in blue) and Validation (in orange) Curves for Cluster 4 from the ZTF light curves.



(a) (b) (c)

(d) (e) (f)

Figure 5.36: Reconstructed test light curves from Cluster 5 of the ZTF light curve sample. Notation same as Figure 5.3

data. However, the training is characterized by many spikes and is overall less smooth. This is a characteristic of many different transformer models tested on the simulated data. Lower learning rates tend to cause the model to oscillate less but prevent convergence. With a higher learning rate, the model tends to choose one label and predict that label for every single data point. Thus, the optimal solution is a learning rate of 10^{-4} with the characteristic spikes.

In Figure 5.39, we see the confusion matrix of the model on test data. The

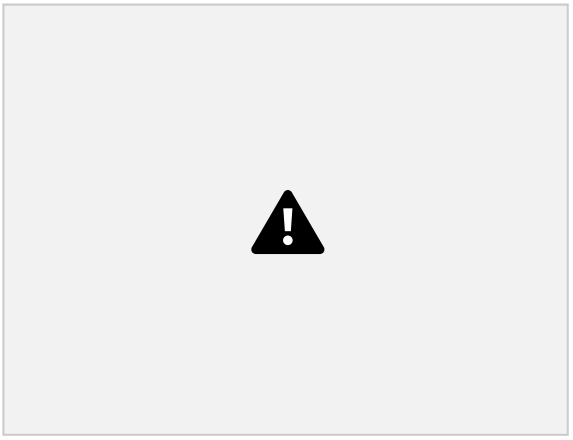


Figure 5.37: The mean magnitudes of the light curves with tick-tocks (in red) vs the light curves without tick-tocks (in green) for randomly simulated light curves.

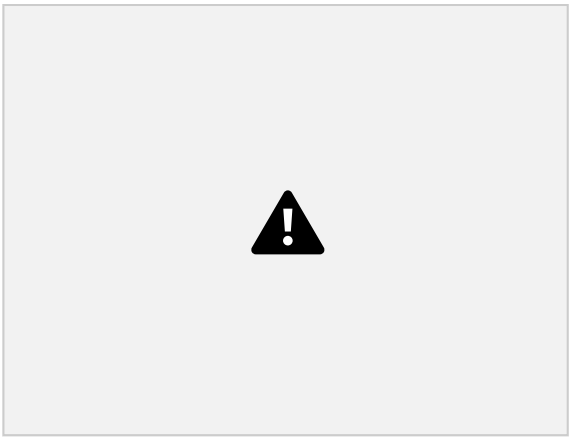


Figure 5.38: Training (in blue) and Validation (in orange) Curves for the transformer model on simulated light curves

model predicts the tick-tock signals with complete accuracy, It predicts only 1 tick tock signal as a non-tick-tock signal. Thus, the model can classify tick-tock signals extremely well in the simulated data as expected.

5.5.2 Performance on LSST AGN Data Challenge

In Figure 5.40, we see that for the light curves in the LSST AGN Data Challenge, there is no obvious bifurcation in the mean magnitude of the light curves. This means that the task will be tougher for the model than the previous case.

We plot the training and validation curves for the transformer model on the LSST AGN Data Challenge light curves in Figure 5.41. We see similar behaviour with multiple spikes in both the test and validation curves. This indicates that the

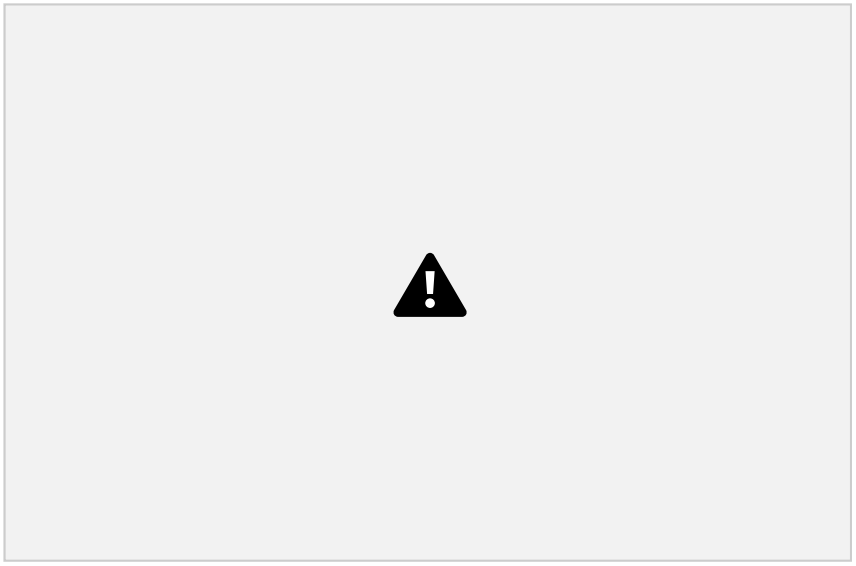


Figure 5.39: Confusion Matrix of the transformer model on simulated light curves. The diagonal entries are correct predictions, while off-diagonal entries are wrong predictions.

