Data-driven Discovery of Transients in the New Era of Time-Domain Astronomy

Daniel Joseph Muthukrishna

Institute of Astronomy
University of Cambridge

This thesis is submitted for the degree of
Doctor of Philosophy

Clare College April 2021

“The nitrogen in our DNA, the calcium in our teeth, the iron in our blood, the carbon in our apple pies were made in the interiors of collapsing stars. The cosmos is within us. We are made of star-stuff. We are a way for the universe to know itself.”

- Carl Sagan

Declaration

This thesis is the result of my own work and includes nothing which is the outcome of work done in collaboration except as declared in the Preface and specified in the text. I further state that no substantial part of my thesis has already been submitted, or, is being concurrently submitted for any such degree, diploma or other qualification at the University of Cambridge or any other University or similar institution except as declared in the Preface and specified in the text. It does not exceed the prescribed word limit for the relevant Degree Committee.

Daniel Joseph Muthukrishna
April 2021

Data-driven Discovery of Transients in the New Era of Time-Domain Astronomy

Daniel Joseph Muthukrishna

Abstract

Time-domain astronomy has reached an incredible new era where unprecedented amounts of data are becoming available. New large-scale astronomical surveys such as the Legacy Survey of Space and Time (LSST) are going to revolutionise transient astronomy, providing opportunities to discover entirely new classes of transients while also enabling a deeper understanding of known classes. LSST is expected to observe over 10 million transient alerts every night, at least two orders of magnitude more than any preceding survey. It has never been more important that astronomers develop fast and automated methods of identifying transient candidates for follow-up observations.

In this thesis, I tackle two major challenges facing the future of transient astronomy: the early classification of transients and the detection of rare or previously unknown transients. I detail my development of a number of novel methods dealing with these issues. In the first chapter, I provide an introduction to the field of transient astronomy and motivate why new methods of transient identification are necessary. In the second chapter, I detail the development of a new photometric transient classifier, called RAPID, that is able to automatically classify a range of astronomical transients in real-time. My deep neural network architecture is the first method designed to provide early classifications of astronomical transients. In Chapter 3, I identify the issue that with such large data volumes, the astronomical community will struggle to identify rare and interesting anomalous transients that have previously been found serendipitously. I outline my novel method that uses a Bayesian parametric fit of light curves to identify anomalous transients in real-time. In Chapter 4, I highlight some issues with current photometric classifiers and improve upon RAPID so that it is capable of dealing with real data instead of just simulations. I present classifiers that perform effectively on real data from the Zwicky Transient Facility and the PanSTARRS surveys. Finally, in the last chapter, I discuss the conclusions of my work and highlight some future opportunities and work needed in preparing for discovery in the new era of time-domain astronomy.

Acknowledgements

There are so many people that I am grateful to for making this chapter in my life such an amazing experience. I would like to thank my supervisors, Kaisey Mandel and Gautham Narayan. I appreciate the time you've both invested in me, your mentoring, and the many academic opportunities you've given me. Thank you, Kaisey, for teaching me to think more deeply as a scientist and for your attention to detail, and thank you, Gautham, for the big ideas and for propelling my academic career with your support, guidance, and belief in me. I also want to thank all of my collaborators, who have guided me and helped me throughout my PhD. In particular, I want to thank Ryan Foley for hosting me at UC Santa Cruz and sharing your infectious passion for transient astronomy. I want to thank Michelle Lochner for the enthusiasm, positivity, support, and ideas you shared with me as a mentor. I'd also like to thank Armin Rest for hosting me during my visits to Baltimore, for sharing your wisdom, and for the entertaining banter on Gautham.

I am also deeply appreciative of the Cambridge Trust and the Cambridge Australia Poynton Scholarship for funding me throughout my PhD. I can hardly express my gratitude for making this incredible opportunity to study at Cambridge possible. I want to express my sincere thank you to everyone at the IoA for their support and friendly tea-time conversations. In particular, my time in the department wouldn't have been as enjoyable as it was without my friends and lunchtime crew, James, Gyuchul, Peter, Luis, and Fruzsi, and my office mates and darts buddies, Stephen and Jonny.

My time at Cambridge has been one of the most enjoyable chapters in my life thanks to Clare College and the great friendships I've made. For all the great times outside the office and while working from home, I want to thank all my housemates at Clare Court 1-16 and my friends. Thomas, Calum, Audrey, Danny, Nick, Sam, Will, Andrea, Lena, and Penny, thank you for all the fun times and support, for keeping me sane and entertained while working from home during the pandemic, and for being some of my best mates over my PhD that I'm sure will continue as lifelong friends.

Finally, I am most thankful to my family. I want to thank my Mum, for her unconditional love and support of me and my progress at every step of my life. My twin brother, Chris, for listening to all my issues from the other side of the world, for being a constant source of positivity, and for giving me advice whenever I need it. My sister, Tash, for her love and support. My brother, Michael, sister-in-law, Steph, and my niblings, Robert and Lucy, for the fun getaways from Cambridge, and for all your advice and support. I wouldn't be where I am today without all of you and all the amazing people in my life, and for that, I am eternally grateful.

Table of contents

List of figures

List of tables

Chapter 1

Introduction

1.1 Scientific Motivation

Over the last century, observations of the time-domain universe have led to some of the most significant discoveries in cosmology and astronomy. In the early 20th century, Henrietta Leavitt identified a relationship between the period and luminosity of Cepheid variable stars (Leavitt, 1908; Leavitt & Pickering, 1912). In 1929, Edwin Hubble used Leavitt's work to discover that the universe was expanding (Hubble, 1929), and independently in 1927, Georges Lemaître discovered a solution to Einstein's equations for an expanding universe (Lemaître, 1927). Around 70 years later, two teams of scientists (Perlmutter et al., 1999; Riess et al., 1998) used the standardised luminosities of Type Ia supernovae (SNe Ia) to independently discover that this expansion was accelerating and not slowing down. This was a major challenge to our understanding of the universe and led to the introduction of dark energy to explain the cosmic acceleration. The simplest dark energy model is Einstein's cosmological constant, with a fixed equation of state, $w = -1$. However, other explanations of the acceleration such as modified gravity, scalar field theories or quintessence predict different, or time varying values of w. The goal of modern supernova cosmology is to constrain the precise behaviour of w using more and better supernova measurements. While constraints on these cosmological parameters have improved as a result of cleaner and larger datasets, and multiple cosmological probes, the nature of dark energy remains mysterious (e.g. Weinberg et al., 2012). Nevertheless, despite this mystery, SNe Ia remain the most important probes of late-time cosmic acceleration and support general relativity as the best tested model for gravity on all length scales from torsion-balance pendulums in a lab to cosmological scales. Recently, this theory has passed another key test with the detection of gravitational waves by LIGO.

In 2017, LIGO and Virgo's interferometers detected gravitational waves from a neutron star merger. Within seconds of this result and for several hours following the event, gamma ray, optical, infrared and x-ray measurements were made, in what has been the largest joint multi-messenger effort by astronomers (e.g. Abbott et al., 2017c; Arcavi et al., 2017; Coulter et al., 2017; Cowperthwaite et al., 2017; Drout et al., 2017; Evans et al., 2017; Pian et al., 2017; Savchenko et al., 2017; Shappee et al., 2017; Smartt et al., 2017; Soares-Santos et al., 2017b; Tanvir et al., 2017; Troja et al., 2017; Valenti et al., 2017). For the first time, kilonovae (Metzger, 2017) were directly observed as the electromagnetic signatures of neutron star mergers - events which shine brighter than their host galaxy and are responsible for all the heavy elements found in the universe today. The joint effort was also able to constrain the speed of gravitational waves, and put general relativity on a stronger footing in cosmology, with several modified theories of gravity being ruled out.

Cepheids, supernovae, and kilonovae have led to exciting discoveries in astronomy, and are just three of the many classes of transient and variable objects in the universe. New large scale surveys will observe orders of magnitude more transient objects than ever before. The upcoming Legacy Survey of Space and Time (LSST) on the Vera Rubin Observatory (VRO), for example, will scan the entire southern sky to observe thousands of new transients every day, and is likely to discover new classes of objects never before seen. This unprecedented opportunity comes with major new data-handling and identification challenges. The sheer number makes it unfeasible to analyse these events one by one; and as such, automated machine learning and data techniques are necessary to process, classify, and discover objects from the data influx. I discuss the literature on machine learning for transient identification in section 2.2.

A major challenge is that our resources for spectroscopic follow-up is limited. The future of time-domain astronomy relies on the automation of identifying which of the millions of transient alerts observed by upcoming surveys are most promising for follow-up observations. In this thesis, I develop methods for classifying transients and discovering anomalous events from survey telescopes. Using these methods, astronomers will be able to prioritise the follow-up of transient candidates based on the photometric classification and the likelihood of a transient candidate being anomalous.

1.1.1 Supernova Cosmology

Supernovae (SNe) are the result of the violent destruction of entire stars. They are rare events. Over the past millennium, only six supernovae have been confirmed in the Milky Way. The expected rate of supernova events in a galaxy like the Milky Way is 2.84 ± 0.60 SNe per century, while the expected volumetric rate of SNe in the local universe is on the order of 10^{-5} to 10^{-4} SNe Mpc^{-3} year^{-1} (Li et al., 2011). In the past two decades, dedicated telescope surveys monitoring changes in the sky has meant that supernovae are now found regularly. The rise in transient surveys was, in part, motivated by the desire to obtain more SNe Ia for use in cosmology. Several surveys, such as the Supernova Factory (Aldering et al., 2002), the Supernova Legacy Survey (SNLS, Astier et al., 2006), ESSENCE (Davis et al., 2007), the Harvard-Smithosonian Center for Astrophysics supernova program (Hicken et al., 2009), the Carnegie Supernova Project (CSP, Folatelli et al., 2010), the Berkeley SNIa Program (BSNIP, Silverman et al., 2012), and the Dark Energy Survey (DES, Dark Energy Survey Collaboration et al., 2016) have had discovering or spectroscopically following up more SNe Ia for cosmology as one of their key motivations. These surveys coupled with the fact that SNe Ia are much brighter and thus easier to find than core-collapse supernovae, has meant that the worldwide collection of supernova observations are significantly dominated by SNe Ia.

For cosmology, it is important to classify observed transients to distinguish a pure sample of SNe Ia. Supernovae are categorised into several types based on features in their optical spectrum taken near maximum brightness (Filippenko, 1997). SNe Ia can be spectroscopically distinguished by the presence of Si II and the lack of hydrogen and helium in their spectral features. Thus, conventional SNIa cosmological analyses are based on spectroscopically confirmed samples (Scolnic et al., 2018). With rapidly increasing samples from transient surveys, there are not enough resources available worldwide to spectroscopically confirm each detection (Kessler et al., 2010a). Therefore, astronomers are interested in developing methods to photometrically classify supernovae to make optimal use of the large data volumes in future cosmological surveys.

Distinguishing SNe based on sparse light curves is a much more difficult challenge than classification from spectra, and there is a risk of contamination due to misclassification. Classifying a SNIa sample with photometric light curves alone means that there is a tradeoff between the purity[1] of the sample, the efficiency[2] in retaining a large sample size, and the

[1] The fraction of classified SNe Ia that really are SNe Ia

[2] The fraction of true SNe Ia in the sample that have been classified as SNe Ia

resulting systematic biases in the constraints on cosmological parameters. As new large surveys decrease their fraction of spectroscopically confirmed SNe Ia in their sample, their purity drops and hence the systematic errors in their cosmology increases. Consequently, the quality of photometric classifiers needs to improve to meet this new challenge.

Recent work by Campbell et al. (2013) and Jones et al. (2017) used a photometrically classified sample of SNe Ia for cosmology with data from SDSS-II and Pan-STARRS, respectively. They used SNANA simulations (Kessler et al., 2009) to estimate that their photometric SNIa sample was affected by up to 5% contamination from core collapse supernovae, which both groups claimed had a small or insignificant effect on their cosmological constraints. They each estimated that they had a SNIa typing efficiency of approximately 70%, limiting their total sample size. LSST will observe orders of magnitude more SNe Ia than both of these surveys, and being able to obtain a high purity and high efficiency photometric sample from accurate classifications will allow for precision cosmological inferences.

1.1.1.1 Hubble tension

Observations of SNe Ia enable the expansion rate of the universe, the Hubble-Lemaître constant H_0, to be measured directly. The value of H_0 has been measured to increasingly better precision over the past decade. The most recent measurement from using the standard distance ladder approach using SNe Ia calibrated with Cepheid variable stars reaches a precision of 1.8% and finds $H_0 = 73.2 \pm 1.3$ km s^{-1} Mpc^{-1} comes from Riess et al. (2021). However, this measurement is in direct tension with predictions from Planck CMB observations that assume a ΛCDM cosmology. The most recent analysis from Planck derives a value $H_0 = 67.44 \pm 0.57$ km s^{-1} Mpc^{-1} (Efstathiou & Gratton, 2019) which differs from the distance ladder measurement by 4.2σ. This tension has come to be known as the *Hubble tension* (see Verde et al. 2019 for a review from the recent workshop on this issue). As illustrated in Figure 1.1, since the release of the Planck measurements in Planck Collaboration et al. (2014), higher precision analyses have led to a continued increase in the tension.

The Planck analysis relies on the assumption of a ΛCDM model of cosmology. The six-parameter ΛCDM cosmology is the Standard Model of Cosmology and has been extraordinarily successful at explaining the cosmic microwave background radiation (CMB) and a range of other astronomical data (e.g. Planck Collaboration et al., 2020). But, the discrepancy between late-time universe measurements of H_0 from the distance ladder and the early-time universe derivations from Planck point to the possibility of new physics being

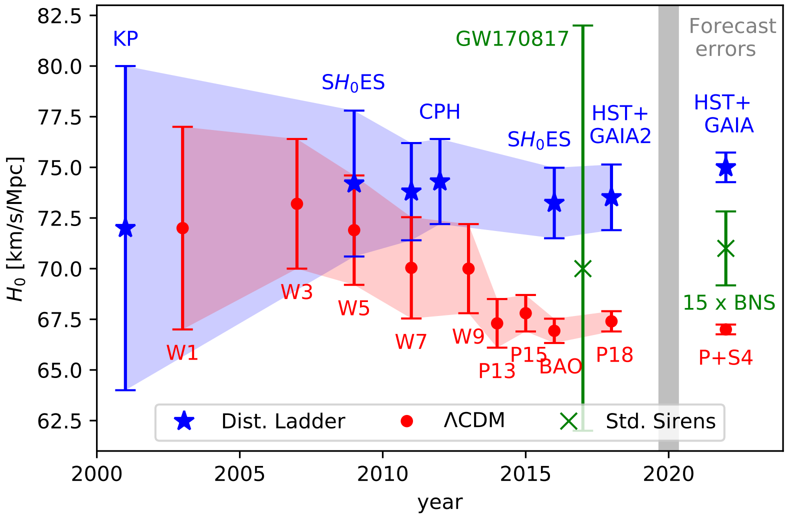


Fig. 1.1 The Hubble tension evolving over time. Measurements of H_0 from the local universe based on Cepheids and SNe Ia are shown in blue, derived values of H_0 from the CMB assuming ΛCDM are shown in red, and direct measurements of H_0 with kilonovae as standard sirens is shown in green. The plot is taken from Figure 8 of Ezquiaga & Zumalacárregui (2018).

required. Ongoing research is putting effort to constrain uncertainties in the analyses to determine whether the tension is caused by an incomplete understanding of the physics of the early-time universe or from systematic uncertainties in the analyses.

One of the largest systematic uncertainties in SNIa cosmology is the calibration of the low-redshift SNIa sample (e.g. Betoule et al., 2014b; Foley et al., 2018b; Scolnic et al., 2018). Thus, supernova surveys, such as the Foundation Supernova Survey (Foley et al., 2018b) are working to provide a large and homogeneous sample of low-redshift ($z < 0.1$) SNe Ia. Obtaining a larger sample of well-calibrated low-redshift SNe Ia will help to alleviate the calibration uncertainty in supernova cosmology which will improve the precision on both w and H_0.

Furthermore, the recent binary neutron star merger observed as a gravitational wave event, GW170817 (Abbott et al., 2017a), with an optical counterpart as a kilonova (Abbott et al., 2017c) enabled the first use of *standard sirens* for cosmology (Abbott et al., 2017b). Using the distance measured using gravitational waves and the recession velocity measured from the electromagnetic signatures, an independent direct measure of the Hubble-Lemaître constant was obtained. A larger collection of kilonovae with gravitational wave counterparts will enable a more precise measurement of H_0 in the future as illustrated in Figure 1.1 (Chen et al., 2018). For these reasons among others, future surveys are aiming to obtain larger and pure samples of kilonovae and supernovae.

1.1.2 Progenitor Physics

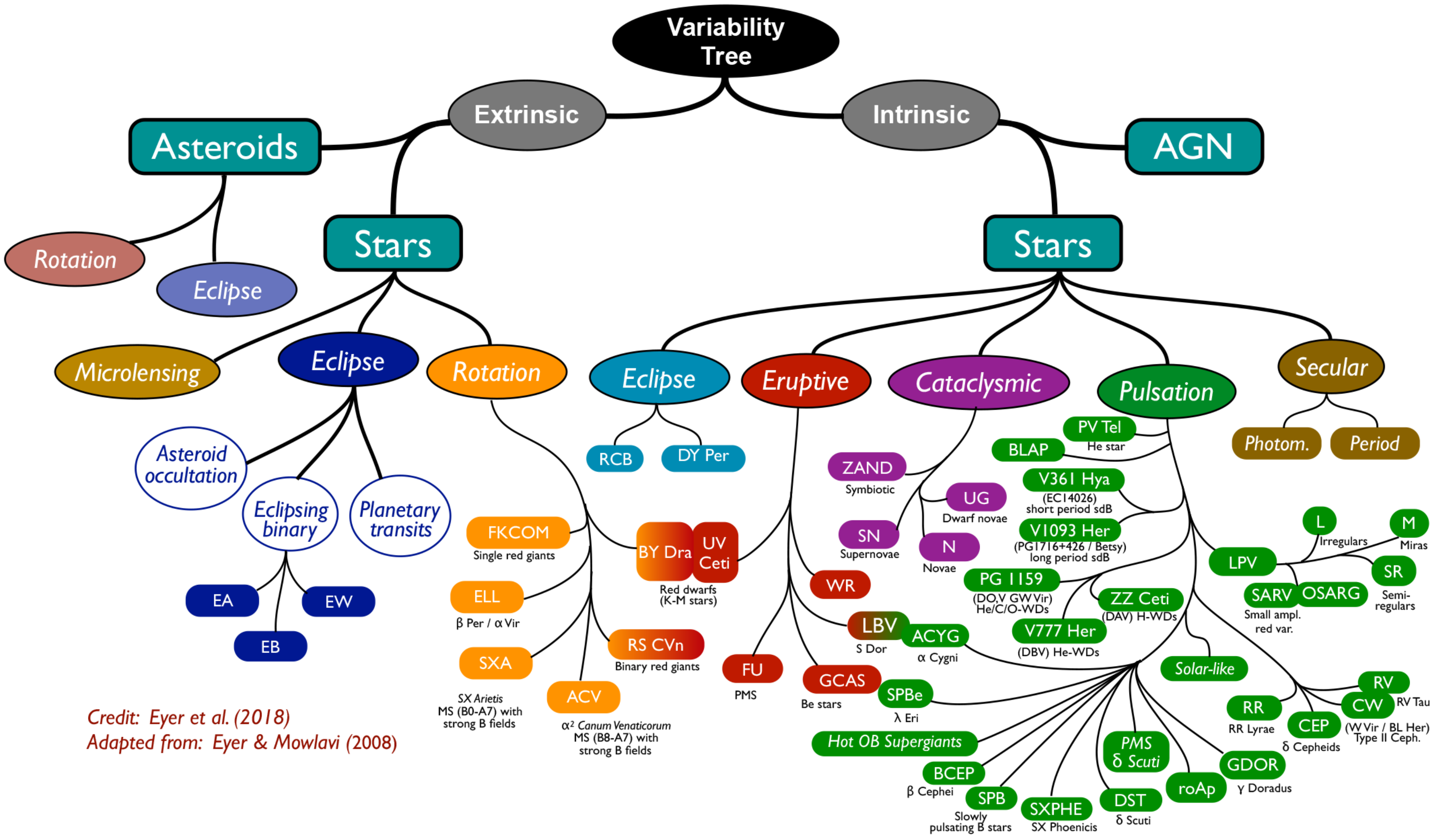
Obtaining a large mostly pure SNIa sample will help to reduce the statistical errors in cosmological analyses. However, understanding both the progenitors and the physical explosion mechanisms is important for reducing the space for unknown systematics that might be due to the evolution of SNIa progenitors with cosmic time. While it is widely accepted that SNe Ia are the thermonuclear explosions of carbon-oxygen white dwarfs, the progenitor scenarios are not well understood and two scenarios are commonly considered. In the single degenerate (SD) scenario, a non-degenerate binary companion star deposits matter onto a white dwarf until it reaches the Chandrasekhar limit and thermonuclear runaway occurs (Whelan & Iben, 1973). The double degenerate (DD) model postulates that a binary white dwarf system merges after a gravitational inspiral and explodes from the subsequent carbon ignition (Iben & Tutukov, 1984). It remains uncertain whether SD or DD scenarios result in the SNIa population, or whether both channels may contribute to the population - perhaps explaining the overall population diversity. Furthermore, whether the thermonuclear burning proceeds as a detonation, a deflagration, a transition from one to the other, or some incomplete burning scenario is also still unresolved (Hillebrandt & Niemeyer, 2000; Noebauer et al., 2017).

The best way to distinguish between whether SNe Ia evolve under the SD or DD progenitor scenarios, and which explosion mechanism is responsible for SNe Ia is by detailed time-resolved observations at very early times. For example Kasen (2010) predicts that in a SD scenario, the shock from the supernova explosion colliding with the companion star will produce an early ultraviolet/blue excess in the first few days after explosion. Marion et al. (2016) and Hosseinzadeh et al. (2017) recently observed an early blue bump in the light curves of two SNe Ia (SN2012cg, SN2017cbv), providing evidence for the SD scenario. These recent observations are an example of the exciting opportunities to learn about the nature of the SNIa progenitor system and their deaths in current and future transient surveys. As LSST plans to observe tens of thousands of transients per day, determining which of these may be worth follow-up for improving our understanding of the SNIa population and hence reducing unknown systematics in cosmology is an important challenge that our work hopes to meet.

While SNe Ia are of particular relevance to cosmologists, understanding the astrophysics of stellar explosions is an interesting science case on its own. The physical mechanism of some core-collapse supernovae (CCSNe) are still under debate (see Smartt (2009) for a review of the progenitors of core-collapse supernovae). The discovery and early-time observations of the famous nearby supernovae, SN1987A, and SN1993J enabled a deeper understanding of the progenitors of CCSNe. However, it has so far been difficult to detect progenitors of supernovae beyond 10 Mpc, and rapid and intense follow-up of supernovae in nearby galaxies is necessary to characterise the explosions and progenitors of supernovae (Smartt, 2009). Future deep and wide-field surveys will discover many more rare SNe in much higher numbers. The opportunity for characterising these events relies on the rapid identification of suitable supernovae for follow-up. Identifying the class of transients using only early-time data, and following up interesting candidates spectroscopically will enable us to understand the progenitors of many poorly understood transients.

1.2 Transient Diversity

The time-domain universe comprises a great variety of transient and variable phenomena that are due to different physical origins. Figure 1.2 illustrates many classes of transients and variables separated by their known causes. Throughout this thesis, we focus on a subset of supernovae and other transients that are commonly confused as supernovae. In Figure 1.3, we plot different classes of transients separated by their peak luminosity and characteristic timescale. Common transients, such as SNe Ia (thermonuclear supernovae) are shown in grey, while more poorly understood objects are shown in coloured regions. The gaps in Figure 1.3 are regions of the parameter space that have been largely unexplored by past surveys of the transient universe. Upcoming and ongoing surveys, such as the LSST and the Transiting Exoplanet Survey Satellite (TESS, Ricker et al., 2015) will explore fainter and faster than ever before, and may reveal entirely new classes of objects. In the following subsections, we detail the physical mechanism and light curve shapes of a range of transient objects that are expected to be observed by current and upcoming transient surveys and that are used throughout this thesis.



Credit: Eyer et al. (2018)
Adapted from: Eyer & Mowlavi (2008)

Fig. 1.2 An organisation of transients and variables. The variability tree is separated into four levels: the distinction of intrinsic versus extrinsic variability, the separation into major types of objects (asteroids, stars, and AGN), the physical origin of the variability, and the class name. The figure is taken from Figure 1 of Gaia Collaboration et al. (2019) which is adapted from Eyer & Mowlavi (2007).

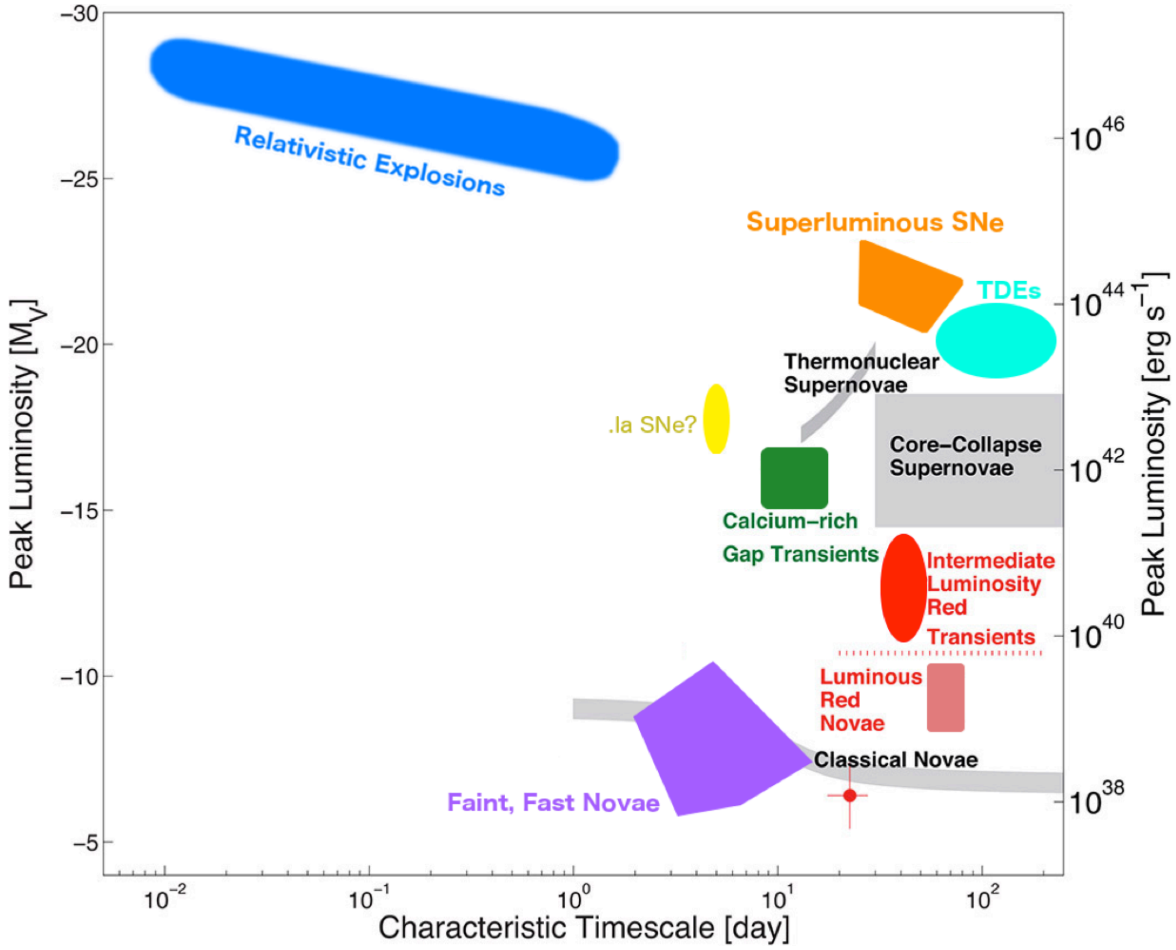


Fig. 1.3 Plot illustrating the characteristic timescale and peak luminosity of many transients. The grey regions represent common transients where "Thermonuclear Supernovae" are SNe Ia, the "Core-Collapse Supernovae" on the right are both SNe Ibc and SNe II. The coloured regions represent rarer and often less understood classes of transients. The Figure is adapted from (Kasliwal, 2011).

1.2.1 Supernovae

Supernovae are the result of the core-collapse of massive stars or the thermonuclear explosions of white dwarfs. Because the exact physical mechanism of many sub-classes of supernovae are not well understood, they are typically classified based on their light curve shape and the presence of certain chemicals in their optical spectrum observed near peak luminosity. Example light curves of several supernova types are plotted in Figure 1.5.

1.2.1.1 Type Ia Supernovae

Type Ia supernovae (SNe Ia) are thought to be caused by the thermonuclear explosion of a binary star system consisting of a white dwarf accreting matter from a companion star. The white-dwarf eventually accretes so much mass that its core reaches a critical density - known as the Chandrasekhar limit - that causes an uncontrolled fusion of carbon and oxygen. The energy of the explosion of SNe Ia produces light curves that can be empirically standardised using the Phillips relation (Phillips, 1993). Phillips (1993) showed that the peak luminosity of SNe Ia is positively correlated with the timescale over which the light curve decays from maximum. Correcting for this correlation as well as other small corrections to the colour have enabled SNe Ia to be used as standardisable candles to measure cosmic distances.

SNe Ia are characterised by a lack of hydrogen in their spectra and strong silicon-II features. In recent years, many subgroups in the SNIa class have been defined to account for the diversity in the class. In Figure 1.4 we reproduce a figure by Taubenberger (2017) that illustrates various sub-classes of observed thermonuclear supernovae on a plot of the B-band magnitude against the light-curve decline rate expressed as the decline within 15 days after peak in B-band, Δm_{15}. The plot can be considered a version of Figure 1.3 that has zoomed in near the "Thermonuclear Supernovae" strip and reversed the timescale axis. For a full review of the different sub-classes of SNe Ia see Taubenberger (2017) or Ruiter (2020). We list a few common sub-classes of SNe Ia that we use throughout this thesis and briefly describe how they differ from normal SNe Ia (SNIa-norm) below.

SNIa-91bg have a significantly lower luminosity than SNIa-norm and are also faster, having narrower light curves. They are slightly redder at maximum light and often look similar to the light curves of SNe Ibc. The first of this class was SN 1991bg (Filippenko et al., 1992a).

SNIa-91T are characterised by higher peak luminosities and broader light curves than SNIa-norm. Their colours are very similar to normal SNe Ia except for sometimes being slightly bluer pre-maximum. The first of this class was SN 1991T (Filippenko et al., 1992b; Ruiz-Lapuente et al., 1992).

SNIa-x typically have lower luminosities and slower ejecta velocities than SNIa-norm. They are often found in young stellar populations. See Foley et al. (2013) or Jha (2017) for a full review of this sub-class.

SNIa-csm have higher luminosities and slower decline rates than SNIa-norm that are thought to be caused by interaction with its circumstellar material.

SNIa-pec include all other transients that resemble SNe Ia but look slightly 'peculiar'.

1.2.1.2 Type Ibc Supernovae

Type Ibc supernovae (SNe Ibc) (examples in Drout et al. 2011) are only found in regions of star formation, and are thought to be the result of the core collapse of massive stars. They are typically characterised from other supernovae by a lack of hydrogen and silicon features in their spectra. Type Ib and Ic supernovae have both had their hydrogen envelopes stripped prior to explosion and thus look quite similar. However, SNe Ic differ from SNe Ib because they have had their helium envelopes stripped as well, and thus lack helium features in their spectra (Modjaz et al., 2014).

Their light curve shape and spectra near maximum light look very similar to SNe Ia, but tend to have magnitudes about 1.0-1.5 times fainter than a typical SNIa. This is however, similar to sub-luminous SNe Ia (such as SNIa-91bg), which makes them difficult to identify from light curves alone.

Some sub-classes of SNe Ibc are briefly described as follows:

SNIb are characterised by a lack of hydrogen and the presence of helium in their optical spectra.

SNIbn are similar to SNe Ib but are distinguished by the presence of narrower emission lines of helium in their early spectra (see Branch & Wheeler, 2017d).

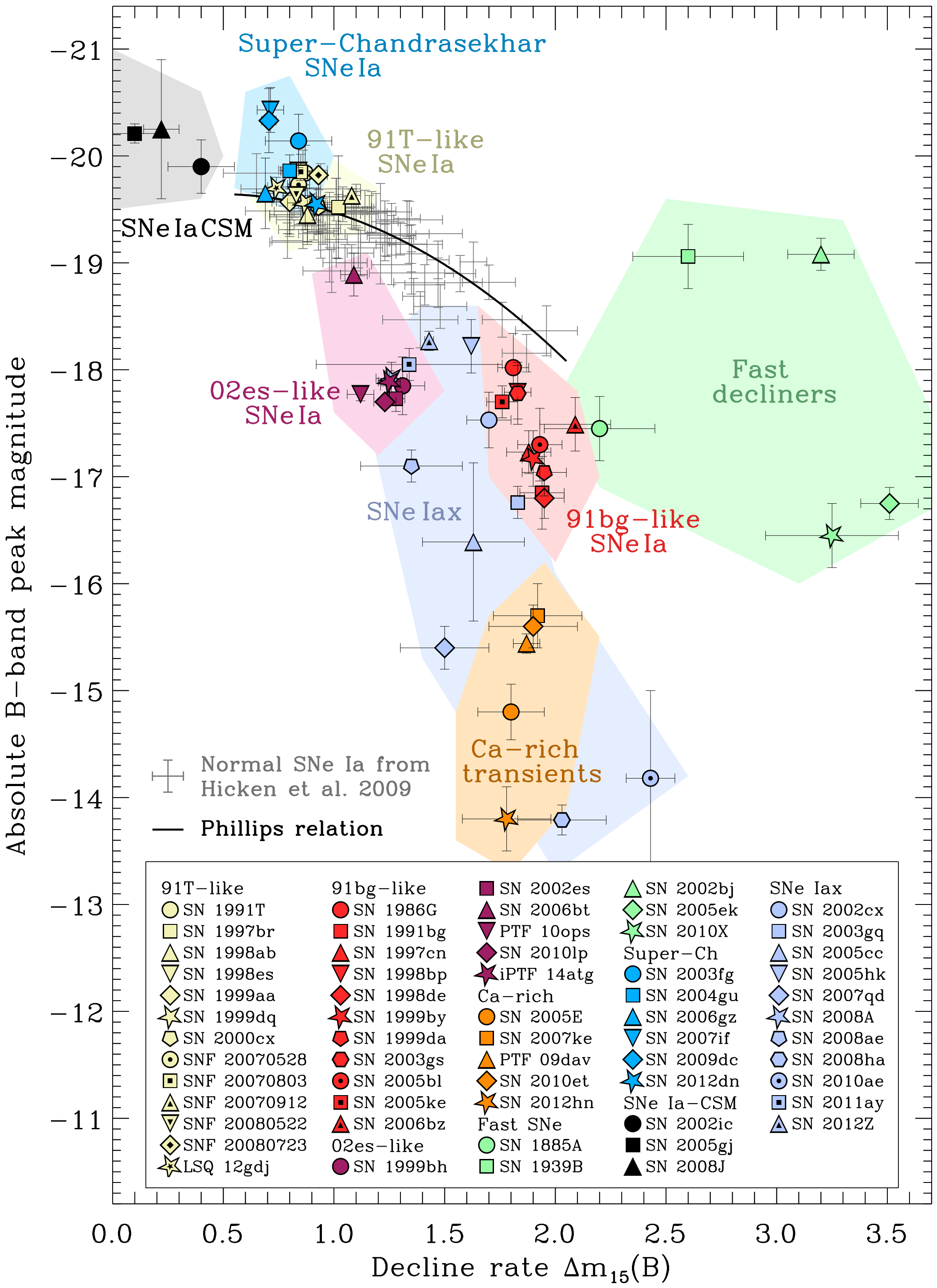


Fig. 1.4 The sub-classes of thermonuclear supernovae plotted on a parameter space showing the peak *B*-band magnitude against the light-curve decline rate expressed as the decline within 15 days after peak in *B*-band, Δm_{15}. Different classes are highlighted in different colours and examples of each class are shown as the data points. The plot is very similar to Figure 1.3, but effectively zooms in near the "Thermonuclear Supernovae" strip of the plot with the horizontal axis being reversed. The plot is taken from Figure 1 of Taubenberger (2017) where more detail about each of these classes can be found.

SNIc are characterised by a lack of hydrogen and helium in their spectra. Their light curve shapes look similar near peak brightness, but have a range of late-time decline rates (see Branch & Wheeler, 2017e).

SNIc-BL have particularly broad spectral lines compared to normal SNe Ic.

SNIIb display much more similarity with SNe Ibc than SNe II. The spectral evolution of SNe IIb reveal that their hydrogen features fade quickly and lines of helium become much more prominent. After the appearance of helium, SNe IIb closely resemble SNe Ib (Branch & Wheeler, 2017c), and thus we include them under the SNIbc class instead of the SNII class throughout this thesis.

1.2.1.3 Type II Supernovae

Similar to SNe Ibc, Type II supernovae (SNe II) are only found in regions of star formation and are thought to be the result of the core collapse of massive stars. They are characterised by the presence of hydrogen emission in their spectra. They often remain luminous for an extended period of time after maximum compared to Type I supernovae.

There is significant diversity within the class (Hillier & Dessart, 2019; Zampieri, 2017) and we briefly describe a few sub-classes below.

SNIIP have prominent hydrogen spectral lines at all epochs in their transient phase. Their light curve shape is characterised by a plateau, whereby after maximum brightness it undergoes an extended phase of nearly constant brightness for up to 100 days before declining. They are thought to also have a shock breakout very shortly after explosion (this has seldom been observed, but has been reported in the x-rays in Fransson & Lundqvist 1989, ultraviolet in Schawinski et al. 2008, Gezari et al. 2015, and optical in Garnavich et al. 2016, for example).

SNIIL differ from SNIIP because their light curves decline linearly in magnitudes (log scale) and do not undergo an extended plateau. See Branch & Wheeler (2017a) for a detailed description of this class.

SNIIn are characterised by the presence of narrower lines of hydrogen in their spectra. These narrower lines are thought to arise from interaction with the circumstellar medium. SNIIn are typically brighter than SNIIP and have a wide variety of light curve shapes. See Branch & Wheeler (2017b) for a detailed description.

SNII-pec include all other transients that resemble SNe II but look slightly ‘peculiar’.

1.2.2 Point Ia Supernovae

Point-Ia supernovae (also known as .Ia SNe) are a hypothetical supernova type which are one-tenth as bright as regular Type Ia supernovae, and have not yet been observed. They are expected to be the result of the early onset of detonation of helium transferring white dwarf binaries known as AM Canum Venacticorum systems (Shen et al., 2010). Helium that accretes onto the carbon-oxygen white dwarfs undergoes unstable thermonuclear flashes when the orbital period is short: in the 2.5–3.5 minute range (Bildsten et al., 2007). This process is strong enough to result in the onset of a detonation.

1.2.3 Super Luminous Supernovae

Super-luminous supernovae (SLSN) (e.g. Quimby et al. 2007) are stellar explosions that are substantially more luminous than normal supernova events: with a luminosity of at least 7×10^{43} erg/s (< -21mag). While the physical mechanism behind these events is not well understood, some candidate models are described in this section, with example observed light curves shown in Figure 1.5. As defined in Gal-Yam (2012b), there are three broad classes of SLSNe: SLSN-II are hydrogen-rich events, SLSN-I lack spectroscopic signatures of hydrogen, and a rarer class with broader and slowly evolving light curves. These were originally called SLSN-R, as their luminosity was thought to be dominated by radioactivity (Gal-Yam, 2012b), but more recent work has indicated that other powering mechanisms are plausible. See Gal-Yam (2019) for a comprehensive review of SLSNe.

1.2.3.1 SLSN-II

SLSN-II are hydrogen-rich supernovae caused by the explosions of massive stars. Their observational characteristics are diverse and they have brightnesses that vary from up to $-\sim$ 22 mag down to the typical brightnesses of the normal core-collapse supernova populations. They are sometimes thought to fall under the SN IIn class. Their explosions typically occur within thick hydrogen envelopes making investigations about their energy source still mostly speculative (Gal-Yam, 2012b). The first well-studied example of SLSN-II was SN 2006gy (Ofek et al., 2007; Smith & McCray, 2007).

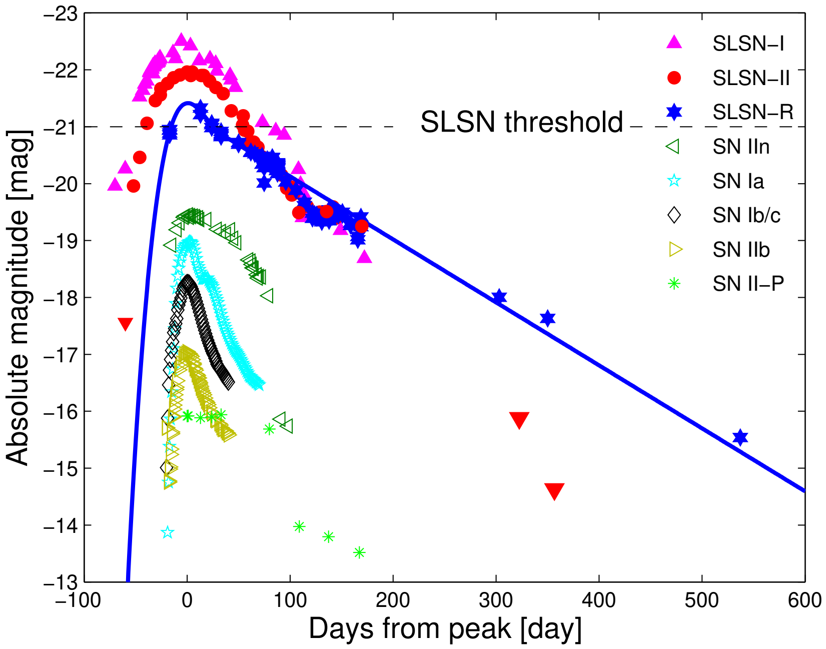


Fig. 1.5 The *r*-band light curves of the prototypical transients of various supernova classes. The dashed black line shows the cutoff for the classification of super-luminous supernovae (SLSN). The blue line shows a model of a SLSN-R powered by the radioactive decay of ^{56}Co. The Figure is taken from Figure 1 of Gal-Yam (2012b).

1.2.3.2 SLSN-I

SLSN-I are some of the brightest optical transients, but differ from SLSN-II because they lack hydrogen in their spectra. Their central engine are thought to be *magnetars* which are young and highly magnetised neutron stars. Their emission is powered by the decay of enormous internal magnetic fields. They tend to be around 50 times brighter than SNe Ia and their host galaxies are typically metal-poor dwarf galaxies (Gal-Yam, 2012b). Their observed population consists of just 29 sources, but are expected to represent 10% of the entire young neutron star population (Kaspi & Beloborodov, 2017).

1.2.3.3 Pair-Instability Supernovae

While massive stars generally explode as Type II or Ibc supernovae, the core-collapse of even more massive stars between 130 and 250 solar masses are known as pair-instability supernovae (PISN). PISNe are thought to be runaway thermonuclear explosions of massive stars with oxygen cores initiated when the internal energy in the core is sufficiently high to initiate pair production. This pair-production from γ-rays in turn leads to a dramatic drop in pressure support, and partial collapse. The rapid contraction leads to accelerated oxygen ignition, followed by explosion. These explosive transients naturally yield several solar masses of Ni-56. While PISNe have never been confirmed (though evidence has been reported in Gal-Yam et al. 2009, Cooke et al. 2012, and Kozyreva et al. 2018), they are thought to be a candidate for the SLSN-R subgroup of super-luminous supernovae (Gal-Yam, 2012a; Ren et al., 2012). Other broad and slowly evolving superluminous supernovae that do not require Ni-56 powering have been proposed (e.g. Nicholl et al., 2013, 2017a).

1.2.4 Intermediate Luminosity Optical Transients

In the energy gap between novae and supernovae, exists a rare transient explosion known as Intermediate Luminosity Optical Transients (ILOT) (e.g. NGC 300 OT2008-1, Berger et al. 2009). The physical mechanism of these objects is not well understood, but they have been modelled as either the eruption of red giants or interacting binary systems (see Pastorello & Fraser 2019 for a useful review and Kashi & Soker 2017 and references therein for examples of ILOTs).

1.2.5 Calcium-Rich Gap Transients

Calcium-rich gap transients (CART) (e.g. PTF11kmb, Lunnan et al. 2017) are a recently discovered transient class that have strong forbidden and permitted calcium lines in their spectra. The physical mechanism of these events is not well understood, but they are known to evolve much faster than average SNe Ia with rise times less than 15 days compared with 18 days for SNe Ia (see Figure 1.4). They have velocities of approximately 6000 to 10000 km s^{-1}, and have absolute magnitudes in the range -15.5 to -16.5 (a factor of 10 to 30 times fainter than SNe Ia) (Kasliwal et al., 2012; Sell et al., 2015). Their brightnesses lie in the "gap" between classical novae and supernovae and the advent of new high-cadence wide-field surveys has populated this region of parameter space. The morphology of their light curves is similar to SNe Ia-91bg or fast evolving SNe Ibc (Taubenberger, 2017). The first recognised CART was SN 2005E (Perets et al., 2010).

1.2.6 Kilonovae

Kilonovae are the mergers of either double neutron star (NS-NS) or black hole neutron star (BH-NS) binaries. Only one kilonova has been confirmed so far, where it was observed as the electromagnetic counterpart of the famous GW170817 gravitational wave event (Abbott et al., 2017c). The neutron-rich ejecta from the binary merger undergoes rapid neutron capture (r-process) nucleosynthesis to produce the Universe's rare heavy elements. The radioactive decay of these unstable nuclei power a rapidly evolving transient event (Metzger, 2017; Yu et al., 2018).

1.2.7 Active Galactic Nuclei

Active Galactic Nuclei (AGN) are the very bright nuclei of galaxies powered by centrally located supermassive black holes. Many AGN have rapidly varying light outputs changing on the scale of hours to weeks, suggesting that the emitting source must be on the order of light hours to light weeks in size (Padovani, 2017).

1.2.8 Tidal Disruption Events

Tidal Disruption Events (TDE) (e.g. ASASSN-15lh, Leloudas et al. 2016) occur when a star in the orbit of a massive black hole is pulled apart by the black hole's tidal forces. Some debris from the event is ejected at high speeds, while the remainder is swallowed by the black hole, resulting in a bright flare lasting up to a few years (Rees, 1988).

1.2.9 Fast Blue Optical Transients

Observations in the past decade have revealed the existence of a growing number of luminous, rare, and rapidly evolving extragalactic transients (e.g. Arcavi et al., 2016; Drout et al., 2014; Rest et al., 2018). These events have optical light curves that rise and decay on a timescale of only a few days and have recently been labelled as a new class of transient called Fast Blue Optical Transients (FBOTs) (e.g. Fang et al., 2019). The most famous of these events is AT2018cow, a transient that was detected very early (Smartt et al., 2018), and was followed up extensively by several telescopes (e.g. Ho et al., 2019; Kuin et al., 2019; Margutti et al., 2019; Perley et al., 2019; Prentice et al., 2018). AT2018cow is the brightest FBOT that has been observed, and has a peak luminosity of $\sim 4 \times 10^{44}$ erg s^{-1}, which exceeds the peak brightness of typical supernovae. The central engine causing FBOTs is unknown. However, recent modelling by Lyutikov & Toonen (2019) suggests that these events are fuelled by the merger of an ONeMg white dwarf star with another white dwarf resulting in electron-capture collapse to a neutron star. These events would be similar to one of the possible progenitors of SNe Ia, except in this case, the white dwarf merger fails to detonate and instead collapses via electron capture, leading to a much shorter time-scale transient.

1.2.10 Variables

Stellar variables are defined by their periodic light curves. RR Lyrae and Cepheids are two common periodic variable stars which gained particular attention after a relation was found between their period and luminosity (Leavitt, 1908; Leavitt & Pickering, 1912). This relation has allowed them to act as some of best distance indicators for low-z cosmology and as the anchors for the higher-z distance ladder. The period of Cepheids vary from a few days to a few months while RR Lyraes have much shorter periods - on the scale of a few hours.

Mira variables are another class of pulsating stars which are cool red giant stars in the asymptotic giant branch in the H-R diagram (Neilson et al., 2016). They are characterised by their periods in excess of 100 days.

Eclipsing binaries occur when one star eclipses another in a binary system. As it is estimated that approximately half of all stars are in binary systems, it is not uncommon to observe one of these stars eclipsing their binary partner in our line of sight. Their periods and light curve shape can vary significantly depending on the binary orbit, the angle of observation, and their stellar types. Examples of these events in the Kepler fields can be found in Maxted & Hutcheon (2018).

1.2.11 M Dwarf Flares

M dwarf stars have very energetic flares with luminosities around 10^{34} erg. Their flares are observed on a timescale of minutes to a few hours. They are caused by magnetic reconnection within the M dwarf and have strong observational signatures in large time domain surveys due to their frequency and luminosity (Hilton et al., 2010).