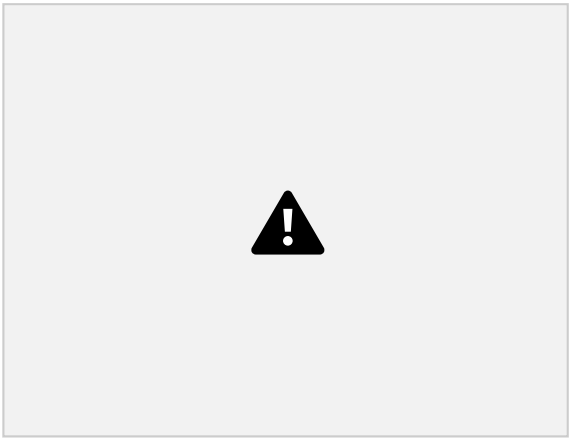


Figure 5.40: The mean magnitudes of the light curves with tick-tocks (in red) vs the light curves without tick-tocks (in green) derived from the LSST AGN Data Challenge

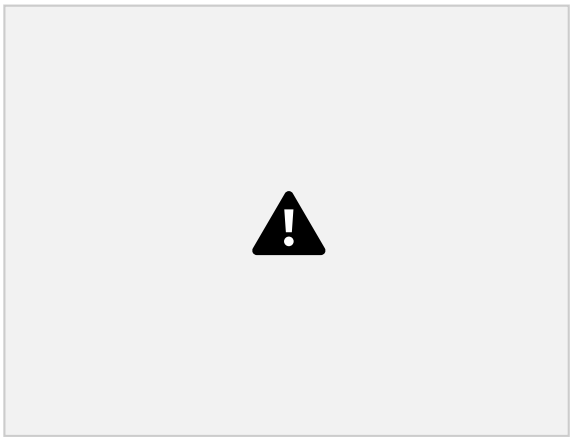


Figure 5.41: Training (in blue) and Validation (in orange) Curves for LSST AGN Data Challenge curves with injected tick-tock signals.

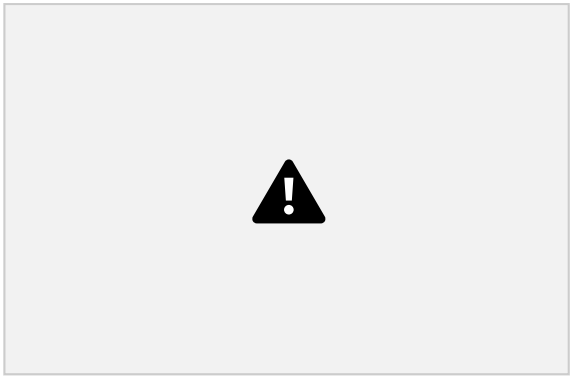


Figure 5.42: Confusion matrix for LSST AGN Data Challenge light curves with injected tick-tock signals. Notation same as 5.39

gradient landscape for any problem of this type is riddled with different minima that the model needs a sufficient learning rate to get out of. However, the model can achieve good results again with a near-zero loss.

In Figure 5.42, the model performs very well again. Other than 2 non-tick-tock signals identified as tick-tock signals, the model can correctly classify the test data. Finally, we allow the trained model to test on the sample of the LSST AGN Data Challenge that was used before. It identified 78 light curves as tick-tock signal candidates. Thus, these light curves can be further followed up on for tick-tock signals.

Through this transformer experiment, we have shown that transformer

models can identify subtle trends within light curves, such as merging binary black hole signals. Through training on larger datasets of simulated tick-tock signals, as well as a more detailed simulation of the tick-tock signals with a more careful selection

of amplitude and variation in the signal, the model can be improved and used on large catalogs of optical quasar light curves. Furthermore, similar experiments on light curves obtained from a higher cadence survey such as the ZTF allow the model to make more informed predictions of tick-tock signals.

Finally, we envision the light curves modeled by Neural Processes to improve the effectiveness of the transformer model. Besides higher cadence data with fewer gaps, attention has been shown to learn periodic signals within stochastic processes (see Dubois et al. [2020] and the reconstruction of periodic kernels). Thus, the SOM, the NP model, and the transformer model can all be used in tandem to perform detailed analysis on light curves obtained from large catalogs.