1.3 Transient classification

Typically, the classification of transients is based on their brightness and their optical spectrum taken near maximum light. In work not shown in this thesis, I have developed a spectral classification tool, called `DASH`[3] (published in Muthukrishna et al. (2019b)) that classifies supernovae into 17 different types. However, identifying supernovae based only on their photometry is a much more challenging task. As a transient evolves, its chemical composition changes, causing the observed spectrum to vary significantly with light curve phase (see Figure 1.6). A passband filter effectively produces a single flux measurement by integrating over the light that would be available in some wavelength range of a spectrum. Using multiple passband filters helps to retrieve more of this information, but it is not as informative as a spectrum would be. We describe briefly how light curves are obtained in the following subsection.

[3] https://astrodash.readthedocs.io/

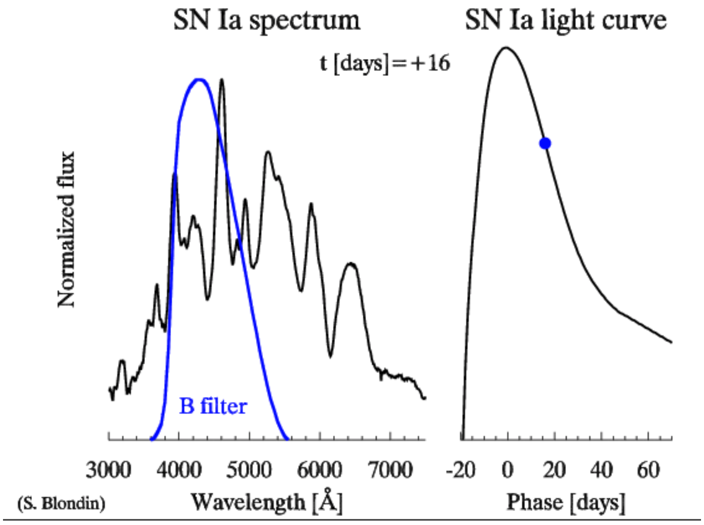


Fig. 1.6 An example light curve of a Type Ia supernovae with a corresponding spectrum in the B-Band. The light curve is shown at right, and the spectrum is shown at 16 days after maximum luminosity (left), where a B-Band filter is shown in blue. Figure taken from https://people.lam.fr/blondin.stephane/graphics/lcfig_anim.gif (Blondin & Tonry, 2007).

1.3.1 Light curves and detecting transients

Difference imaging is the primary method used for detecting transient events. It entails subtracting a new image from a real-time or archival image to detect a change in observed flux, and has been shown to be effective, even in fields that are crowded or associated with highly non-uniform unresolved surface brightnesses (Bond et al., 2001a; Tomaney & Crotts, 1996). Most transient surveys use this method, and “trigger” a transient event when there is a detection in a difference image that exceeds some S/N threshold. We show an example of the detection method in Figure 1.7. The total flux in a particular ellipse of the subtracted image is integrated to obtain a flux measurement.

A light curve consists of a series of flux measurements as a function of time in a particular passband. Time-domain surveys typically observe transients in a few different passbands. The change in luminosity in each filter over time - illustrated as light curves - is used to distinguish between different objects, and for understanding the nature of each object. Transients are defined by light curves whose duration is temporary (e.g. stellar explosions or collisions), while variables are defined by light curves which regularly oscillate in their brightness (e.g. RR Lyraes and Cepheids).

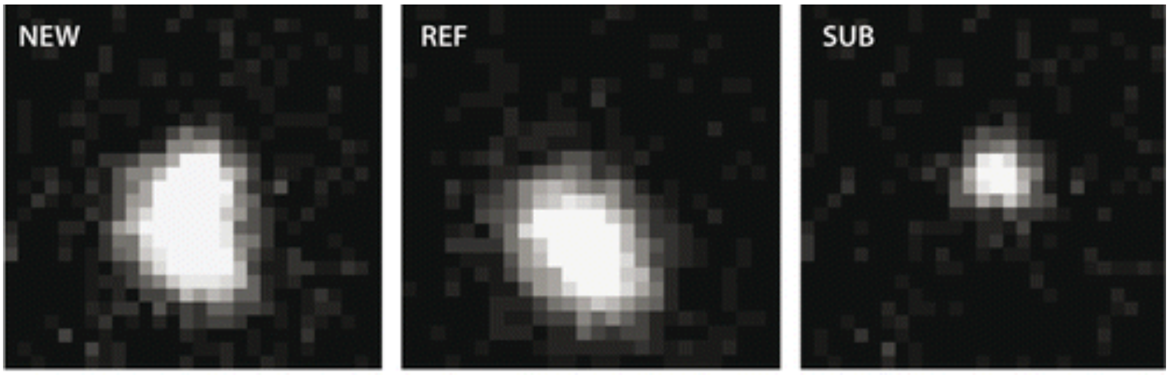


Fig. 1.7 An example of how transients are detected. Images separated by a range of times are subtracted from one another (NEW - REF = SUB). A reference image (REF) is selected before the transient phase of the event (ideally containing only the background and host galaxy), at some later time a new image (NEW) is observed, and the difference between these ideally leaves just the transient object. The shown example is taken from Figure 4 of Goobar & Leibundgut (2011)

1.4 Time-domain surveys

This thesis is largely motivated by the development of wide-field survey telescopes that will increase the transient catalogue by orders of magnitude. The Legacy Survey of Space and Time (LSST) is the most revolutionary of the upcoming surveys and is the primary motivation for this thesis. However, ongoing surveys such as the Zwicky Transient Facility (ZTF), the Panoramic Survey Telescope and Rapid Response System Telescope (Pan-STARRS, Chambers et al., 2016), Catalina Real-Time Transient Survey, (CRTS, Djorgovski et al., 2011)), the Planet Search Survey Telescope, (Dunham et al., 2004), and the Asteroid Terrestrial-impact Last Alert System (ATLAS Tonry et al., 2018) are already observing several hundreds to thousands of transient objects each day. In this thesis, we use data from ZTF and Pan-STARRS because LSST has not begun operations yet. We show our methods' success on these surveys to demonstrate that they may eventually be effectively applied to LSST.

1.4.1 LSST

The Vera C. Rubin Observatory Legacy Survey of Space and Time (LSST) will likely revolutionise our understanding of the transient universe. It is aiming to probe deeper, wider, and faster than any survey before it, and expects to yield up to 250,000 SNe Ia observations per year. This is at least a two order of magnitude increase from any previous survey. It will obtain larger samples of known transients, enabling detailed statistical studies of transient populations, and enable follow-up of specific transients. The sheer number of transients that it will observe means that it will also likely observe entirely new rarer classes of transients.

The Vera Rubin Observatory will have an effective aperture of 6.7 metres with a 9.6 deg^2 field of view (LSST Science Collaboration et al., 2009). The LSST will regularly scan 20,000 square degrees of the southern sky in 6 broad passband filters (*ugrizY*) from 0.35 to 1.1 microns. The transmission through these filters is illustrated in Figure 1.8. Each exposure reaches a depth of ~ 24 mag and it is expected to obtain 10 million transient alerts each night. Over the course of the 10-year survey, each pointing will be observed 2,000 times, following a cadence pattern tentatively defined by the `minion_1016`[4] OpSims cadence model (LSST Science Collaboration et al., 2009), to reach an integrated point-source depth of r~27.5 magnitudes. However, several alternate observing strategies that improve the survey characteristics for time-domain astrophysics are under active consideration (e.g. AltSched Rothchild 2018). In addition to the primary Wide-Fast-Deep survey, LSST will conduct several mini-surveys. The Deep Drilling Fields (DDF) survey, will aim to provide a higher cadence with a longer exposure in a smaller patch of the sky.

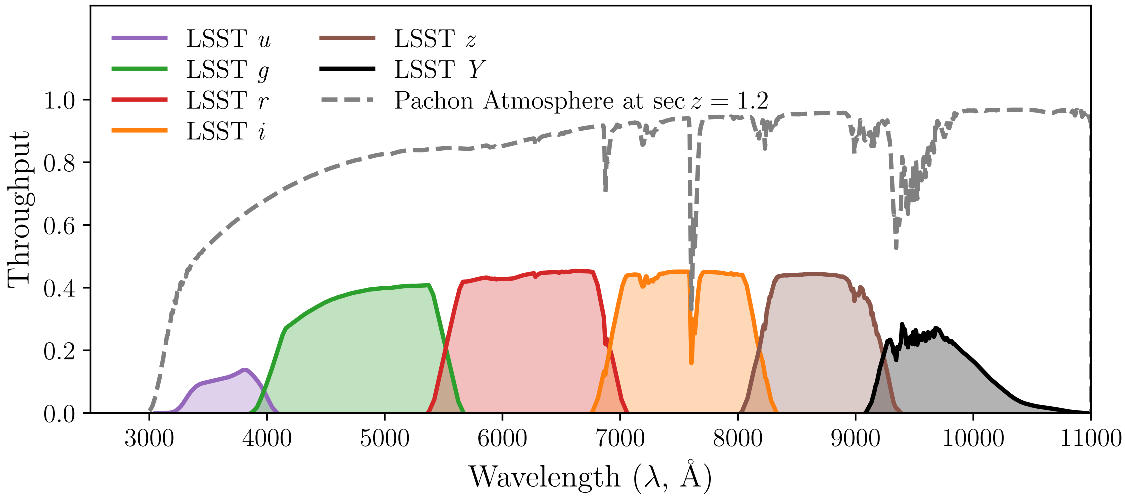


Fig. 1.8 Throughput of LSST filters as a function of wavelength. The atmospheric absorption lines are shown in grey and each passband is shown in different shaded regions.

[4] https://www.lsst.org/scientists/simulations/opsim/opsim-survey-data

1.4.2 ZTF

The Zwicky Transient Facility (ZTF, Bellm et al., 2019) is laying the groundwork for the future LSST. It is a wide-field synoptic survey that has been running since 2017. The public Mid Scale Innovations Program (MSIP) scans the sky every three nights in the g and r passbands with the 47 deg^2 imager mounted on the Palomar 48-inch telescope. It has a detection limit of $r = 20.5$ mag and issues around 10^5 transient alerts per night (about two orders of magnitude less than that expected from LSST). ZTF has probed the previously unexplored parameter space of shorter timescale transients and has discovered many new fast transients, CARTs, and TDEs. The large field of view also means that the ZTF has been essential in multi-messenger astronomy, whereby it has followed up the many potential optical counterparts to gravitational wave events.

1.4.3 Pan-STARRS

The Panoramic Survey Telescope and Rapid Response System Telescope (Pan-STARRS, Chambers et al. 2016) has recently released the first Petabyte-sized data release. It has observed several new transients as it uses its 1.4 gigapixel camera with a 7.1 deg^2 field of view to survey the northern sky. The Pan-STARRS1 Medium Deep Survey (PS1-MDS) covers a total sky-area of approximately 70 deg^2 in four passband filters $griz$. It observes each passband with a limiting magnitude of ~ 23.3 per visit with a cadence of approximately three days.

Recently, the Young Supernova Experiment (YSE, Jones et al., 2021b) is running on the Pan-STARRS telescopes and will survey 1500 deg^2 of the sky every three days. It aims to detect and follow-up transients at early times. YSE can discover transients as faint as ~ 21.1 mag in the gri and ~ 20.5 mag in the z passbands. The survey expects to find ~ 5000 new SNe per year and at least two SNe within 3 days of explosion per month. For YSE's success, it is imperative that they can perform fast and early classification. A motivation of this thesis is to identify YSE transients in real-time.

1.4.4 TiDES

The Time-Domain Extra Galactic Survey[5] (TiDES) is establishing a dedicated spectroscopic follow-up survey using 4MOST (de Jong et al., 2012) that will target live transients discovered photometrically by LSST. As LSST plans to probe wider and deeper than any survey before it, it is likely that it will discover new and rare transient events, and the opportunity to identify and follow-up these transients very early on should lead to exciting new transient physics. Another motivation of our work is to automatically supply TiDES and other spectroscopic surveys with follow-up candidates.

1.5 Transient Alert Brokers

The prospect of LSST with its huge data volumes has meant that dedicated *transient alert brokers* have been set up to ingest, process, and distribute the data from LSST to the astronomical community. Real-time transient alert brokers provide data management systems that continuously associate the detections from individual images into full light curves, characterise and classify them, as well as provide a filtering service to the astronomical community. Three of the leading brokers bidding for the LSST brokerage status are the USA-based ANTARES (Arizona-NOAO Temporal Analysis and Response to Events System) (Saha et al., 2016), the UK-based Lasair (Smith et al., 2019), and the Chile-based ALerCE (Automatic Learning for the Rapid Classification of Events, Förster et al., 2020). To prepare for LSST, they are each currently processing alerts from ongoing surveys such as ZTF and Pan-STARRS. The primary goal of these brokers is to streamline the classification and follow-up paradigm of transients. This pipeline flows from the detection of transients in the sky, to classification, and then to distributing follow-up alerts to the astronomical community, so that scientifically interesting objects can be followed up spectroscopically and in different wavelength bands.

These transient brokers require real-time classifiers capable of quickly identifying potentially interesting transients for follow-up. A key motivation of this thesis is to develop classifiers and anomaly detection frameworks for use by these brokers.

[5] https://www.4most.eu/cms/surveys/extragalactic/

1.6 PLAsTiCC

One of the key issues with developing classifiers for upcoming transient surveys is that we do not have a training set that is representative of each survey's data stream, cadence and noise properties. To meet this challenge for LSST, the PLAsTiCC (Photometric LSST Astronomical Time-series Classification Challenge) collaboration has set up a community-wide challenge to develop algorithms to classify astronomical transients. Its focus lies in preparing some of the required software infrastructure for the upcoming LSST transient survey. In doing so, it has developed model light curves simulating the expected transient catalogue that will be observed by LSST. With the use of supernova analysis tools such as SNANA (Kessler et al., 2009), transient templates, and model light curves from leading transient experts, a range of transient models have been developed and included in the PLAsTiCC dataset (PLASTICC Team & PLASTICC Modelers, 2019). This modelling procedure is illustrated in Figure 1.9 and detailed in Kessler et al. (2019). The simulated transients were provided to the community as part of the classification challenge on Kaggle[6]. The results of the challenge are described in Hložek et al. (2020).

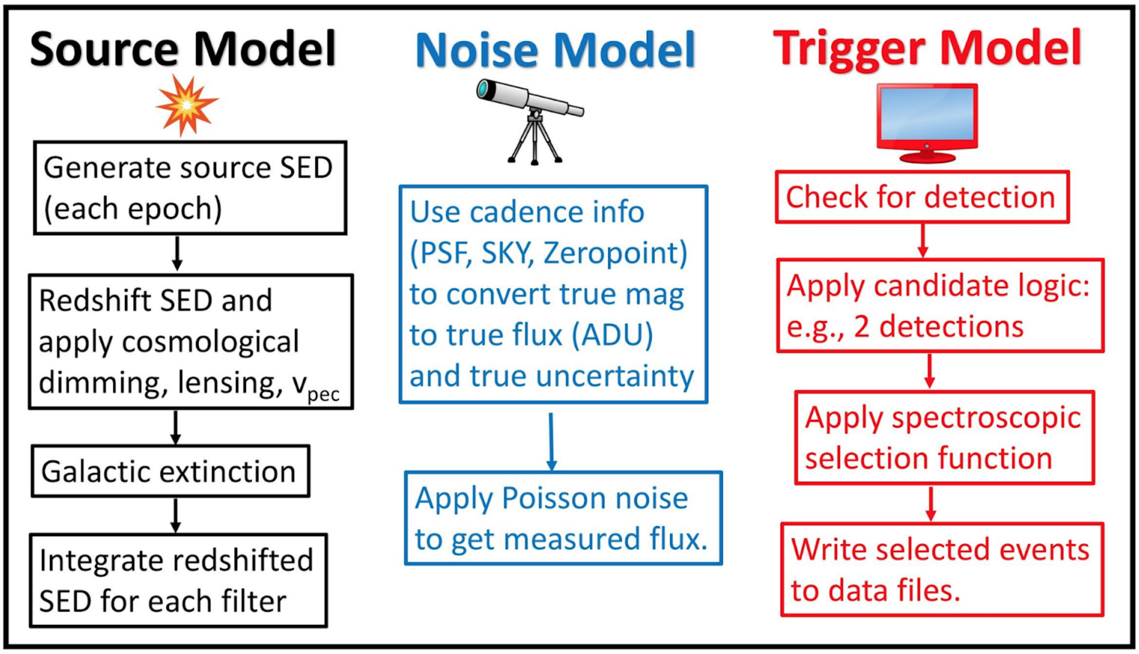

Source Model
Generate source SED (each epoch)
Redshift SED and apply cosmological dimming, lensing, v_{pec}
Galactic extinction
Integrate redshifted SED for each filter
Noise Model
Use cadence info (PSF, SKY, Zeropoint) to convert true mag to true flux (ADU) and true uncertainty
Apply Poisson noise to get measured flux.
Trigger Model
Check for detection
Apply candidate logic: e.g., 2 detections
Apply spectroscopic selection function
Write selected events to data files.

Fig. 1.9 The modelling procedure of the PLAsTiCC simulations. Figure taken from Figure 13 of Kessler et al. (2019).

[6] https://www.kaggle.com/c/PLAsTiCC-2018

Throughout this thesis, I worked closely with the PLAsTiCC team to help validate the simulations before they were released in the Kaggle challenge. I then used the simulation software to create ZTF-like simulations that we have used extensively throughout this thesis.

1.7 Overview

The large influx of transients from upcoming surveys means that it is no longer feasible to visually examine each object for follow-up purposes. The expected data deluge brings new challenges. In this thesis, I detail novel methods of dealing with two important issues brought about by these new large-scale surveys. The first challenge, is how can we categorise the influx of transient objects into classes of interest, and importantly, how do we do this early so that we can follow-up specific objects in real-time? The second challenge is the question of serendipity. With such high volumes of data, are we still going to be able to make serendipitous discoveries? Serendipity has often led to major new breakthroughs in transient astrophysics. For example, the incidental discovery of pulsars by Jocelyn Bell, or the Fast Radio Bursts (FRBs) discovered by Duncan Lorimer were both serendipitous discoveries that may have been buried away if the data volumes were too large to deal with. Thus, the second challenge we deal with in this thesis is whether we can automate serendipity by automatically flagging anomalous transients.

In Chapter 2 of this thesis, I detail my development of a real-time classifier of transients, called RAPID, and apply it to ZTF-like simulations. In Chapter 3, I develop two new frameworks for real-time anomaly detection of transients and apply it to both ZTF-like simulations and real data from the ZTF MSIP. Since supervised classification is not capable of finding new classes of objects, anomaly detection is required to help identify novelties in the dataset. In Chapter 4, I improve RAPID to be more applicable to real datasets. I train classifiers to perform well on Pan-STARRS and ZTF light curves. Finally in Chapter 5, I discuss conclusions and look forward to future directions of my work.

Chapter 2

RAPID: Early classification of explosive transients using deep learning

This chapter is adapted from the paper published in Muthukrishna et al. (2019a). The coauthors of that paper have only contributed indirectly through helping to secure the data and by providing useful feedback on the analysis, results, and writing of the manuscript.

2.1 Summary

In this chapter, I present `RAPID` (Real-time Automated Photometric IDentification), a novel time-series classification tool capable of automatically identifying transients from within a day of the initial alert, to the full lifetime of a light curve. Using a deep recurrent neural network with Gated Recurrent Units (GRUs), we present the first method specifically designed to provide early classifications of astronomical time-series data, typing 12 different transient classes. Our classifier can process light curves with any phase coverage, and it does not rely on deriving computationally expensive features from the data, making `RAPID` well-suited for processing the millions of alerts that ongoing and upcoming wide-field surveys such as the Zwicky Transient Facility (ZTF), and the Legacy Survey of Space and Time (LSST) will produce. The classification accuracy improves over the lifetime of the transient as more photometric data becomes available, and across the 12 transient classes, we obtain an average area under the receiver operating characteristic curve of 0.95 and 0.98 at early and late epochs, respectively. We demonstrate `RAPID`'s ability to effectively provide early classifications of observed transients from the ZTF data stream. We have made `RAPID`

available as an open-source software package[1] for machine learning-based alert-brokers to use for the autonomous and quick classification of several thousand light curves within a few seconds.

2.2 Introduction

Observations of the transient universe have led to some of the most significant discoveries in astronomy and cosmology. From the use of Cepheids and type Ia supernovae (SNe Ia) as standardisable candles for estimating cosmological distances, to the recent detection of a kilonova event as the electromagnetic counterpart of GW170817, the transient sky continues to provide exciting new astronomical discoveries.

In the past, transient science has had significant successes using visual classification by experienced astronomers to rank interesting new events and prioritize spectroscopic follow-up. Nevertheless, the visual classification process inevitably introduces latency into follow-up studies, and spectra for many objects are obtained several days to weeks after the initial detection. Existing and upcoming wide-field surveys and facilities will produce several million transient alerts per night, e.g. the Legacy Survey of Space and Time (LSST, Ivezić et al., 2019), the Dark Energy Survey (DES, Dark Energy Survey Collaboration et al., 2016), the Zwicky Transient Facility (ZTF, Bellm, 2014), the Catalina Real-Time Transient Survey (CRTS, Djorgovski et al., 2011), the Panoramic Survey Telescope and Rapid Response System (Pan-STARRS, Chambers et al., 2016), the Asteroid Terrestrial-impact Last Alert System (ATLAS, Tonry et al., 2018), and the Planet Search Survey Telescope (PSST, Dunham et al., 2004). This unprecedented number means that it will be possible to obtain early-time observations of a large sample of transients, which in turn will enable detailed studies of their progenitor systems and a deeper understanding of their explosion physics. However, with this deluge of data comes new challenges, and individual visual classification for spectroscopic follow-up is utterly unfeasible.

Developing methods to automate the classification of photometric data is of particular importance to the transient community. In the case of SNe Ia, cosmological analyses to measure the equation of state of the dark energy w and its evolution requires large samples with low contamination. The need for a high purity sample necessitates expensive spectroscopic observations to determine the type of each candidate, as classifying SNe Ia

[1] https://astrorapid.readthedocs.io

based on sparse light curves[2] runs the risk of contamination with other transients, particularly type Ibc supernovae. Even with human inspection, the differing cadence, observer frame passbands, photometric properties, and contextual information of each transient light curve constitute a complex mixture of sparse information, which can confound our visual sense, leading to potentially inconsistent classifications. This failing of visual classification, coupled with the large volumes of data, necessitates a streamlined automated classification process. This is our motivation for the development of our deep neural network (DNN) for Real-time Automated Photometric IDentification (`RAPID`), the focus of this work.

2.2.1 Previous Work on Automated Photometric Classification

In 2010, the Supernova Photometric Classification Challenge (SNPhotCC, Kessler et al., 2010a,b), in preparation for the Dark Energy Survey (DES), spurred the development of several innovative classification techniques. The goal of the challenge was to determine which techniques could distinguish SNe Ia from several other classes of supernovae using light curves simulated with the properties of the DES. The techniques used for classification varied widely, from fitting light curves with a variety of templates (Sako et al., 2008), to much more complex methodologies that use semi-supervised learning approaches (Richards et al., 2012) or parametric fitting of light curves (Karpenka et al., 2013). A measure of the value of the SNPhotCC is that the dataset is still used as the reference standard to benchmark contemporary supernova light curve classification schemes, such as Bloom et al. (2012); Charnock & Moss (2017); Ishida & de Souza (2013); Lochner et al. (2016); Narayan et al. (2018); Pasquet et al. (2019); Revsbech et al. (2018); Richards et al. (2012).

Nearly all approaches to automated classification developed using the SNPhotCC dataset have either used empirical template-fitting methods (Sako et al., 2008, 2011) or have extracted features from supernova light curves as inputs to machine learning classification algorithms (Karpenka et al., 2013; Lochner et al., 2016; Möller et al., 2016; Narayan et al., 2018; Newling et al., 2011; Sooknunan et al., 2018). Lochner et al. (2016) used a feature-based approach, computing features using either parametric fits to the light curve, template fitting with SALT2 (Guy et al., 2007), or model-independent wavelet decomposition of the data. These features were independently fed into a range of machine learning architectures including Naive Bayes, k-nearest neighbours, multilayer perceptrons, support vector machines, and boosted decision

[2] We define light curves as photometric time-series measurements of a transient in multiple passbands. *Full light curves* refer to time series of objects observed over nearly their full transient phase. We refer to *early light curves* as time series observed up to 2 days after a trigger alert, defined in §2.3.2.

trees (see Lochner et al. 2016 for a brief description of these) and were used to classify just three broad supernova types. The work concluded that the non-parametric feature extraction approaches were most effective for all classifiers, and that boosted decision trees performed most effectively. Surprisingly, they further showed that the classifiers did not improve with the addition of redshift information. These previous approaches share two characteristics:

1. they are largely tuned to discriminate between different classes of supernovae, and
2. they require the full phase coverage of each light curve for classification.

Both characteristics arise from the SNPhotCC dataset. As it was developed to test photometric classification for an experiment using SNe Ia as cosmological probes, the training set represented only a few types of non-SNe Ia that were likely contaminants, whereas the transient sky is a menagerie. Additionally, SNPhotCC presented astronomers with full light curves, rather than the streaming data that is generated by real-time transient searches, such as ZTF. While previous methods can be extended with a larger, more diverse training set, the second characteristic they share is a more severe limitation. Requiring complete phase coverage of each light curve for classification (e.g. Lochner et al. 2016) is a fundamental design choice when developing the architecture for automated photometric classification, and methods cannot trivially be re-engineered to work with sparse data.

2.2.2 Early Classification

While retrospective classification after the full light curve of an event has been observed is useful, it also limits the scientific questions that can be answered about these events, many of which exhibit interesting physics at early-times. Detailed observations, including high-cadence photometry, time-resolved spectroscopy, and spectropolarimetry, shortly after the explosion provides insights into the progenitor systems that power the event and hence improves our understanding of the objects' physical mechanism. Therefore, ensuring a short latency between a transient detection and its follow-up is an important scientific challenge. Thus, a key goal of our work on `RAPID` has been to develop a classifier capable of identifying transient types within 2 days of detection. We refer to photometric typing with observations obtained in this narrow time-range as early classification.

The discovery of the electromagnetic counterpart (Abbott et al., 2017c; Arcavi et al., 2017; Coulter et al., 2017; Soares-Santos et al., 2017a) from the binary neutron star merger event, GW170817, has thrown the need for automated photometric classifiers capable of identifying exotic events from sparse data into sharp relief. As we enter the era of multi-messenger astrophysics, it will become evermore important to decrease the latency between the detection and follow-up of transients. While the massive effort to optically follow up GW170817 was heroic, it involved a disarray of resource coordination. With the large volumes of interesting and unique data expected by surveys such as LSST ($\sim 10^7$ alerts per night), the need to streamline follow-up processes is crucial. The automated early classification scheme developed in this work alongside the new transient brokers[3] such as ALeRCE[4] (Förster et al., 2020), LASAIR[5] (Smith et al., 2019), ANTARES[6] (Saha et al., 2016) are necessary to ensure organized and streamlined follow-up of the high density of exciting transients in upcoming surveys.

There have been a few notable efforts at early-time photometric classification, particularly using additional contextual data. Sullivan et al. (2006) successfully discriminated between SNe Ia and core-collapse SNe in the Supernova Legacy Survey using a template fitting technique on only two to three epochs of multiband photometry. Poznanski et al. (2007) similarly attempted to distinguish between SNe Ia and core-collapse SNe using only single-epoch photometry along with a photometric redshift estimate from the probable host-galaxy. A few contemporary techniques such as Foley & Mandel (2013) and the `sherlock` package[7] use only host galaxy and contextual information with limited accuracy to predict transient classes (e.g. the metallicity of the host galaxy is correlated with supernova type).

The most widely used scheme for classification is `pSNid` (Sako et al., 2008, 2011). It has been used by the Sloan Digital Sky Survey and DES (D'Andrea et al., 2018) to classify pre-maximum and full light curves into 3 supernova types (SNIa, SNII, SNIbc). For each class, it has a library of template light curves generated over a grid of parameters (redshift, dust extinction, time of maximum, and light curve shape). To classify a transient, it performs an exhaustive search over the templates of all classes. It identifies the class of the template that best matches (with minimum χ^2) the data and computes the Bayesian evidence by marginalizing the likelihood over the parameter space. The latest version employs computationally-expensive nested sampling to compute the evidence. Therefore, the main computational burden (which increases with the number of templates used) is incurred

[3] Transient brokers are automated software systems that manage the real-time alert streams from transient surveys such as ZTF and LSST. They sift through, characterize, annotate and prioritise events for follow-up.

[4] http://alerce.science

[5] https://lasair.roe.ac.uk

[6] https://antares.noao.edu/

[7] https://github.com/thespacedoctor/sherlock

every time it used to predict the class of each new transient. As new data arrives, this cost is multiplied as the procedure must be repeated each time the classifications are updated.

In contrast, `RAPID` covers a much broader variety of transient classes and learns a function that directly maps the observed photometric time series onto these transient class probabilities. The main computational cost is incurred only once, during the training of the DNN, while predictions obtained by running new photometric time series through the trained network are very fast. Because of this advantage, updating the class probabilities as new data arrives is trivial. Furthermore, we specifically designed our RNN architecture for temporal processes. In principle, it is able to save the information from previous nights so that the additional cost to update the classifications as new data are observed is only incremental. These aspects make `RAPID` particularly well-suited to the large volume of transients that new surveys such as LSST will observe.

2.2.3 Deep Learning in Time-Domain Astronomy

Improving on previous feature-based classification schemes, and developing methods that exhibit good performance even with sparse data, requires new machine learning architectures. Advanced neural network architectures are non-feature-based approaches that have recently been shown to have several benefits such as low computational cost, and being robust against some of the biases that can afflict machine learning techniques that require "expert-designed" features (Aguirre et al., 2018; Charnock & Moss, 2017; Moss, 2018; Naul et al., 2018). The use of Artificial Neural Networks (ANN, McCulloch & Pitts, 1943) and deep learning, in particular, has seen dramatic success in image classification, speech recognition, and computer vision, outperforming previous approaches in many benchmark challenges (Krizhevsky et al., 2012; Razavian et al., 2014; Szegedy et al., 2015).

In time-domain astronomy, deep learning has recently been used in a variety of classification problems including variable stars (Hinners et al., 2018; Naul et al., 2018), supernova spectra (Muthukrishna et al., 2019b), photometric supernovae (Charnock & Moss, 2017; Möller & de Boissière, 2020; Moss, 2018; Pasquet et al., 2019), and sequences of transient images (Carrasco-Davis et al., 2018). A particular class of ANNs known as Recurrent Neural Networks (RNNs) are particularly suited to learning sequential information (e.g. time-series data, speech recognition, and natural language problems). While ANNs are often feed-forward (e.g. convolutional neural networks and multilayer perceptrons), where information passes through the layers once, RNNs allow for cycling of information through the layers.

They are able to encode an internal representation of previous epochs in time-series data, which along with real-time data, can be used for classification.

A variant of RNNs known as Long Short Term Memory Networks (LSTMs, Hochreiter & Schmidhuber, 1997) improve upon standard RNNs by being able to store long-term information, and have achieved state-of-the-art performance in several time-series applications. In particular, they revolutionized speech recognition, outperforming traditional models (Fernández et al., 2007; Hannun et al., 2014; Li & Wu, 2015) and have very recently been used in the trigger word detection algorithms popularized by *Apple's Siri*, *Microsoft's Cortana*, *Google's voice assistant*, and *Amazon's Echo*. Naul et al. (2018) and Hinners et al. (2018) have had excellent success in variable star classification. Charnock & Moss (2017) applied the technique to supernova classification. They used supernova data from the SNPhotCC and fed the multiband photometric full lightcurves into their LSTM architecture to achieve high SNIa vs non-SNIa binary classification accuracies. Moss (2018) recently followed this up on the same data with a novel approach applying a new phased-LSTM (Neil et al., 2016) architecture. These approaches have the advantage over previous supernova photometric classifiers of not requiring computationally-expensive and user-defined (and hence, possibly biased) feature engineering processes.

While the manuscript that this chapter is based on was under review, Möller & de Boissière (2020) released a similar algorithm for photometric classification of a range of supernova types. It uses a recurrent neural network architecture based on BRNNs (Bayesian Recurrent Neural Networks) and does not require any feature extraction. At a similar time, Pasquet et al. (2019) released a package for full light curve photometric classification based on Convolutional Neural Networks, again able to use photometric light curves without requiring feature engineering processes. These approaches made use of datasets adapted from the SNPhotCC, using light-curves based on the observing properties of the Dark Energy Survey and with a smaller variety of transient classes than the PLAsTiCC-based training set used in our work. Pasquet et al. (2019) use a framework that is very effective for full light curve classification, but is not well suited to early or partial light curve classification. Möller & de Boissière (2020), on the other hand, is one of the first approaches able to classify partial supernova light curves using a single bi-directional RNN layer, and achieve accuracies of up to $96.92 \pm 0.26\%$ for a binary SNIa vs non-SNIa classifier. While their approach is similar, the type of RNN, neural network architecture, dataset, and focus on supernovae differs from the work presented in this chapter. `RAPID` is focused on early and real-time light curve classification of a wide range of transient classes to identify interesting objects early, rather than on full light curve classification for creating pure photometric samples for SNIa cosmology. We are currently applying our work to the real-time alert stream from the ongoing ZTF survey through the ANTARES broker, and plan to develop the technique further for use on the LSST alert stream.

2.2.4 Overview

In this chapter, we build upon the approach used in Charnock & Moss (2017). We develop `RAPID` using a deep neural network (DNN) architecture that employs a very recently improved RNN variation known as Gated Recurrent Units (GRUs, Cho et al., 2014). This novel architecture allows us to provide real-time, rather than only full light curve, classifications.

Previous RNN approaches (including Charnock & Moss (2017); Hinners et al. (2018); Möller & de Boissière (2020); Moss (2018); Naul et al. (2018)) all make use of bi-directional RNNs that can access input data from both past and future frames relative to the time at which the classification is desired. While this is effective for full light curve classification, it does not suit the real-time, time-varying classification that we focus on in this work. In real-time classification, we can only access input data previous to the classification time. Therefore, to respect causality, we make use of uni-directional RNNs that only take inputs from time-steps previous to at any given classification time. `RAPID` also enables multi-class and multi-passband classifications of transients as well as a new and independent measure of transient explosion dates. We further make use of a new light curve simulation software developed by the recent Photometric LSST Astronomical Time-series Classification Challenge (PLAsTiCC, The PLAsTiCC team et al., 2018).

In Section 2.3 we discuss how we use the PLAsTiCC models with the SNANA software suite (Kessler et al., 2009) to simulate photometric light curves based on the observing characteristics of the ZTF survey, and describe the resulting dataset along with our cuts, processing, and modelling methods. In Section 2.4, we frame the problem we aim to solve and detail the deep learning architecture used for `RAPID`. In Section 2.5, we evaluate our classifier's performance with a range of metrics, and in Section 2.6 we apply the classifier to observed data from the live ZTF data stream. An illustration of the different sections of this chapter and their connections is shown in Fig. 2.1. Finally, in Section 2.7, we compare `RAPID` to a feature-based classification technique we implemented using an advanced Random Forest classifier improved from Narayan et al. (2018) and based on Lochner et al. (2016). We present conclusions in section 2.8.

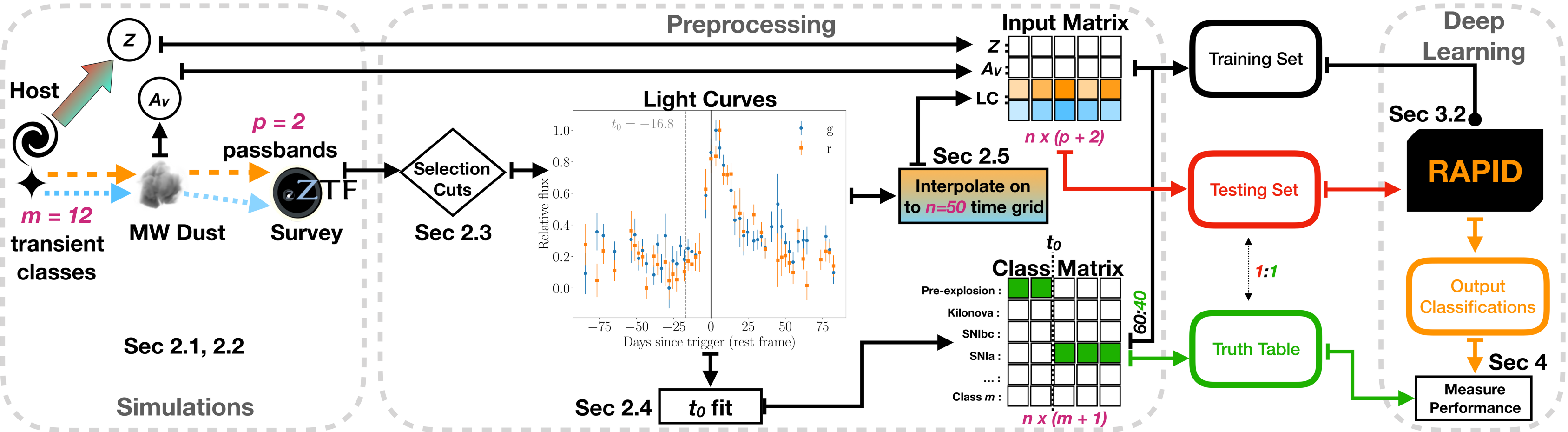


Fig. 2.1 Schematic illustrating the preprocessing, training set preparation, and classification processes described throughout this chapter.

2.3 Data

2.3.1 Simulations

One of the key challenges with developing classifiers for upcoming transient surveys is the lack of labelled samples that are appropriate for training. Moreover, even once a survey commences, it can take a significant amount of time to accumulate a well-labelled sample that is large enough to develop robust learning algorithms. To meet this difficulty for LSST, the PLAsTiCC collaboration has developed the infrastructure to simulate light curves of astrophysical sources with realistic sampling and noise properties. This effort was one component of an open-access challenge to develop algorithms that classify astronomical transients. By adapting supernova analysis tools such as SNANA (Kessler et al., 2009) to process several models of astrophysical phenomena from leading experts, a range of new transient behavior included in the PLAsTiCC dataset. The challenge has recently been released to the public on Kaggle[8] (The PLAsTiCC team et al., 2018) along with the metric framework to evaluate submissions to the challenge (Malz et al., 2018). The PLAsTiCC models are the most comprehensive enumeration of the transient and variable sky available at present.

We use the PLAsTiCC transient class models and the simulation code developed in Kessler et al. (2019) to create a simulated dataset that is representative of the cadence and observing properties of the ongoing public "Mid Scale Innovations Program" (MSIP) survey at the ZTF (Bellm, 2014). This allows us to compare the validity of the simulations with the live ZTF data stream, and apply our classifier to it as illustrated in Section 2.6.

2.3.1.1 Zwicky Transient Facility

ZTF is the first of the new generation of optical synoptic survey telescopes and builds upon the infrastructure of the Palomar Transient Factory (PTF, Rau et al., 2009). It employs a 47 square degree field-of-view camera to scan more than 3750 square degrees an hour to a depth of 20.5 - 21 mag (Graham et al., 2019). It is a precursor to the LSST and will be the first survey to produce one million alerts a night and to have a trillion row data archive. To prepare for this unprecedented data volume, we build an automated classifier trained on a large simulated ZTF-like dataset that contains a labelled sample of transients.

[8] https://www.kaggle.com

We obtained logs of ZTF observing conditions (E. Bellm, private communication) and photometric properties (zeropoints, FWHM, sky brightness etc.), and a collaborator, Dr. Rahul Biswas, converted these into a library suitable for use with SNANA. SNANA simulates millions of light curves for each model, following a class-specific luminosity function prescription within the ZTF footprint. The sampling and noise properties of each observation on each light curve is set to reflect a random sequence from within the observing conditions library. The simulated light curves thus mimic the ZTF observing properties with a median cadence of 3 days in the g and r passbands. As ZTF had only been operating for four months when we constructed the observing conditions library, it is likely that our simulations are not fully representative of the survey. Nevertheless, this procedure is more realistic than simulating the observing conditions entirely, as we would have been forced to do if we had developed `RAPID` for LSST or *WFIRST*. We verified that the simulated light curves have similar properties to observed transient sources detected by ZTF that have been announced publicly. The dataset consists of a labelled set of 48029 simulated transients evenly distributed across a range of different classes. An example of a simulated light curve from each class is shown in Fig. 2.2.

The PLAsTiCC models are the largest collection of transient models assembled to date. The specific models used in the simulations derived from Kessler et al. (2019) are SNIa-norm: Guy et al. (2010); Kessler et al. (2013); Pierel et al. (2018), SNIbc: Guillochon et al. (2018a); Kessler et al. (2010b); Pierel et al. (2018); Villar et al. (2017), SNII: Guillochon et al. (2018a); Kessler et al. (2010b); Pierel et al. (2018); Villar et al. (2017), SNIa-91bg: (Galbany et al. in prep.), SNIa-x: Jha (2017), pointIa: Shen et al. (2010), Kilonovae: Kasen et al. (2017), SLSN: Guillochon et al. (2018a); Kasen & Bildsten (2010); Nicholl et al. (2017b), PISN: Guillochon et al. (2018a); Kasen et al. (2011); Villar et al. (2017), ILOT: Guillochon et al. (2018a); Villar et al. (2017), CART: Guillochon et al. (2018a); Kasliwal et al. (2012); Villar et al. (2017), TDE: Guillochon et al. (2018a); Mockler et al. (2019); Rees (1988). In section 1.2, we describe these transient classes.

Each simulated transient dataset consists of a time series of flux and flux error measurements in the g and r ZTF bands, along with sky position, Milky Way dust reddening, a host-galaxy redshift, and a photometric redshift. The models used in PLAsTiCC were extensively validated against real observations by several complementary techniques, as described by Hložek et al. (2020). We split the total set of transients into two parts: 60% for the *training set* and 40% for the *testing set*. The *training set* is used to train the classifier to

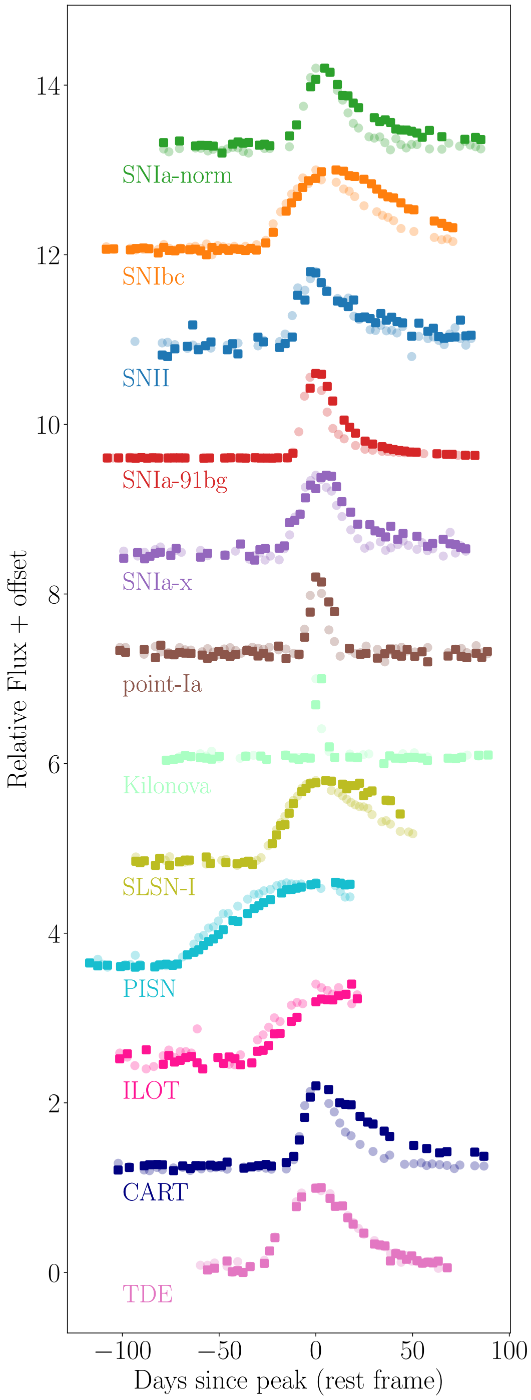


Fig. 2.2 The light curves of one example transient from each of the 12 transient classes is plotted with an offset. We have only plotted transients with a high signal-to-noise and with a low simulated host redshift ($z < 0.2$) to facilitate comparison of light curve shape between the classes. The dark-coloured square markers plots the r band light curves of each transient, while the lighter-coloured circle markers are the g band light curves of each transient.

identify the correct transient class, while the *testing set* is used to test the performance of the classifier.

2.3.2 Trigger for Issuing Alerts

The primary method used for detecting transient events is to subtract real-time or archival data from a new image to detect a change in observed flux. This is known as *difference imaging* (see section 1.3.1, and has been shown to be effective, even in fields that are crowded or associated with highly non-uniform unresolved surface brightness (Bond et al., 2001b; Tomaney & Crotts, 1996). Most transient surveys, including ZTF, use this method, and 'trigger' a transient event when there is a detection in a difference image that exceeds a 5σ signal-to-noise (S/N) threshold. Throughout this work, we use *trigger* to identify this time of detection. We refer to early classification as classification made within 2 days of this trigger, and full classification as classifications made after 40 days since trigger.

2.3.3 Selection Criteria

To create a good and clean training sample, we made a number of cuts before processing the light curves. The selection criteria is described as follows.

$z < 0.5$ **and** $z \neq 0$
: Firstly, we cut objects with host-galaxy redshifts $z = 0$ or $z > 0.5$ such that all galactic objects and any higher redshift objects were removed as these candidates are too faint to be useful for the early-time progenitor studies that motivated the development of this classifier in the first place. While our work relies on knowing the redshift of each transient, in this low redshift range, we should be able to obtain a redshift from the host galaxy from existing catalogs.

Sufficient data in the early light curve Next, we ensured that the selected light curves each had at least three measurements before trigger, and at least two of these were in different passbands. Even if these measurements were themselves insufficient to cause a trigger, they help establish a baseline flux. This cut therefore removes objects that triggered immediately after the beginning of the observing season, as these are likely to be unacceptably windowed.

$b > 15^\circ$ **and** $b < -15^\circ$

Any object in the galactic plane, with latitude $-15^\circ < b < 15^\circ$, was also cut from the dataset because our analysis only considers extragalactic transients.

Selected only transient objects

Finally, while the PLAsTiCC simulations included a range of variable objects, including AGN, RR Lyrae, M-dwarf flares, and Eclipsing Binary events, we removed these from the simulated dataset. The long-lived variable candidates are Galactic and will likely be identified to a very high completeness. Similarly, the AGN will likely either be easily be removed from known AGN catalogues or will be identified to a very high completeness over the redshift range under consideration. Therefore, these classes are unlikely to be misidentified as a class of astrophysical interest for early-time studies.

2.3.4 Preprocessing

Arguably, one of the most important aspects in an effective learning algorithm is the quality of the training set. In this section we discuss efforts to ensure that the data is processed in a uniform and systematic way before we train our DNN.

The light curves are measured in flux units, as is expected for the ZTF difference imaging pipeline. The simulations have a significant fraction of the observations being 5-10 sigma outliers. These outliers are intended to replicate the difference image analysis artifacts, telescope CCD deficiencies, and cosmic rays seen in observational data. We perform ‘sigma clipping’ to reject these outliers. We do this by rejecting photometric points with flux uncertainties that are more than 3σ from the mean uncertainty in each passband, and iteratively repeat this clipping 5 times. Next, we correct the light curves for interstellar extinction using the reddening function of Fitzpatrick (1999). We assume an extinction law, $R_V = 3.1$, and use the central wavelength of each ZTF filter to de-redden each light curve listed as follows[9]:

$$g\text{: 4767 Å, } r\text{: 6215 Å.}$$

Following this, we account for cosmological time dilation using the host redshifts, z, and convert the observer frame time since trigger to a rest-frame time interval,

$$t = (T_{\text{obs}} - T_{\text{trigger}})/(1+z), \tag{2.1}$$

[9] We use the `extinction` code: https://extinction.readthedocs.io

where capital T refers to an observer frame time in MJD and lowercase t refers to a rest-frame time interval relative to trigger. We define trigger as the epoch at which the ZTF difference imaging detects a 5σ threshold change in flux.

We then calculate the luminosity distance, $d_L(z)$, to each transient using the known host redshift and assuming a ΛCDM cosmology with $\Omega_M = 0.3$, $\Omega_\Lambda = 0.7$ and $H_0 = 70$. We correct the flux for this distance by multiplying each flux by d_L^2 and scaling by some normalizing factor, norm $= 10^{18}$, to keep the flux values in a good range for floating-point machine precision. A measure for the distance-corrected flux, which is proportional to luminosity, is

$$L_{\text{data}}(t) = (F(t) - F(t)_{\text{med}}) \cdot \frac{d_L^2(z)}{\text{norm}}, \tag{2.2}$$

where $F(t)$ is the raw flux value and $F(t)_{\text{med}}$ is the median value of the raw flux points that were observed before the trigger. This median value is representative of the background flux. Even for objects observed by a single survey, with a common set of passbands on a common photometric system, comparing the fluxes of different sources in the same rest-frame wavelength range requires that the light curve photometry be transformed into a common reference frame, accounting for the redshifting of the sources. However, this k-correction (Hogg et al., 2002) requires knowledge of the underlying spectral energy distribution (SED) of each source, and therefore its type — the goal of this work. Therefore, we have not k-corrected these data into the rest-frame, and hence, L_{data} cannot be considered the true rest-frame luminosity in each passband.

2.3.4.1 Modeling the Early Light Curve

The ability to predict the class of an object as a function of time is one of the main advantages of `RAPID` over previous work. Critical to achieving this is determining the epoch at which transient behaviour begins (usually the date of explosion) so that we can *teach* our DNN what a pre-explosion looks like. Basic geometry suggests that a single explosive event should evolve in flux proportional to the square of time (Arnett, 1982). While future work might try to fit a power law, we are limited by the sparse and noisy data in the early light curve. Therefore, we model the pre-maximum part of each light curve in each passband, λ, with a simple t^2 fit as follows,

$$L^{\lambda}_{\text{mod}}(t; t_0, a^{\lambda}, c^{\lambda}) = \left[a^{\lambda}(t - t_0)^2\right] \cdot H(t - t_0) + c^{\lambda}, \tag{2.3}$$

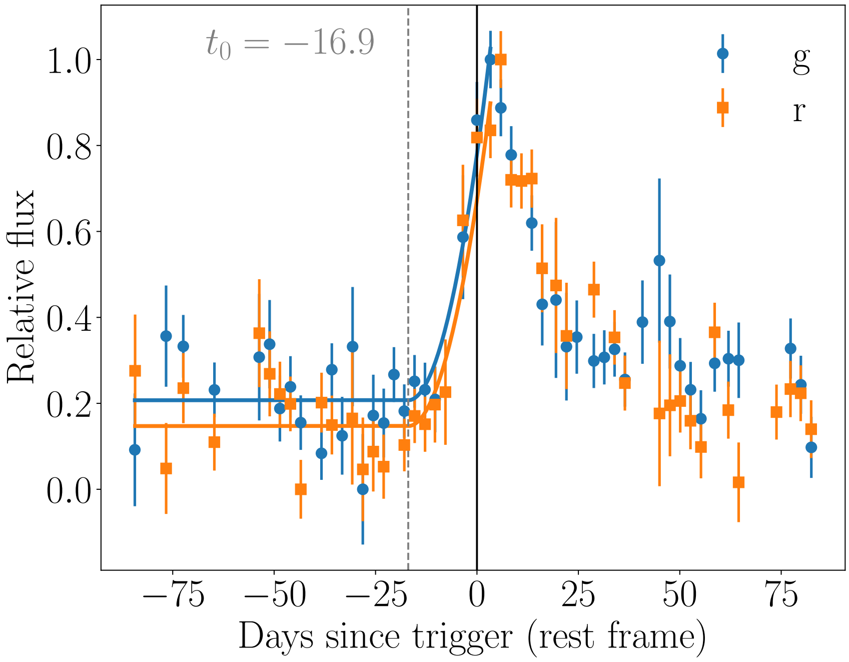


Fig. 2.3 An example preprocessed Type Ia Supernova light curve from the ZTF simulated dataset (redshift= 0.174). The normalized fluxes of the *r* and *g* passbands are plotted with errors and the solid line is the best fit model of the pre-maximum part of the light curve (up to t_{peak}) using equations 2.3 and 2.4. The horizontal axis is plotted in the rest-frame (redshift corrected), while the vertical axis is the relative de-reddened and distance-corrected flux (or relative luminosity). The vertical black solid line is the date that difference imaging records a trigger alert, and the vertical grey dashed line is the model's prediction of the explosion date with respect to trigger.

where $L_{\text{mod}}(t)$ is the modelled relative luminosity, t_0 is the estimate for time of the initial explosion or collision event, $H(t-t_0)$ is the Heaviside function, and a and c are constants for the amplitude and intercept of the early light curve. The Heaviside function is used to model the t^2 relationship after the explosion, t_0, and fit a constant flux, c, before t_0. We define the pre-maximum part of the light curve as observations occurring up to the simulated peak luminosity of the light curve, t_{peak}. We emphasize that this early model is only required for the training set, and therefore, we are able to use the simulated peak time, which will not be available on observed data and is not used for the testing set.

We make the assumption that the light curves from each passband have the same explosion date, t_0, and fit the light curves from all passbands simultaneously. This is a 5 parameter model: two free parameters, slope (a^λ) and intercept (c^λ), for each of the two passbands, and a shared t_0 parameter. We aim to optimize the model's fit to the light curve by first defining the chi-squared for each transient as:

$$\chi^2(t_0,\mathbf{a},\mathbf{c}) = \sum_{\lambda} \sum_{t=-\infty}^{t_{\text{peak}}} \frac{[L^\lambda_{\text{data}}(t) - L^\lambda_{\text{mod}}(t;t_0,a^\lambda,c^\lambda)]^2}{\sigma^\lambda(t)^2}, \tag{2.4}$$

where λ is the index over passbands, $\sigma(t)$ are the photometric errors in L_{data}, and the sum is taken over all observations at the position of the transient until the time of the peak of the light curve, t_{peak}.

We sampled the posterior probability $\propto \exp\left(-\frac{1}{2}\chi^2\right)$ using MCMC (Markov Chain Monte Carlo) with the affine-invariant ensemble sampler as implemented in the Python package, `emcee` (Foreman-Mackey et al., 2013). We set a flat uniform prior on t_0 to be in a reasonable range before trigger, $-35 < t_0 < 0$, and have a flat improper prior on the other parameters. We use 200 walkers and set the initial positions of each walker as a Gaussian random number with the following mean values: the median of L^λ_{data} for c^λ, the mean of the L^λ_{data} for a^λ, and -12 for t_0. We ran each walker for 700 steps, which after analyzing the convergence of a few MCMC chains, we deemed reasonable to not be too computationally expensive while still finding approximate best fits for $\mathbf{a},\mathbf{c}$ and t_0. The best fit early light curve for an example Type Ia supernova in the training set is illustrated in Fig. 2.3.

We summarize the selection criteria and preprocessing stages applied to the testing and training sets as detailed in Sections 2.3.3 - 2.3.4 in Table 2.1.

Selection Criteria	Applied To
$0 < z \leq 0.5$	Train & Test
$b < 15^\circ$ or $b > 15^\circ$	Train & Test
At least 3 points pre-trigger	Train & Test
Preprocessing	**Applied to**
Sigma-clipping fluxes	Train & Test
Undilate the light-curves by $(1+z)$	Train & Test
Correct light curves for distance.	Train & Test
Correct Milky Way extinction	Train & Test
Rescale light curves between 0 and 1	Train & Test
Model early light curve to obtain t_0	Train only
Keep only $-70 < t < 80$ days from trigger	Train & Test

Table 2.1 Summary of the cuts and preprocessing steps applied to the training and testing sets. The selection criteria help match the simulations to what we expect from the observed ZTF data-stream.

2.3.5 Training Set Preparation

Irregularly sampled time-series data is a common problem in machine learning, and is particularly prevalent in astronomical surveys where the intranight cadence choices and seasonal constraints lead to naturally arising temporal gaps. Therefore, once the light curves have been processed and t_0 has been computed for each transient, we linearly interpolate between the unevenly sampled time series data. From this interpolation, we impute data points such that each light curve is sampled at 3-day intervals between $-70 < t < 80$ days since trigger (or as far as the observations exist), to give a vector of length $n = 50$, where we set the values outside the data range to zero. We ensure that each light curve in a given passband is sampled on the same 3-day grid. The final input image for each transient s is $\boldsymbol{I}^s$, which is a matrix with each row composed of the imputed light curve fluxes for each passband and two additional rows containing repeated values of the host-galaxy redshift in one row and the MW dust reddening in the other row. Hence, the input image is an $n \times (p+2)$ matrix, where p is the number of passbands. This input image, $\boldsymbol{I}^s$, is illustrated as the *Input Matrix* in Fig. 2.1.

One of the key differences in this work compared to previous light curve classification approaches is our ability to provide time-varying classifications. Key to computing this, is labelling the data at each epoch rather than providing a single label to an entire light curve. Using the value of t_0 computed in Section 2.3.4.1, we define two phases of each transient light curve: the pre-explosion phase (where $t < t_0$), and the transient phase (where $t \geq t_0$).

Therefore, the label for each light curve is a vector of length n identifying the transient class at each time-step. This n-length vector is subsequently one-hot encoded, such that each class is changed to a zero-filled vector with one element set to 1 to indicate the transient class (see equation 2.7). This transforms the n-length label vector into an $n \times (m+1)$ vector, where m is the number of transient classes. This is illustrated as the *Class Matrix* in Fig. 2.1.

2.4 Model

2.4.1 Framing the Problem

In this work, we train a deep neural network (DNN) to map the light curve data of an individual transient s onto probabilities over classes $\{c = 1, \ldots, (m+1)\}$. The DNN models a function that maps an input multi-passband light curve matrix, $\boldsymbol{I}^{st}$, for transient s up to a discrete time t, onto an output probability vector,

$$\boldsymbol{y}^{st} = \boldsymbol{f}_t(\boldsymbol{I}^{st}; \boldsymbol{\theta}), \tag{2.5}$$

where $\boldsymbol{\theta}$ are the parameters (e.g. weights and biases of the neurons) of our DNN architecture. We define the input $\boldsymbol{I}^{st}$ as an $n \times (p+2)$ matrix[10] representing the light curve up to a time-step, t. The output $\boldsymbol{y}^{st}$ is a probability vector with length $(m+1)$, where each element y_c^{st} is the model's predicted probability of each class c (at each time step), such that $y_c^{st} \geq 0$ and $\sum_{c=1}^{m+1} y_c^{st} = 1$.

First, to quantify the discrepancy between the model probabilities and the class labels we define a weighted *categorical cross-entropy*,

$$H_w(\boldsymbol{Y}^{st}, \boldsymbol{y}^{st}) = -\sum_{c=1}^{m+1} w_c Y_c^{st} \log(y_c^{st}), \tag{2.6}$$

[10] The reader can consider $\boldsymbol{I}^{st}$ as an image that zeros out all future fluxes after a time t, hence preserving the $n \times (p+2)$ matrix shape irrespective of the image phase coverage. The function $\boldsymbol{f}_t(\cdot; \boldsymbol{\theta})$ only uses the information in the input light curve up to time t.

where w_c is the weight of each class, $\boldsymbol{Y}^{st}$ is the label for the true transient class at each time-step and is a one-hot encoded vector of length $(m+1)$ such that,

$$Y_c^{st} = \begin{cases} 1 & \text{if } c \text{ is the true class of transient } s \text{ at time } t \\ 0 & \text{otherwise} \end{cases} \tag{2.7}$$

where the label, $\boldsymbol{Y}^{st}$, has two phases, the pre-explosion phase with class $c = 1$ when $t < t_0$ and the transient phase with class $c > 1$ when $t \geq t_0$.

If weights were equal for all classes, Eq. 2.6 is proportional to the negative log-likelihood of the probabilities of a categorical distribution (or a generalized Bernoulli distribution). However, to counteract imbalances in the distribution of classes in the dataset which may cause more abundant classes to dominate in the optimization, we define the weight for each class c as

$$w_c = \frac{N \times n}{N_c}, \tag{2.8}$$

where N_c is the number of times a particular class appears in the $N \times n$ training set.

We define the global objective function as

$$\text{obj}(\boldsymbol{\theta}) = \sum_{s=1}^{N} \sum_{t=0}^{n} H_w(\boldsymbol{Y}^{st}, \boldsymbol{y}^{st}), \tag{2.9}$$

where we sum the weighted categorical cross-entropy over all n time-steps and N transients in the training set. To train the DNN and determine optimal values of its parameters $\hat{\boldsymbol{\theta}}$, we minimize this objective function with the sophisticated and commonly used `Adam` gradient descent optimiser (Kingma & Ba, 2015). The model $\boldsymbol{f}_t(\boldsymbol{I}^{st}; \hat{\boldsymbol{\theta}})$ is represented by the complex DNN architecture illustrated in Fig. 2.4 and is described in the following section.

2.4.2 Recurrent Neural Network Architecture

Recurrent Neural Networks (RNNs), such as Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) networks have been shown to achieve state-of-the-art performance in many benchmark time-series and sequential data applications (Bahdanau et al., 2014; Che et al., 2018; Sutskever et al., 2014). Its success in these applications is due to its ability to retain an internal memory of previous data, and hence capture long-term temporal dependencies of variable-length observations in sequential data. We extend this architecture

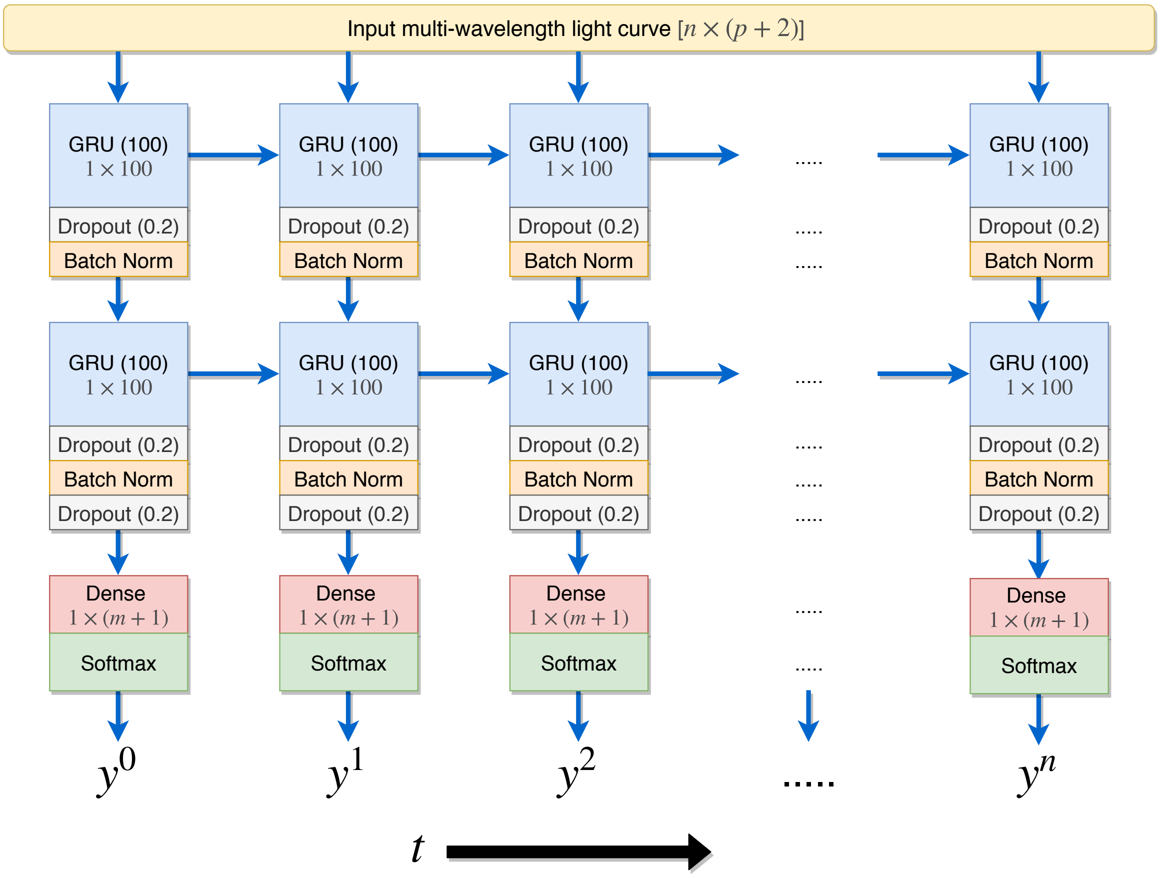


Fig. 2.4 Schematic of the deep recurrent neural network architecture used in RAPID. Each column in the diagram is one of the n time steps of the processed light curve, while each row represents a different neural network layer. The grey text in each block states the shape of the output matrix of each layer in that block. The input image is composed of an $n \times (p+2)$ matrix consisting of the light curve fluxes, host redshift, and Milky Way reddening. Two uni-directional gated recurrent unit layers of size 100 are used for encoding and decoding the input sequences, respectively. It is in these RNN layers that information about previous time-steps is encoded. Batch normalization is applied between each layer to normalize the network parameters and hence, speed the training process. To counter overfitting during training, we employ the dropout optimization technique (Srivastava et al., 2014) to the neurons in each of the GRU and Batch Normalization layers, and set the dropout rate to 20%. Finally, a fully-connected (dense) layer with a softmax regression activation function is applied to compute the probability of each class at each time-step. We wrap the final layer in Keras' *Time Distributed* layer so that each time step is treated independently, and only uses information from the current and previous time-steps.

to our case with a time-varying multi-channel (multiple passbands) input and a time-varying multi-class output.

Recently, Naul et al. (2018), Charnock & Moss (2017), Moss (2018), and Hinners et al. (2018) have used two RNN layers for this framework on astronomical time-series data. However, our work differs from these by making use of *uni-directional* GRUs instead of *bi-directional* RNNs. Bi-directional RNNs are able to pass information both forwards and backwards through the neural network representation of the light curve, and can hence preserve information on both the past and future at any time-step. However, this is only suitable for retrospective classification, because it requires that we wait for the entire light curve to complete before obtaining a classification. The real-time classification used in our work is a novel approach in time-domain astronomy, but necessitates the use of uni-directional RNNs. Hence, our two RNN layers read the light curve chronologically.

The deep neural network (DNN) is illustrated in Fig. 2.4. We have developed the network with the high level `Python` API, `Keras` (Chollet et al., 2015), built on the recent highly efficient `TensorFlow` machine learning system (Abadi et al., 2016). We describe the architecture in detail here.

Input As detailed in Section 2.3.5, the input is an $n \times (p+2)$ matrix. However, as we are implementing a sequence classifier, we can consider the input at each time-step as being vector of length $(p+2)$.

First GRU Layer Gated Recurrent Units are an improved version of a standard RNN and are a variation of the LSTM (see Chung et al. 2014 for a detailed comparison and explanation). We have selected GRUs instead of LSTMs in this work, as they provide appreciably shorter overall training time, without any significant difference in classification performance. Both are able to capture long-term dependencies in time-varying data with parameters that control the information that should be remembered at each step along the light curve. We use the first GRU layer to read the input sequence one time-step at a time and encode it into a higher-dimensional representation. We set-up this GRU layer with 100 units such that the output is a vector of shape 1×100.

Second GRU Layer The second GRU layer is conditioned on the input sequence. It takes the output of the previous GRU and generates an output sequence. Again, we use 100 units in the GRU to maintain the $n \times 100$ output shape. We use uni-directional GRUs that enable only information from previous time-steps to be encoded and passed onto future time-steps.

Batch Normalization We then apply *Batch Normalization* (first introduced in Ioffe & Szegedy 2015) to each GRU layer. This acts to improve and speed up the optimization while adding stability to the neural network and reducing overfitting. While training the DNN, the distribution of each layer's inputs changes as the parameters of the previous layers change. To allow the parameter changes during training to be more stable, batch normalization scales the input. It does this by subtracting the mean of the inputs and then dividing it by the standard deviation.

Dropout We also implement dropout regularization to each layer of the neural network to reduce overfitting during training. This is an important step that effectively ignores randomly selected neurons during training such that their contribution to the network is temporarily removed. This process causes other neurons to more robustly handle the representation required to make predictions for the missing neurons, making the network less sensitive to the specific weights of any individual neuron. We set the dropout rate to 20% of the neurons present in the previous layer each time the Dropout block appears in the DNN in Fig. 2.4.

Dense Layer A dense (or fully-connected) layer is the simplest type of neural network layer. It connects all 100 neurons at each time-step in the previous layer, to the $(m+1)$ neurons in the output layer simply using equation 2.10. As this is a classification task, the output is a vector consisting of all m transient classes and the *Pre-explosion* class. However, as we are interested in time-varying classifications, we wrap this Dense layer with a *Time-Distributed* layer, such that the dense layer is applied independently at each time-step, hence giving an output matrix of shape $n \times (m+1)$.

Neurons The output of each neuron in a neural network layer can be expressed as the weighted sum of the connections to it from the previous layer:

$$\hat{y}_i = f\left(\sum_{j=1}^{M} W_{ij} x_j + b_i\right), \tag{2.10}$$

where x_j are the different inputs to each neuron from the previous layer, W_{ij} are the weights of the corresponding inputs, b_i is a bias that is added to shift the threshold of where inputs become significant, j is an integer running from 1 to the number of connected neurons in the previous layer (M), and i is an integer running from 1 to the number of neurons in the next layer. For the Dense layer, $\mathbf{x}$ is simply the (1×100) matrix from the output of the GRU and Batch Normalisation, $\mathbf{y}$ is made up of the

$(m+1)$ output classes, j runs from 1 to $(m+1)$ and i runs across the 100 input neurons from the GRU. The matrix of weights and biases in the Dense layer and throughout the GRU layers are some of the free parameters that are computed by `TensorFlow` during the training process.

Activation function As with any neural network, each neuron applies an activation function $f(\cdot)$ to bring non-linearity to the network and hence help it to adapt to a variety of data. For feed-forward networks it is common to make use of Rectified Linear Units (ReLU, Nair & Hinton, 2010) to activate neurons. However, the GRU architecture uses sigmoid activation functions as it outputs a value between 0 and 1 and can either let no flow or complete flow of information from previous time-steps.

Softmax regression The final layer applies the softmax regression activation function, which generalises the sigmoid logistic regression to the case where it can handle multiple classes. It applies this to the Dense layer output at each time-step, so that the output vector is normalized to a value between 0 and 1 where the sum of the values of all classes at each time-step sums to 1. This enables the output to be viewed as a relative probability of an input transient being a particular class at each time-step. The output probability vector,

$$\boldsymbol{y} = \mathrm{softmax}(\hat{\boldsymbol{y}}), \tag{2.11}$$

is computed with a softmax activation function that is defined as

$$\mathrm{softmax}(\boldsymbol{x})_i = \frac{e^{x_i}}{\sum\limits_j e^{x_j}}. \tag{2.12}$$

We use the output softmax probabilities to rank the best matching transient classes for each transient light curve at each time-step.

We reiterate that the overall architecture is simply a function that maps an input $n \times (p+2)$ light curve matrix onto an $n \times (m+1)$ softmax probability matrix indicating the probability of each transient class at each time-step. In order to optimize the parameters of this mapping function, we specify a weighted categorical cross-entropy loss-function that indicates how accurately a model with given parameters matches the true class for each input light curve (as defined in equation 2.6).

We minimize the objective function defined in equation 2.9 using the commonly used, but sophisticated stochastic gradient descent optimizer called the `Adam` optimizer (Kingma

& Ba, 2015). As the class distribution is inevitably uneven, and the pre-explosion class is naturally over-represented as it appears in each light curve label, we prevent bias towards over-represented classes by applying class-dependent weights while training as defined in equation 2.8.

The several layers in the DNN create a model that has over one hundred thousand free parameters. As we feed in our training set in batches of 64 light curves at a time, the neural network updates and optimizes these parameters. While the size of the parameter space seems insurmountable, the `Adam` optimizer is able to compute individual adaptive learning rates for different parameters from estimates of the mean and variance of the gradients and has been shown to be extraordinarily effective at optimizing high-dimensional deep learning models.

With the often quoted 'black box' nature of machine learning, it is always a worry that the machine learning algorithms are learning traits that are specific to the training set but do not reflect the physical nature of the classes more generally. Ideally, we would like to ensure that the model we build both accurately captures regularities in the training data while simultaneously generalizing well to unseen data. Simplistic models may fail to capture important patterns in the data, while models that are too complex may overfit random noise and capture spurious patterns that do not generalize outside the training set. While we implement regularization layers (dropout) to try to prevent overfitting, we also monitor the performance of the classifier on the training and testing sets during training. In particular, we ensure that we do not run the classifier over so many iterations that the difference between the values of the objective function evaluated on the training set and the testing set become significant.

2.5 Results

In this section we detail the performance of `RAPID` trained on simulated ZTF light curves. The dataset consists of 48029 transients split between 12 different classes, where each class has approximately 4000 transients. We trained our DNN on 60% of this set and tested its performance on the remaining 40%. The data was preprocessed using the methods outlined in Sections 2.3.3 - 2.3.5. Processing this set, and then training the DNN on it, was computationally expensive, taking several hours to train on the *Nvidia* GPU machines available on `Google Colaboratory`[11]. Once the DNN is trained, however, it is able to classify several thousands of transients within a few seconds.

2.5.1 Hyper-parameter Optimization

One of the key criticisms of deep neural networks is that they have many hyper-parameters describing the architecture that need to be set before training. As training our DNN architecture takes several hours, optimizing the hyper-parameter space by testing the performance of a range of setup values is a very slow process that most similar work have not attempted. Despite this challenge, we performed a broad grid-search of three of our DNN hyper-parameters: number of neurons in each GRU layer, and the dropout fraction. After testing 12 different setup parameters, we found that there was only a 2% variation on the overall accuracy. The hyper-parameters that are shown in Fig. 2.4 were the best performing set of parameters.

2.5.2 Accuracy

We go beyond previous attempts at photometric classification in two important ways. Firstly, we aim to classify a much larger variety of sparse multi-passband transients, and secondly, and most significantly, we provide classifications as a function of time. An example of this is illustrated in Fig. 2.5. At each epoch along the light curve, the trained DNN outputs a softmax probability of each transient class. As more photometric data is provided along the light curve, the DNN updates the class probabilities of the transient based on the state of the network at previous time-steps plus the new time-series data. Within just a few days of the explosion, and well before the ZTF trigger, the DNN was able to correctly learn that the transient evolved from *Pre-explosion* to a SNIa.

To assess the performance of `RAPID`, we make use of several metrics. The most obvious metric is simply the accuracy, that is, the ratio of correctly classified transients in each class to the total number of transients in each class. At each epoch along every light curve in the testing set, we select the highest probability class and compare this to the true class. After aligning each light curve by its trigger, we obtained the prediction accuracy of each class as a function of time since trigger. This is plotted in Fig. 2.6.

The total classification accuracy of each class in the testing set increases quickly before trigger, but then begins to flatten out with only small increases after approximately 20 days

[11] https://colab.research.google.com/

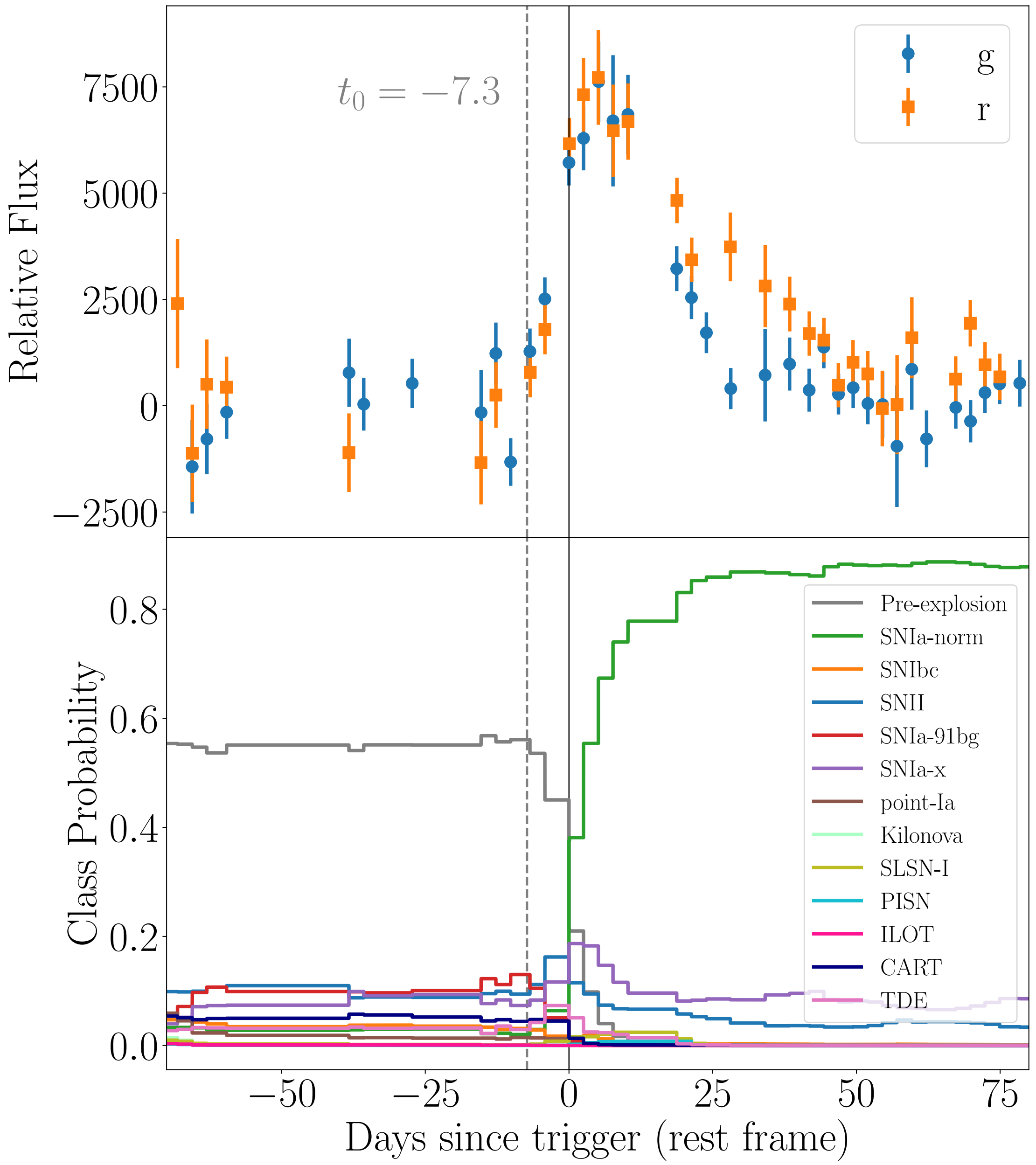


Fig. 2.5 An example normal SNIa light curve from the testing set (redshift= 0.174) is shown in the top panel, and the softmax classification probabilities from `RAPID` are plotted as a function of time over the light curve in the bottom panel. The plot shows the rest frame time since trigger. The vertical grey dashed line is the predicted explosion date from our t^2 model fit of the early light curve (see Section 2.3.4.1). Initially, the object is correctly predicted to be *Pre-explosion*, before it is more confidently predicted as a SNIa-Normal at -20 days before the trigger. Hence, the neural network predicts the explosion date only 4 days after early light curve model fit's prediction. The confidence in the predicted classification improves over the lifetime of the transient.

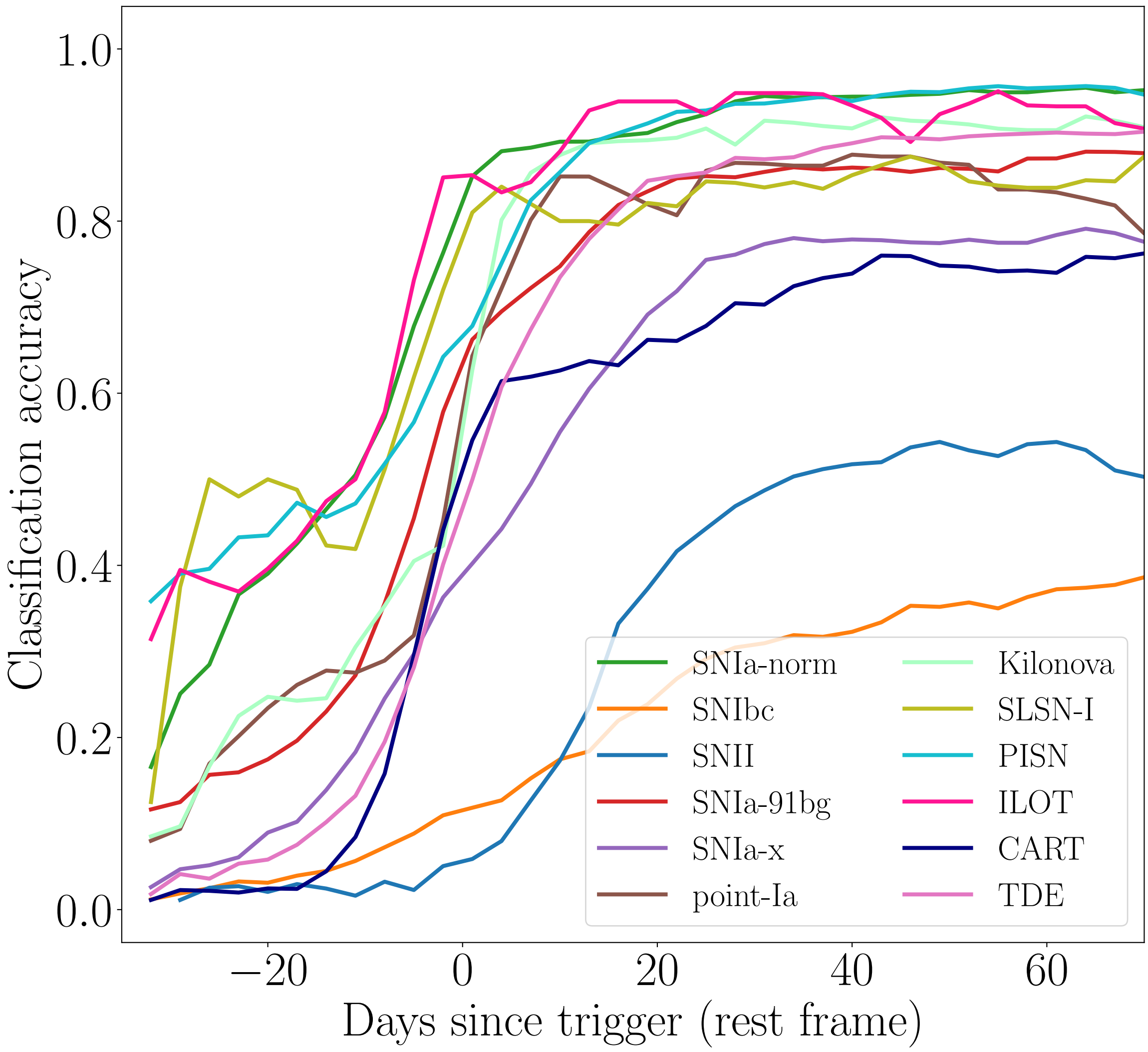
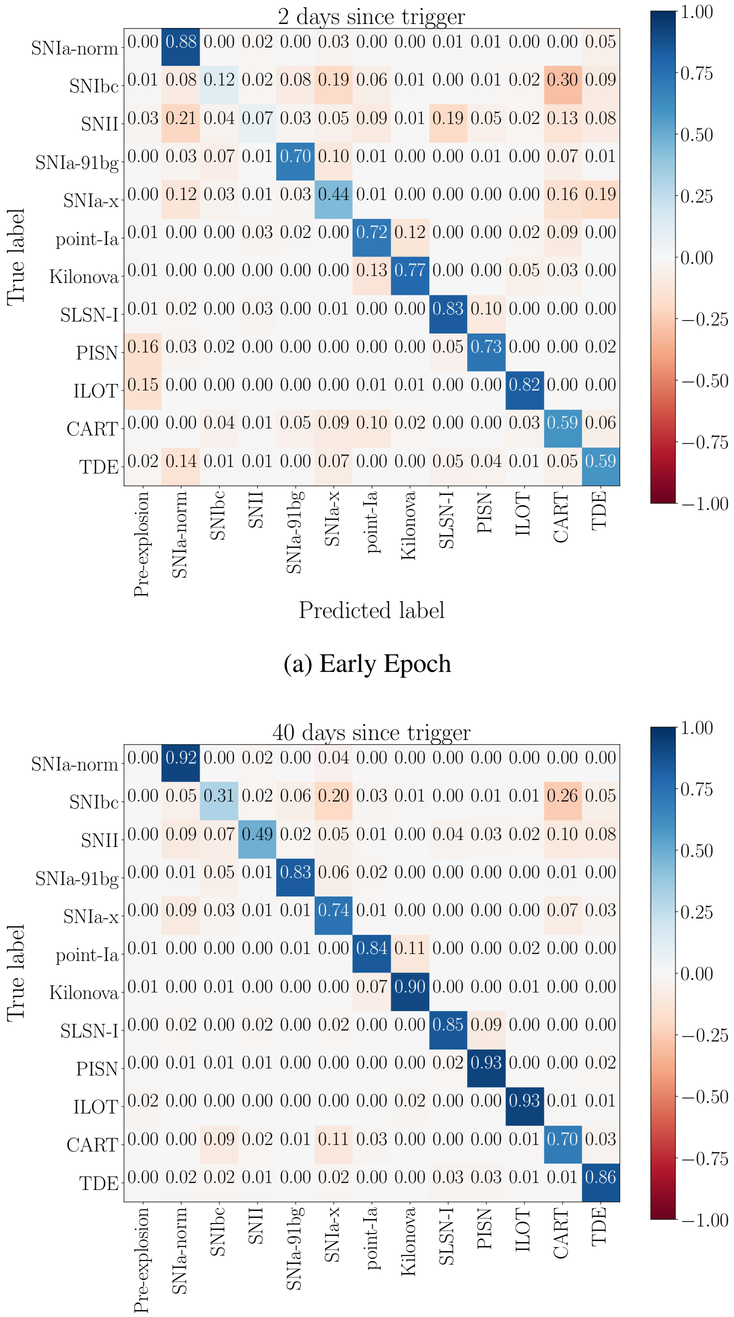


Fig. 2.6 The classification accuracy of each transient class as a function of time since trigger in the rest frame. The values correspond to the diagonals of the confusion matrices at each epoch.



(a) Early Epoch

(b) Late Epoch

Fig. 2.7 The normalized confusion matrices of the 12 transient classes at 2 days past trigger (top), and at 40 days past trigger (bottom). The confusion matrices show the classification performance tested on 40% of the dataset after the classifier was trained on 60% of the dataset. The colour bar and cell values indicate the fraction of each *True Label* that were classified as the *Predicted Label*. Negative colour bar values are used only to indicate misclassifications. Please see the online material (https://www.ast.cam.ac.uk/~djm241/rapid/cf.gif) for an animation of the evolution of the confusion matrix as a function of time since trigger (showing epochs from -25 to 70 days from trigger).

post-trigger. For most classes, the transient behaviour of the light curve is generally nearing completion at this stage, and hence we can expect that new photometric data adds little to improving the classification as the brightness tends towards the background flux level. The classification performance of the core-collapse supernovae, SNIbc and SNII, are particularly poor. To better understand this, it is useful to see where misclassifications occurred.

2.5.3 The Confusion Matrix

The confusion matrix is often a good way to visualize this. Typically, each entry in the matrix describes counts of the number of transients of the true class, c, that had the highest predicted probability in class, $\hat{c}$. For ease of interpretability, we make use of a specially normalized confusion matrix in this work. We normalize the confusion matrix such that the $(c, \hat{c})$ entry is the fraction of transients of the true class c that are classified into the predicted class $\hat{c}$. With this normalization, each row in the matrix must sum to 1. Therefore each row is an estimate of the classifier's conditional distribution of (maximum probability) predicted labels given each true class label.

In Fig. 2.7, we plot the normalized confusion matrices at an early (2 days post-trigger) and late (40 days post-trigger) stage of the light curve. In the online material, we provide an animation of this confusion matrix evolving in time since trigger (instead of just the two epochs shown here)[12].

The overall classification performance is, as expected, slightly better at the late phase of the light curve. However, the performance only 2 days after trigger is particularly promising for our ability to identify transients at early times to gather a well-motivated follow-up candidate list. SNe Ia have the highest classification accuracy at early times with most misclassification occurring with other subtypes of Type Ia supernovae. At late times, the Intermediate Luminosity Transients and TDEs are best identified. The core-collapse supernovae (SNIbc, SNII) appear to be most often confused with calcium-rich transients and other supernova types. CARTs are a newly discovered class of transients and their physical mechanism is not yet well-understood. However, the reason for the confusion most likely stems from their fast rise-times similar to many core-collapse SNe. This is illustrated in Fig. 2.8.

[12] Animations of the plots shown in this chapter can be found here: https://www.ast.cam.ac.uk/~djm241/rapid/

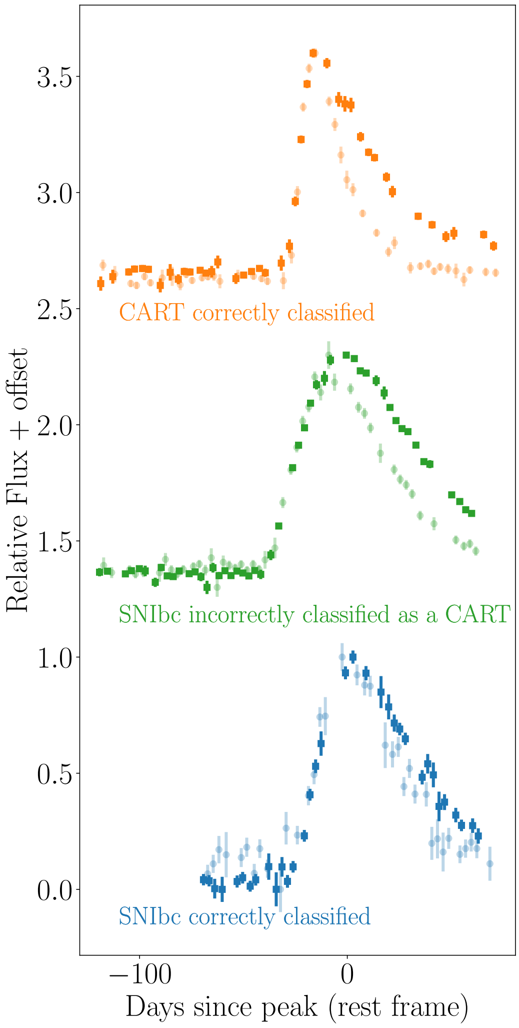


Fig. 2.8 Three of the simulated light curves from our sample - a correctly classified CART (top), a SN Ibc incorrectly classified as a CART (middle), and a correct classified SN Ibc (bottom). The dark-coloured square markers are the *r* band fluxes and the lighter-coloured circle markers are the *g* band fluxes. Our classifier is sensitive to light curve shape, and the limited colour information available with ZTF leads to a significant fraction of SN Ibc objects being misclassified as CARTs. We expect classification performance to improve with LSST, which will provide *ugrizy* light curves.

2.5.4 Receiver Operating Characteristic Curves

The confusion matrix is a good measure of the performance of a classifier, but it does not make use of the full suite of probability vectors we obtain for every transient, and instead only uses the highest scoring class. The Receiver Operating Characteristic (ROC) Curve, on the other hand, makes use of the classification probabilities. Instead of selecting just the highest probability class, we use a probability threshold p_{thresh}. For each class c, transient s at time t is considered to be classified as c if $y_c^{st} > p_{\text{thresh}}$. We sweep the values of p_{thresh} between 0 and 1. The ROC curve plots the true positive rate (TPR) against the false positive rate (FPR) for these different probability thresholds. In a multi-class framework, the TPR is a measure of recall or completeness; it is the ratio between the number of correctly classified objects in a particular class (TP) to the total number of objects in that class (TP + FN).

$$\text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{2.13}$$

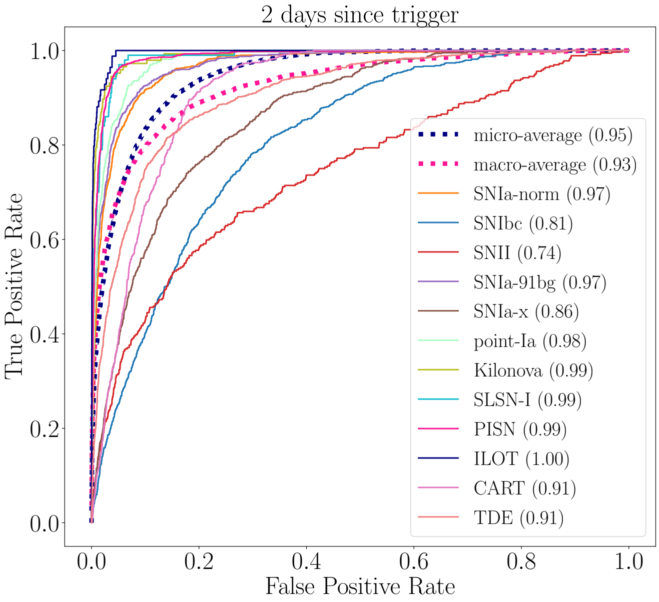
Conversely, the FPR is a measure of the false alarm rate; it is the ratio between the number of transients that have been misclassified as a particular class (FP) and the total number of objects in all other classes (FP + TN).

$$\text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}} \tag{2.14}$$

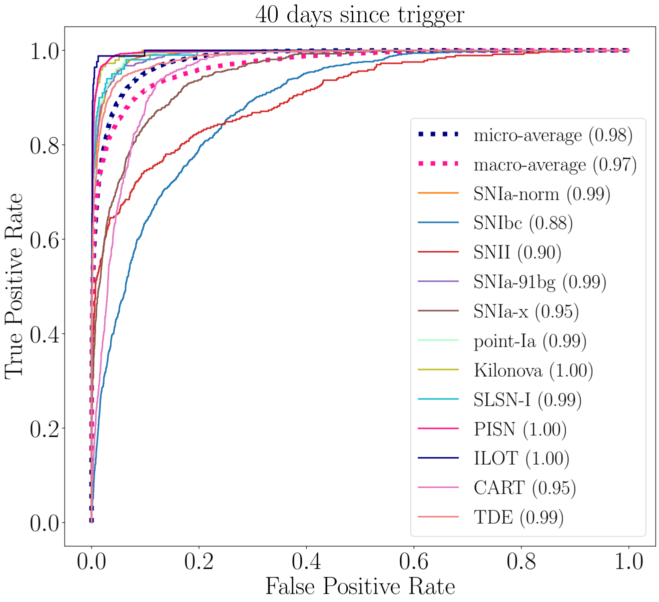
A good classifier is one that maximizes the area under the ROC curve (AUC), with a perfect classifier having an AUC = 1, and a randomly guessing classifier having an AUC = 0.5. Typically, values above 0.9 are considered to be very good classifiers. In Fig. 2.9, we plot the ROC curve at an early and late phase in the light curve. Here, the classification performance looks very good with several classes having AUC values above 0.99 and the overall micro-averaged[13] values being 0.95 for the early stage and 0.98 in the late stage. The macro-averaged ROC is simply the average of all of the ROC curves computed independently. Differently, the micro-averaged ROC aggregates the TPR and FPR contributions of all classes, and is equivalent to the weighted average of all the ROCs considering the number of transients in each class. As the class distribution in the dataset is not too unbalanced, these values are quite close.

In the online material we plot an animation of the ROC curve evolving in time since trigger, rather than the two phases plotted here. As a still-image measure of this, we plot

[13] The micro-averaged ROC curve aggregates the results from all objects from all classes. It is effectively the average of the ROC curves from all classes weighted by the number of objects in each class.



(a) Early Epoch



(b) Late Epoch

Fig. 2.9 Receiver operating characteristic (ROC) curves for the 12 transient classes at an early epoch at 2 days past trigger (left), and at a late epoch at 40 days past trigger (right). Each curve represents a different transient class with the area under the ROC curve (AUC) score in the brackets. The macro-average and micro-average curves which are an average and weighted-average representation of all classes, respectively (see Section 2.5) are also plotted. We compute the metric on the 40% of the dataset used for *testing*. Please see the online material for an animation of the evolution of the ROC curve as a function of time since trigger. (https://www.ast.cam.ac.uk/~djm241/rapid/roc.gif)

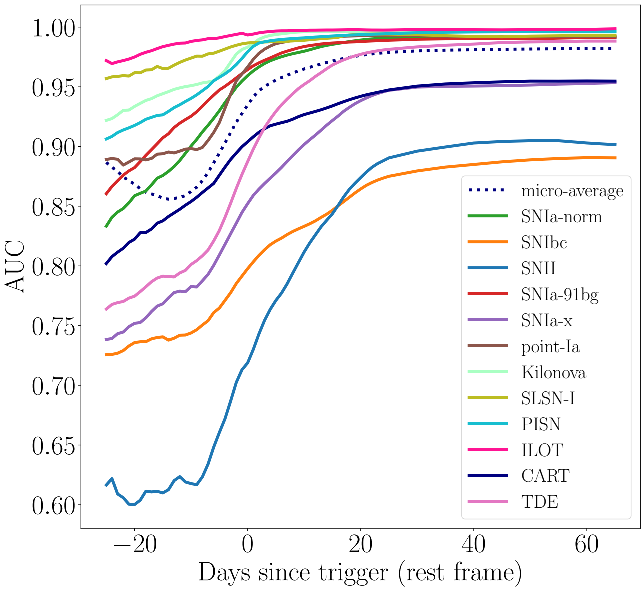
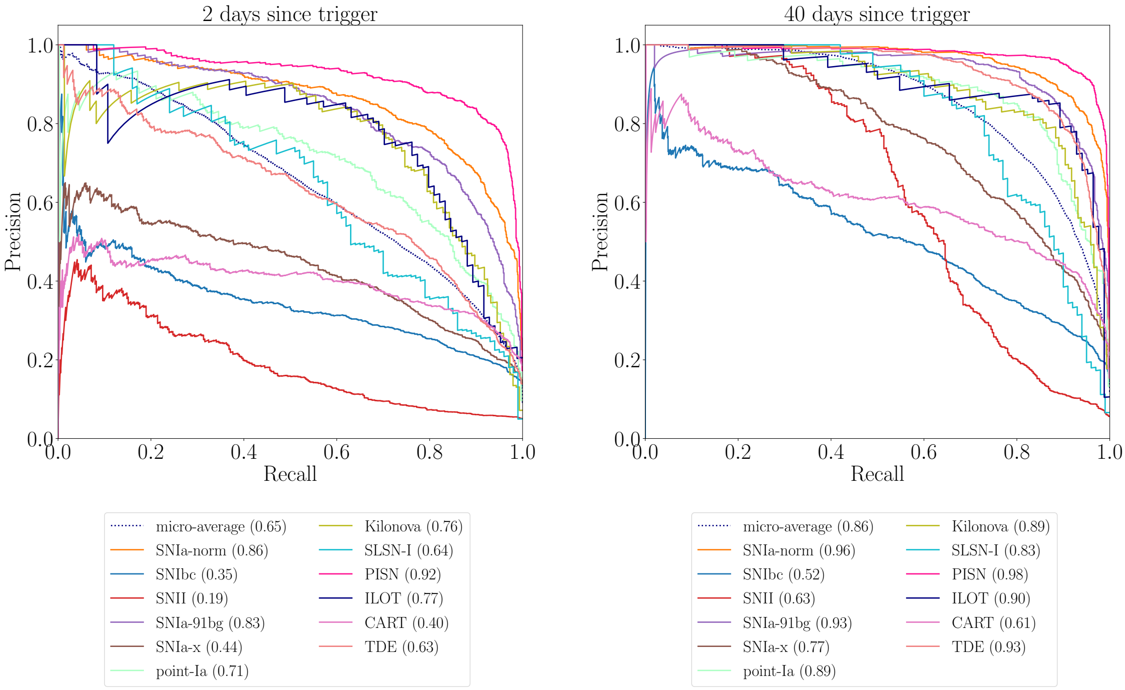


Fig. 2.10 The area under the ROC curve (AUC) of each class as a function of time since trigger. Fig. 2.9 illustrates the ROC curves at two epochs with the AUC of each listed in the legend; this plot shows hows the AUC evolves with time for each class, and is a still-representation of the animation of the ROC curves shown in the online material. The overall performance of the classifier is best judged with the shape of the 'micro-average' curve.



(a) Early Epoch

(b) Late Epoch

Fig. 2.11 Precision-recall curves for the 12 transient classes at an early epoch at 2 days past trigger (left), and at a late epoch at 40 days past trigger (right). We compute the metric on the 40% of the dataset used for *testing*. Please see the online material for an animation of the evolution of the Precision-Recall as a function of time since trigger. (https://www.ast.cam.ac.uk/~djm241/rapid/pr.gif)

the AUC of each class as a function of time since trigger in Fig. 2.10. Within 5 to 20 days after trigger, the AUC flattens out for most classes. We see that the ILOTs, kilonovae, SLSNe, and PISNs are predicted with high accuracy well before trigger. These transients are fainter than most other classes, and hence, do not trigger an alert until their light curves approach maximum brightness. This means, that by the time a trigger happens, the transient behaviour of the light curve is mature, and the classifier has more information to be confident in its prediction. On the other hand, the two core collapse supernovae, SNII and SNIbc, have comparatively low AUCs. We expect that additional colour information will help to separate these from other supernova types. The overall performance is best illustrated by the micro-averaged AUC curve shown as the dotted blue curve. The AUC is initially low due to the misclassifications with Pre-explosion, but within just a few days after trigger, plateaus to a very respectable AUC of 0.98.

2.5.5 Precision-Recall

We compute the Precision-Recall metric. This metric has been shown to be particularly good for classifiers trained on imbalanced datasets (Saito & Rehmsmeier, 2015). The precision (also known as purity) is a measure of the number of correct predictions in each class compared to the total number of predictions of that class, and is defined as,

$$\text{precision} = \frac{\text{TP}}{\text{TP}+\text{FP}}. \tag{2.15}$$

The Recall (also known as completeness) is the same as the true positive rate. It is a measure of the number of correct predictions in each class compared to the total number of that class in the testing set, and is defined as,

$$\text{recall} = \frac{\text{TP}}{\text{TP}+\text{FN}}. \tag{2.16}$$

A good classifier will have both high precision and high recall, and hence the area under the precision-recall curve will be high. In making the precision-recall plot, instead of simply selecting the class with the highest probability for each object, we apply a probability threshold as plotted in Fig. 2.11. By using a very high probability threshold (instead of just selecting the most probable class), we can obtain a much more pure subset of classifications. The PISN, SNIa-norm, SNIa-91bg, kilonovae, pointIa, and ILOTs have very good precision and recall at the late epoch, and quite respectable at the early phase. The core collapse SNe and CARTs are again shown to perform poorly. Overall, this plot highlights some flaws in the classifier that the previous metrics did not capture. In particular, the CART class is shown to perform much more poorly than in previous metrics, highlighting that it does not have a high precision and that there are many false positives for it. As there are fewer CARTs in the test set than other classes, this was not as obvious in the other metrics (see Saito & Rehmsmeier (2015) for an analysis of precision-recall vs ROC curves as classification metrics).

2.5.6 Weighted Log Loss

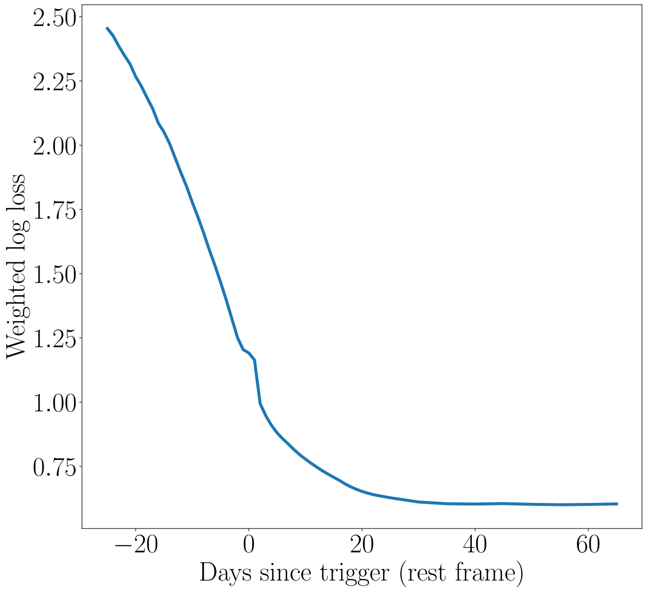


Fig. 2.12 The weighted log loss defined in equation 2.17 and used in PLAsTiCC (Malz et al., 2018) is plotted as a function of time since trigger. The weighted log-loss of the early (2 days after trigger) and late (40 days after trigger) epochs are 1.09 and 0.64, respectively.

In each of the previous metrics we have treated each class equally. However, it is often useful to weight the classifications of particularly classes more favourably than others. Malz et al. (2018) recently explored the sensitivity of a range of different metrics of classification probabilities under various weighting schemes. They concluded that a weighted log-loss provided the most meaningful interpretation, defined as follows

$$\ln \mathrm{Loss}^t = -\left(\frac{\sum_{c=1}^{(m+1)} w_c \cdot \sum_{s=1}^{N_c} \frac{Y_c^{st}}{N_c} \cdot \ln y_c^{st}}{\sum_{c=1}^{(m+1)} w_c} \right) \tag{2.17}$$

where c is an index over the $(m+1)$ classes and j is an index over all N members of each class, y_c^{st} is the predicted probability that object s at time t is a member of class c, and $\boldsymbol{Y}^{st}$ is the truth label. The weight of each class w_c can be different, and N_c is the number of objects in each class.

This metric is currently being used in the PLAsTiCC Kaggle challenge to assess the classification performance of each entry. We apply a weight of 2 to the classes that PLAsTiCC deemed to be rare or interesting and 1 to the remaining classes:

Weight 1 SNIa, SNIbc, SNII, Ia-91bg, Ia-x, Pre-explosion

Weight 2 Kilonova, SLSN, PISN, ILOT, CART, TDE

We plot the weighted log-loss as a function of time since trigger in Fig. 2.12. The metric of the early (2 days after trigger) and late (40 days after trigger) epochs are 1.09 and 0.64, respectively, where a perfect classifier receives a score of 0. While we have applied our classifier to ZTF simulations, we find that the raw scores are competitive with top scores in the PLAsTiCC Kaggle challenge. The sharp improvement in performance at trigger is primarily due to the prior placed on the Pre-explosion phase of the light curve that forces it to be before trigger. Within approximately 20 days after trigger, the classification performance plateaus, as the transient phase of most light curves is ending.

2.6 Application to Observational Data

One of the primary challenges with developing classifiers for astronomical surveys is obtaining a labelled sample of well-observed transients across a wide range of classes. While it may be possible to obtain a labelled sample of common supernovae during the initial stages of a survey, the observation rates of less common transients (such as kilonovae and CARTs, for example) mean that a diverse and large training set of observed data is impossible to obtain. Therefore, a classifier that is trained on simulated data but can classify observational data streams is of significant importance to the astronomical community. To this end, a key goal of `RAPID` is to be able to classify observed data using an architecture trained on only simulated light curves. In this section, we provide a few examples of `RAPID`'s performance on transients from the ongoing ZTF data stream. In Chapter 4, we extend this analysis to test the classification performance on a much larger set of observed light curves.

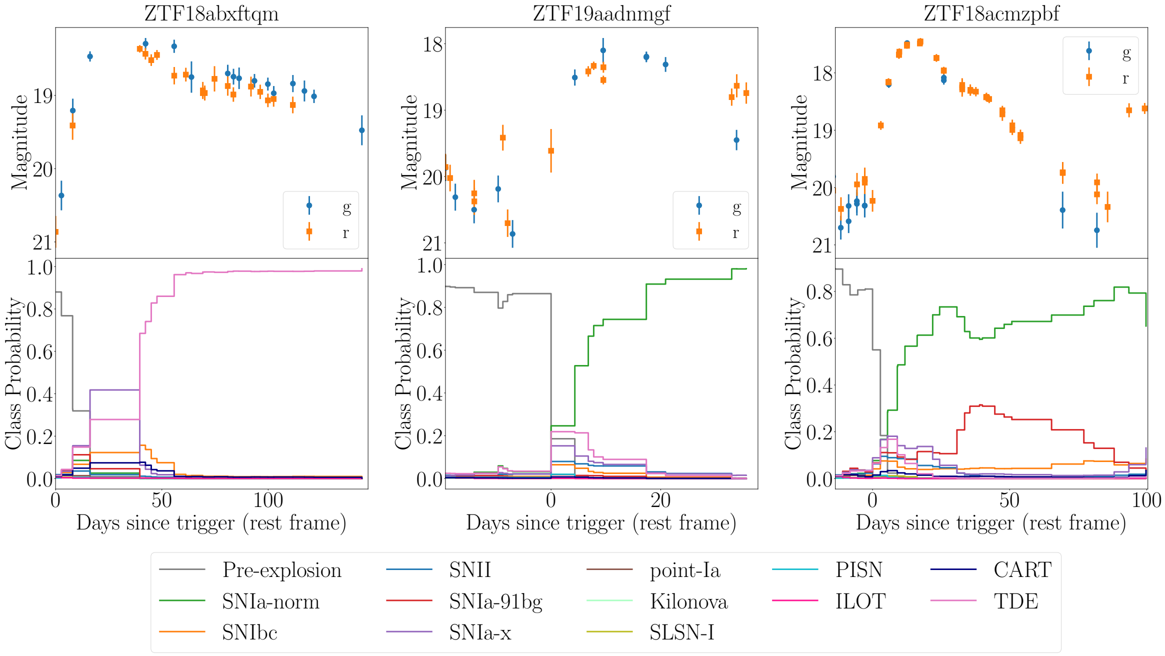


Fig. 2.13 Classification of three light curves from the observed ZTF data stream. In each of the three example cases, `RAPID` correctly classifies the transient well before peak brightness, and often within just a few epochs. The ZTF names are listed in the titles of each transient plot and from left to right they are also known by the following names: AT2018hco, SN2019bly, SN2018itl. These objects were spectroscopically classified as a TDE ($z = 0.09$), SNIa ($z = 0.08$), and SNIa ($z = 0.036$), respectively (Fremling et al., 2018, 2019; van Velzen et al., 2018).

.

In Fig. 2.13 we have tested RAPID on three objects recently observed by ZTF to highlight its direct use on observational data: ZTF18abxftqm, ZTF19aadnmgf, and ZTF18acmzpbf (also known as AT2018hco[14], SN2019bly[15], SN2018itl[16], respectively). These have already been spectroscopically classified as a TDE ($z = 0.09$), SNIa ($z = 0.08$), and SNIa ($z = 0.036$), respectively (Fremling et al., 2018, 2019; van Velzen et al., 2018). In the bottom panel of Fig. 2.13, we see that RAPID was able to correctly confirm the class of each transient well before maximum brightness, and within just a couple of epochs after trigger.

The two SNIa light curves were correctly identified after just one detection epoch, and the confidence in these classifications improved over the lifetime of the transient. While the SNIa-norm probability was lower for ZTF18acmzpbf, this is a good example of where the confidence in this transient being any subtype of SNIa was actually much higher. Given that the second most probable class was a SNIa-91bg, we can sum the two class probabilities to obtain a much higher probability of the transient being a SNIa.

While we have shown RAPID's effective performance on some observational data, future revisions of the software can be used to identify differences between the simulated training set and observations. This will help to improve the transient class models that were used to generate the light curve simulations. Future iterations to improve the simulations will in turn lead to a classifier that is even more effective at classifying observational data (see Chapter 4).

Moreover, as it stands, RAPID can classify 12 different transient classes. However, if an unforeseen transient were passed into the classifier, the class probabilities would split between the classes that were most similar to the input. RAPID is a supervised learning algorithm, and is not designed for anomaly detection. However, cases where RAPID is not confident on a classification may warrant closer attention for the possibility of an unusual transient. In Chapter 3, we deal specifically with the issue of anomaly detection for such cases where classification is not suitable.

2.6.1 Balanced or representative datasets

Machine learning based classifiers such as neural networks often fail to cope with imbalanced training sets as they are sensitive to the proportions of the different classes (Kotsiantis

[14] https://wis-tns.weizmann.ac.il/object/2018hco
[15] https://wis-tns.weizmann.ac.il/object/2019bly
[16] https://wis-tns.weizmann.ac.il/object/2018itl

et al., 2006). As a consequence, these algorithms tend to favour the class with the largest proportion of observations. This is particularly problematic when trying to classify rare classes. The intrinsic rate of some majority classes, such as SNe Ia, is orders of magnitude higher than some rare classes, such as kilonovae. A neural network classifier trained on such a representative dataset, and that aims to minimise the overall unweighted objective function (equation 2.9), will be incentivised to learn how to identify the most common classes rather than the rare ones. An example of the poorer performance of classifiers trained on representative transient datasets compared to balanced datasets is illustrated well by Figure 7 in Narayan et al. (2018). The shown t-SNE (t-distributed Stochastic Neighbour Embedding, van der Maaten & Hinton 2008) plot is able to correctly cluster classes much more accurately when the dataset is balanced.

Moreover, building a representative dataset is a very difficult task and there is a non-representativeness between spectroscopic and photometric samples. The detection rates of different transient classes in photometric surveys are often biased by brightness and the ease at which some classes can be classified over others. Spectroscopic follow-up strategies have been dominated by SNe Ia for cosmology, and have hence led to biased spectroscopic sets. Recently, however, Ishida et al. (2019b) identified a framework for constructing a more representative training set. They make use of real-time active learning to improve the way labelled samples are obtained from spectroscopic surveys to ultimately optimise the scientific performance of photometric classifiers. Employing such a framework will allow for the construction of a training set that is more representative.

In our work, we simulate a balanced training set in an attempt to mitigate the effects of the bias present in existing datasets and to improve our classifier's accuracy on rare classes. While machine learning classifiers tend to perform better on balanced datasets (Kotsiantis et al., 2006; Narayan et al., 2018), future work should verify this for photometric identification by comparing the performance of classifiers trained on balanced and representative training sets. However, until the issue of the non-representativeness between spectroscopic and photometric samples is mitigated with approaches like Ishida et al. (2019b), a representative dataset remains difficult to build.

2.7 Feature-based Early Classification

To compare the performance of RAPID against traditional light curve classification approaches which often use extracted light curve features for classification, we developed a Random

Forest-based classifier that computed statistical features of each light curve to use as input, rather than directly using the photometric information. This has been the most commonly used approach in light curve classification tasks to date (e.g. Karpenka et al., 2013; Lochner et al., 2016; Möller et al., 2016; Narayan et al., 2018; Newling et al., 2011; Revsbech et al., 2018). We extend upon the approach developed in Narayan et al. (2018) and based on Lochner et al. (2016) by using a wider variety of important features and by extending the problem for early light curve classification. Specifically, we only compute features using data up to 2 days after trigger so that the Random Forest classifier can be directly compared with the DNN early classifications.

Extracting features from light curves provides us with a uniform method of comparing between transients which are often unevenly sampled in time. We can train directly on a synthesised feature vector instead of the photometric light curves. For time-series data, extracting moments is the most obvious way to start obtaining features. We compute several moments of the light curve, and a list of the distilled features used in classification are listed in Table 2.2. While we focus on early classification in this chapter, we also list some full-light curve features that we used in work not shown in this chapter that some readers may find useful. As we have two different passbands, we compute the features for each passband and obtain twice the number of moment-based features listed in the table. We also make use of more context specific features, such as redshift and colour information.

We compute the *early rise rate* feature for each passband as the slope of the fitted early light curve model defined in Section 2.3.4.1,

$$\mathrm{rate}^{\lambda} = \frac{L^{\lambda}_{\mathrm{mod}}(t_{\mathrm{peak}}) - L^{\lambda}_{\mathrm{mod}}(t_0)}{(t_{\mathrm{peak}} - t_0)}. \tag{2.18}$$

We use the rise rate, and the early light curve model parameter fits $\hat{\boldsymbol{a}}$ and $\hat{\boldsymbol{c}}$ from equation 2.3 as features in the early classifier. We then define the colour as a function of time,

$$\mathrm{colour}(t) = -2.5\log_{10}\left(\frac{L^{g}_{\mathrm{mod}}(t)}{L^{r}_{\mathrm{mod}}(t)}\right), \tag{2.19}$$

where $L^{\lambda}_{\mathrm{mod}}(t)$ is the modelled relative luminosity (defined in equation 2.3) at a particular passband, λ.

We use the colour curves computed from each transient to define several features. Using equation 2.19, we compute the colour of each object at a couple of well-spaced points on the early light curve (5 days and 9 days after t_0) and use them as features in our early classifier.

Early light curve features only	
Early rise rate	Slope of early light curve (see equation 2.18).
a	Amplitude of quadratic fit to the early light curve (see equation 2.3).
c	Intercept of quadratic fit to the early light curve (see equation 2.3).
Colour at n days	Logarithmic ratio of the flux in two passbands (see equation 2.19).
Early colour slope	Slope of the colour curve.
Early and full light curve features	
Redshift	Photometric cosmological redshift.
Milky Way Dust Extinction	Interstellar extinction.
Historic Colour	Logarithmic ratio of the flux in two passbands before trigger. 2.19).
Variance	Statistical variance of the flux distribution.
Amplitude	Ratio of the 99th minus 1st and 50th minus 1st percentile of the flux distribution.
Standard Deviation / Mean	A measure of the average inverse signal-to-noise ratio.
Median Absolute Deviation	A robust estimator of the standard deviation of the distribution.
Autocorrelation Integral	The integral of the correlation vs time difference (Mislis et al., 2016).
Von-Neumann Ratio	A meausure of the autocorrelation of the flux distribution.
Entropy	The Shannon entropy assuming a Gaussian CDF following Mislis et al. (2016).
Rise time	Time from trigger to peak flux.
Full light curve features only	
Kurtosis	Characteristic “peakedness” of the magnitude distribution.
Shapiro-Wilk Statistic	A measure of the flux distribution’s normality.
Skewness	Characteristic asymmetry of the flux distribution.
Interquartile Range	The difference between the 75th and 25th percentile of the flux distribution.
Stetson K	An uncertainty weighted estimate of the kurtosis following Stetson (1996).
Stetson J	An uncertainty weighted estimate of the Welch-Stetson Variability Index (Welch & Stetson, 1993).
Stetson L	Product of the Stetson J and Stetson K moments (Kinemuchi et al., 2006; Stetson, 1996).
HL Ratio	The ratio of the amplitudes of points higher and lower than the mean.
Fraction of observations above trigger	Fraction of light curve observations above the trigger.
Period	Top ranked periods from the Lomb-scargle periodogram fit of the light curves (Lomb, 1976; Scargle, 1982).
Period Score	Period weights from the Lomb-scargle periodogram fit of the light curves (Lomb, 1976; Scargle, 1982).
Colour Amplitude	Ratio of the amplitudes in two passbands.
Colour Mean	Ratio of the mean fluxes in two passbands.

Table 2.2 Description of the features extracted from each passband of each light curve in the dataset. Some of these are redefined from Table 2 of Narayan et al. (2018).

We also compute the slope of the colour curve and use that as an additional feature for the early classifier. For the full light curve classifier, we compute the colour amplitude as the difference in the light curve amplitudes in two different passbands, and also compute the colour mean as the ratio of the mean flux value of two different passbands.

We feed the feature set into a Random Forest classifier. Random Forests (Breiman, 2001) are one of the most flexible and popular machine learning architectures. They construct an ensemble of several fully grown and uncorrelated decision trees (Morgan & Sonquist, 1963) to create a more robust classifier that limits overfitting. Each decision tree is made up of a series of hierarchical *branches* that check whether values in the feature vector are in a particular range until it ascertains each of the class labels in the form of *leaves*. The trees are trained recursively and independently, selecting which feature and boundary provide the highest information gain for classifications. A single tree is subject to high variance and can easily overfit the training set. By combining an ensemble of decision trees - providing each tree with a subset of data that is randomly replaced during training - a Random Forest is able to decrease the variance by averaging the results from each tree.

We have designed the Random Forest with 200 estimators (or trees) and have run it through twice. On the first run we feed the classifier the entire feature-set. We then rank the features by importance in classification and select the top 30 features. We feed only these top 30 features into the second run of the classifier. As many of the features are obviously highly correlated with each other, this acts to reduce *feature dilution*, whereby we remove features that do not provide high selective power.

We compute features using light curves up to only the first 2 days after trigger. As the Random Forest is much quicker to classify than the DNN, we perform 10-fold cross-validation to obtain a more robust estimate of the classifier's performance. We then produce the confusion matrix in Fig. 2.14 and the ROC curve in Fig. 2.15. We can compare these metrics to the early epoch metrics at 2 days after trigger produced with the deep neural network in Figures 2.7 and 2.9. We find that the performance in the early part of the light curve is marginally worse than the DNN with a micro-averaged AUC of 0.92 compared to 0.95. Moreover, the ability of the DNN to provide time-varying classifications makes it much more suited to early classification than the Random Forest.

In Fig. 2.16, we rank the importance of the top 30 features in the Random Forest classifier. While redshift is clearly the most important feature in the dataset, we have also built classifiers without using redshift as a feature and found that the performance was only marginally worse. This provides an insight into the classifier's robustness when applied to surveys where

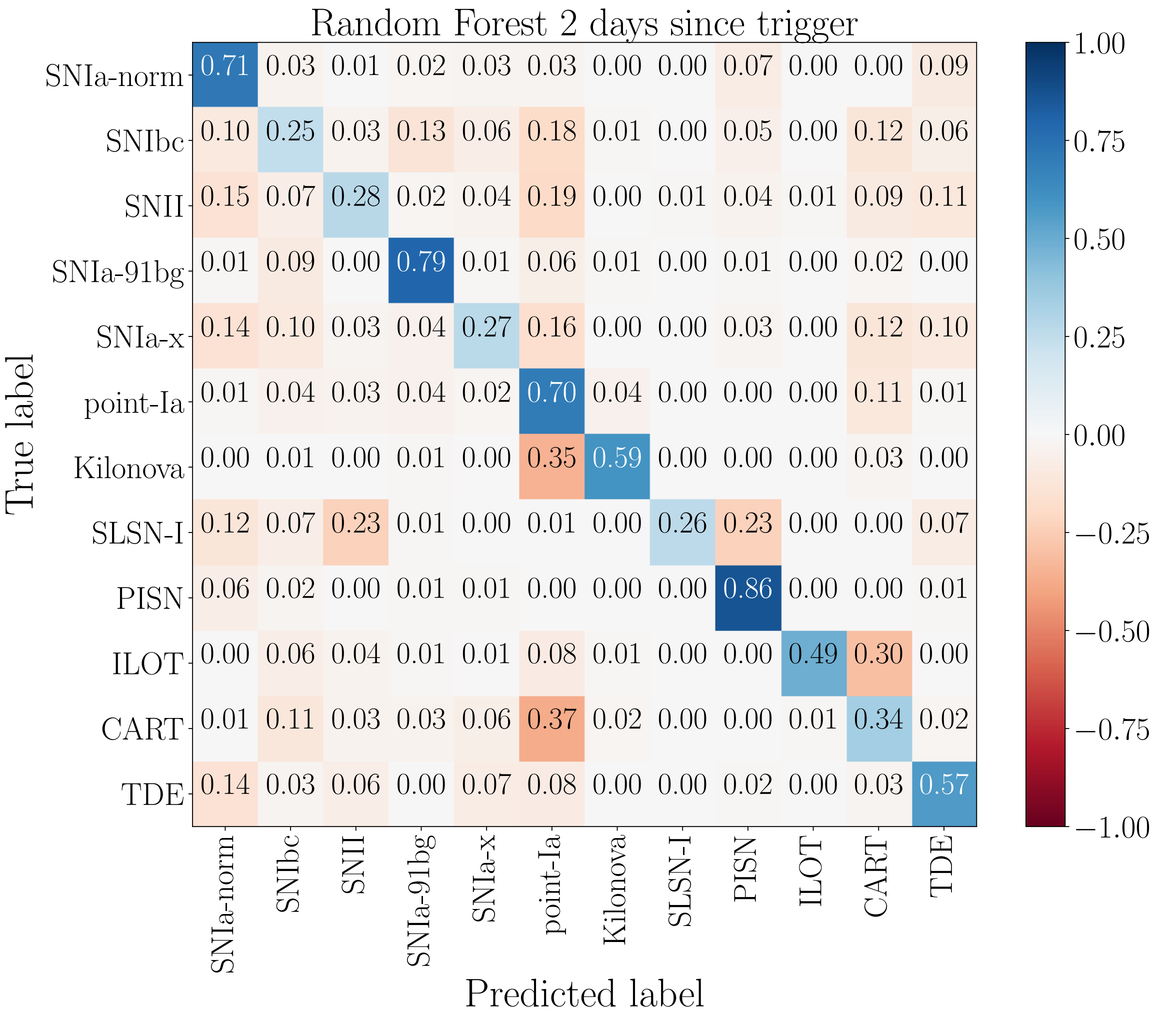


Fig. 2.14 Confusion matrix of the early light curve Random Forest classifier trained on 60% and tested on 40% of the dataset described in Section 2.3. The classifier makes use of 200 estimators (trees) in the ensemble. The colour bar and values indicate the percentage of each *true label* that were classified as the *predicted label*. Negative colour bar values are used only to indicate misclassifications.

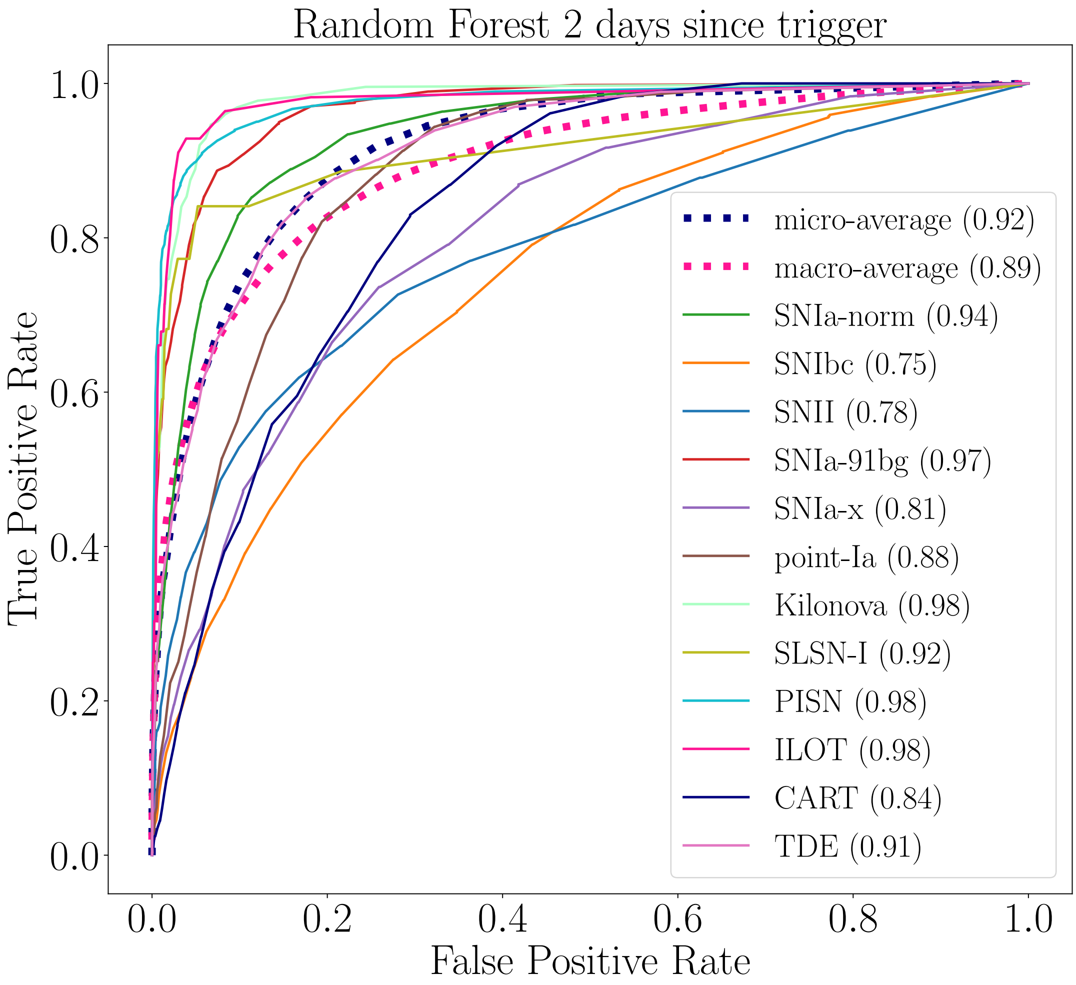


Fig. 2.15 Receiver operating characteristic of the feature-based Random Forest approach. The features are computed on photometric data up to 2 days past trigger, and are fed into a Random Forest classifier. Each curve represents a different transient class with the area under the curve (AUC) score in brackets. The macro and micro average curves which are an average and weighted-average representation of all classes are also plotted. The metric computed on 40% of the dataset.

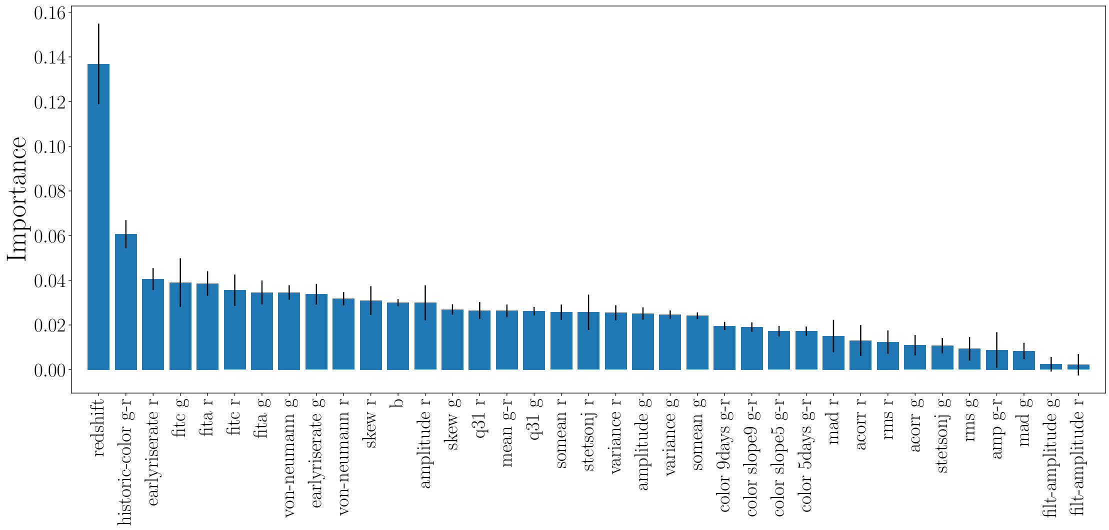


Fig. 2.16 Features used in the early classifier ranked by importance. The bar height is the average relative importance of each feature in each tree of the random forest. The error bar is the standard deviation of the importance of each feature in the 200 trees of the Random Forest. Only the top 30 features have been plotted in the histogram.

redshift is not available. The next best feature is the historic colour, suggesting that the type of host-galaxy is important contextual information to discern transients. The early slope of the light curve also ranks highly, as it is able to distinguish between faster rising core-collapse supernovae and other slower rising transients such as magnetars (SLSNe).

2.8 Conclusions

Existing and future wide-field optical surveys will probe new regimes in the time-domain, and find new astrophysical classes, while enabling a deeper understanding of presently rare classes. In addition, correlating these sources with alerts from gravitational wave, high-energy particle, and neutrino observatories will enable new breakthroughs in multi-messenger astrophysics. However, the alert-rate from these surveys far outstrips the follow-up capacity of the entire astronomical community combined. Realising the promise of these wide-field surveys requires that we characterize sources from sparse early-time data, in order to select the most interesting objects for more detailed analysis.

We have detailed the development of a new real-time photometric classifier, `RAPID`, that is well-suited for the millions of alerts per night that ongoing and upcoming wide-field surveys such as ZTF and LSST will produce. The key advantages that distinguish our approach from others in the literature are:

1. Our deep recurrent neural network with uni-directional gated recurrent units, allows us to classify transients using the available data as a function of time.
2. Our architecture combined with a diverse training set allows us to identify 12 different transient classes *within days of its explosion, despite low S/N data and limited colour information*.
3. We do not require user-defined feature extraction before classification, and instead use the processed light curves as direct inputs.
4. Our algorithm is designed from the outset with speed as a consideration, and it can classify the tens of thousands of events that will be discovered in each LSST image within a few seconds.

This critical component of `RAPID` that enables early classification is our ability to use measurements of the source before an alert is triggered — “precovery” photometry with insufficient significance to trigger an alert, but that nevertheless encodes information about the transient. While we designed `RAPID` primarily for early classification, the flexibility of our architecture means that it is also useful for photometric classification with any available phase coverage of the light curves. It is competitive with contemporary approaches such as Charnock & Moss (2017); Lochner et al. (2016); Narayan et al. (2018) when classifying the full light curve.

There is no satisfactory single metric that can completely summarise classifier performance, and we have presented detailed confusion matrices, ROC curves and measures of precision vs recall for all the classes represented in our training set. The micro-averaged AUC, the most common single metric used to measure classifier performance, evaluated across the 12 transient classes is 0.95 and 0.98 at 2 days and 40 days after an alert trigger, respectively. We further evaluated `RAPID`’s performance on a few transients from the real-time ZTF data stream, and, as an example, have shown its ability to effectively identify a TDE and two SNe Ia well before maximum brightness. The results at early-times are particularly significant as, in many cases, they can exceed the performance of trained astronomers attempting visual classification.

We also developed a second early classification approach that trained a Random Forest classifier on features extracted from the light curve. This allowed us to directly compare the feature-based Random Forest approach to RAPID's model-independent approach. We found that the classification performances are comparable, but the RNN has the advantage of obtaining time-varying classifications, making it ideal for transient alert brokers. To this end, we have recently begun integrating the RAPID software with the ANTARES alert-broker, and plan to apply our DNN to the real-time ZTF data stream in the near future.

Overall, RAPID provides a novel and effective method of classifying transients and providing prioritised follow-up candidates for the new era of large scale transient surveys.

Chapter 3

Transient Anomaly Detection

3.1 Overview

New large-scale transient surveys, such as the Rubin Observatory Legacy Survey of Space and Time (LSST), will observe millions of transient alerts each night, making standard approaches of visually identifying new and interesting transients infeasible. In this chapter, we present two novel methods of automatically detecting anomalous transient light curves in real-time. The first approach is a probabilistic deep neural network built using state-of-the-art Temporal Convolutional Networks (TCNs) aimed at predicting future observations in a light curve. The second approach is based on an interpretable Bayesian parametric model of a light curve, where we predict future observations by fitting past partial light curve data. We provide anomaly scores as a function of time by using a metric that compares our predictions with photometric observations. We build autoregressive models of six transient classes. We have demonstrated the performance of our methods on light curve simulations matching the observing properties of the Zwicky Transient Facility (ZTF) and on real ZTF light curves from the public Mid Scale Innovation Program (MSIP) survey. In particular, we identify anomalies with respect to common supernova classes with low false anomaly rates and high true anomaly rates achieving Area Under the Receive Operating Characteristic (ROC) Curve (AUC) scores well above 0.8 for most rare classes such as kilonovae, tidal disruption events, intermediate luminosity transients, and pair-instability supernovae. Our ability to identify anomalies improves over the lifetime of the light curves. Our frameworks used in conjunction with transient classifiers will enable fast and prioritised follow-up of unusual transients from ongoing and upcoming wide-field surveys.

3.2 Introduction

Astronomy is reaching an unprecedented era of big data, where astronomers are observing more transient events than they can possibly visually examine. Upcoming large-scale surveys of the transient universe such as the Vera C. Rubin Observatory Legacy Survey of Space and Time (LSST) will observe two orders of magnitude more transient events than any survey to date, (Ivezić et al., 2019). LSST is expected to observe over 10 million transient alerts each night, making it infeasible to visually examine or follow up any significant fraction of transient candidates. However, for a long time, discovery in astronomy has been driven by serendipity and by identifying anomalies in data sets. To this end, identifying anomalous objects and prioritising which of the millions of alerts are most suitable for spectroscopic followup is a challenge that needs to be automated. In this chapter, we develop a novel framework for identifying anomalous transients in real-time.

There have been several efforts to automate the identification of astronomical transients in large-scale surveys (e.g. Lochner et al. 2016; Narayan et al. 2018; Pasquet et al. 2019). These efforts are useful for dealing with the big datasets in recent surveys, but they require the full phase coverage of each light curve for classification. While retrospective classification after the full light curve of an event has been observed is useful, it also limits the scientific questions that can be answered about these events, many of which exhibit interesting physics at early times. To prioritise followup, the type of transient and its phase of evolution are most important.

Obtaining detailed followup observations shortly after a transient's explosion provides insights into the progenitor systems that power the event and hence improves our understanding of the object's physical mechanism. While the mechanism of some transients are reasonably well-understood, the central engine of various exotic classes such as calcium-rich gap transients, super-luminous supernovae, and some newly discovered fast blue optical transients (FBOTs) are poorly understood (e.g. Coppejans et al., 2020). Moreover, even though Type Ia Supernovae (SNe Ia) have been well studied, their progenitor system remains mysterious (e.g. Livio & Mazzali, 2018; Ruiter, 2020, for reviews of the current state of SNIa progenitor origin). Furthermore, the discovery of the electromagnetic counterpart from the binary neutron star merger gravitational wave event, GW170817 (Abbott et al., 2017a), and the considerably human effort that went into the followup, has made it clear that automated photometric classifiers are necessary. The need for rapid identification of these events means automatically sifting through the millions of transient alerts produced each night and identifying candidates at early times.

To this end, recent methods such as SuperNNova (Möller & de Boissière, 2020) and RAPID (Muthukrishna et al., 2019a) have developed early and real-time classifiers capable of identifying the specific type of transient shortly after explosion. These approaches use state-of-the-art deep recurrent neural networks (RNNs) to model a function that maps real-time photometric information onto a range of different transient classes, and are able to update their prediction as new photometric data along a transient's light curve become available. They enable astronomers to prioritise candidates for follow-up observations.

However, one major caveat of all existing approaches, is that classification is inherently a supervised learning task, and hence, requires either comprehensive labelled data for training an algorithm, or well-understood models that enable simulating a training set. They are unable to classify events that they have not been specifically trained on. But, with the deluge of data coming from upcoming wide-field surveys, that are probing deeper, wider, and faster than ever before, we should be prepared for an influx of completely new classes of objects. LSST will have a point source depth of r$\sim$27.5 (LSST Science Collaboration et al., 2009), and will be able to probe fainter than any other wide-scale survey to date, while the Transiting Exoplanet Science Survey (TESS, Ricker et al. 2015) will use its wide field-of-view to explore transient phenomena at the minutes to hours timescale which is a region of parameter space that has been relatively unexplored. Consequently, astronomers are in need of methods capable of discovering new and unknown transient phenomena within the context of the huge datasets in modern time-domain astronomy.

Anomaly detection is a data-driven approach to finding such outliers. The goal is to detect outliers that are scientifically interesting, rather than random statistical fluctuations. Within astronomy, anomaly detection algorithms have been used in a range of applications, and recently Lochner & Bassett (2020) has developed an active learning framework called Astronomaly to make the identification of anomalies in a range of datasets systematic and easily accessible.

However, applying anomaly detection to time-series such as astronomical light curves is a more challenging problem than identifying anomalies in static datasets such as images or spectra. Recently, there have been a few anomaly detection algorithms applied to astronomical light curves (e.g. Giles & Walkowicz, 2019; Ishida et al., 2019a; Lochner & Bassett, 2020; Malanchev et al., 2020; Martínez-Galarza et al., 2020; Nun et al., 2014; Pruzhinskaya et al., 2019; Rebbapragada et al., 2009; Sadeh, 2019; Solarz et al., 2017; Soraisam et al., 2020; Villar et al., 2020b, 2021; Webb et al., 2020). These approaches predominantly use unsupervised clustering algorithms such as Density-Based Spatial Clustering of Applications with Noise (DBSCAN) (e.g. Giles & Walkowicz, 2019; Webb et al., 2020), RNN-based autoencoders that identify anomalies in a lower dimensional subspace (e.g. Sadeh, 2019; Villar et al., 2021), or outlier detection algorithms such as Isolation Forests (e.g. Giles & Walkowicz, 2019; Ishida et al., 2019a; Lochner & Bassett, 2020; Malanchev et al., 2020; Pruzhinskaya et al., 2019) and one-class support vector machines (e.g. Solarz et al., 2017). These approaches are effective at identifying anomalies once the full light curve has been observed, but many of them prove problematic for real-time detection in large-scale transient surveys. However, Soraisam et al. (2020) and Villar et al. (2021) have recently developed some of the first methods that perform real-time anomaly detection. Villar et al. (2021) uses a variational recurrent autoencoder to learn an encoded form of each light curve before obtaining anomaly scores by passing the encoded form into an isolation forest. Soraisam et al. (2020) uses the distribution of magnitude changes over time intervals in a population of light curves and computes the likelihood of a new observation being consistent with the population to identify outliers. In this chapter, we employ a unique method that performs regression over light curves to predict future fluxes, and uses the deviation between the predictions and observations to identify anomalies.

This chapter is organised as follows. In Section 3.3 we detail the ZTF light curve simulations and preprocessing methods used in this analysis. In Section 3.4 we develop two independent autoregressive models, the first being a probabilistic deep neural network that predicts future fluxes in a light curve, and the second being a Bayesian parametric model of a transient class built from the Bazin model of a light curve (Bazin et al., 2009). We present and compare the results of these two models at fitting transients and identifying anomalies in Section 3.5. We then also apply our models to real ZTF observational data taken from the public MSIP data stream in Section 3.6, and finally, in Section 3.7, we present our conclusions and discuss future applications of our work.

3.3 Data

3.3.1 Zwicky Transient Facility

The Zwicky Transient Facility (ZTF, Bellm et al., 2019) is the first of the new generation of optical synoptic survey telescopes and is a precursor survey to the upcoming LSST. It builds upon the infrastructure of the Palomar Transient Factory (PTF, Rau et al., 2009) using the 48-inch Schmidt telescope yielding an order of magnitude improvement in survey speed. It employs a 47 deg^2 field-of-view camera to scan more than 3750 deg^2 per hour to a depth of 20.5-21 mag (Graham et al., 2019). Using a prototype of the LSST alert distribution system, it is streaming up to one million transient alerts per night in two passband filters (gr). Due to this unprecedented data volume, it is necessary to build automated algorithms capable of processing and sifting through this amount of data.

ZTF has been producing transient observations since 2017, and has successfully identified several thousand supernovae. In Section 3.6, we detail our collection of over 2000 supernova light curves from the public ZTF Mid Scale Innovations Program (MSIP) survey to illustrate the performance of our method on real ZTF observations.

3.3.2 Simulations

The number of confirmed ZTF SNe is impressive, but the distribution is dominated by SNe Ia (see Section 3.6). Neural network based algorithms notoriously require a large training set before they are able to develop a model that generalises well to new data. Thus, while we may be able to create a good training set for SNe Ia, there are very few observations of many of the other classes, and we are not able to create a dataset of these classes that encompasses the variety of objects we expect to observe, even with significant data augmentation.

To this end, we used simulations that match the observing properties of the ZTF as described in Section 2.3 for the results shown in Section 3.5. These simulations were created using the SNANA (Kessler et al., 2009) software developed for the Photometric LSST Astronomical Time-series Classification Challenge (PLAsTiCC, Kessler et al., 2019; The PLAsTiCC team et al., 2018). As described in Section 2.3, the simulations were made using a year's worth of observing logs from the public MSIP survey at the ZTF. The simulated light curves mimic the ZTF observing properties with a median cadence of 3 days in the g and r passbands. We simulated approximately 10,000 events for each of the following classes: SNe Ia, SNe Ibc, SNe II, kilonovae, superluminous supernovae (SLSNe), tidal disruption events (TDEs), pair-instability supernovae (PISNe), intermediate luminosity transients (ILOTs), calcium-rich gap transients (CARTs), microlensing from binary star systems (uLens-BSR). The latter four classes were used as the anomaly class in the aforementioned PLAsTiCC challenge; and hence, we built models for each of the former 6 classes, and included the latter four for testing purposes. Example light curves from each class are illustrated in Figures

1-3 of Kessler et al. 2019. Each simulated transient consists of a time-series of flux and flux uncertainty measurements in the g and r ZTF passbands, an indicator of whether the flux was a detection or non-detection, sky position, Milky Way dust reddening, and a host galaxy redshift.

We define the date of *trigger* throughout this chapter as the first detection in a light curve, defined as the first observation that exceeds a 5σ signal-to-noise (S/N) measurement in a difference image. Hence, in the rest of this chapter, time t refers to the number of Modified Julian Date (MJD) days since trigger:

$$t = \mathrm{MJD} - \mathrm{MJD}_{\mathrm{trigger}} \tag{3.1}$$

The models and LSST simulations developed for PLAsTiCC, which these simulations are based from, were validated using numerous techniques as described in Hložek et al. (2020). We split the total set of transients for each class into two parts: 80% for the *training set* and 20% for the *testing set*, respectively. The *training set* is used to train the model that predicts future fluxes, while the *testing set* is used to test the performance of the model.

3.3.3 Preprocessing

One of the most important aspects in an effective learning algorithm is the quality of the training set. We ensured that the data were processed in a uniform and systematic way before training the model. We perform 'sigma clipping' to reject photometric points with flux uncertainties that are more than 3σ from the mean uncertainty in each passband, and iteratively repeat this clipping 5 times. Next, we correct the light curves for Milky Way extinction using the reddening function of Fitzpatrick (1999). We assume an extinction law, $R_V = 3.1$, use the central wavelength of each ZTF filter (g: 4767 Å, r: 6215 Å) and the sky position to compute the line-of-sight reddening caused by the Milky Way and de-redden each light curve[1].

As we are only interested in parts of the light curve near trigger, we ignored any observations more than 70 days before trigger and removed any data more than 150 days from the first observation after $t = -70$.

[1] We use the extinction code: https://extinction.readthedocs.io

3.3.3.1 Training set preparation

The ZTF observations are irregularly sampled due to intranight cadence choices and seasonal constraints that lead to naturally arising temporal gaps. However, our neural network framework requires regular time-sampling of the input data. Thus, despite the ZTF observations having a roughly 3-day cadence, we interpolate our observed data onto a grid with a cadence of exactly 3 days. This interpolation is not necessary for our Bazin method, but for the sake of comparing the results between the two methods, we use the interpolated data as the input for both models described in Section 3.4.

Gaussian process (GP) regression (Rasmussen & Williams, 2006) has been shown to be effective for astronomical light curve modelling and interpolation (Boone, 2019; Lochner et al., 2016). However, typically when GPs are used for preprocessing light curves to interpolate irregularly sampled data to a regular grid (e.g. Boone, 2019; Villar et al., 2021), the GP is conditioned on the entire light curve and makes use of long-range covariance kernels (e.g. squared-exponential). For the purposes of retrospective analyses using the full light curve, this interpolation method is effective; however, for real-time usage this approach unrealistically uses future observations that would not be available at a particular prediction time. Instead we use linear interpolation which respects causality as follows. The linearly interpolated value at a given grid time depends only on the two neighboring observations. Hence, relative to a prediction time $T_{\rm pred}$ between two observation times $t_{{\rm obs},i} < T_{\rm pred} < t_{{\rm obs},i+1}$, the interpolated values at all grid points $T_{{\rm grid},j}$ before $t_{{\rm obs},i} < T_{\rm pred}$ depend only on past observations at times earlier or equal to $t_{{\rm obs},i} < T_{\rm pred}$. In contrast, the GP conditioned on the full light curve produces interpolated values at earlier grid times $T_{{\rm grid},j} < T_{\rm pred}$ that depend, though the covariance kernel, on $t_{{\rm obs},i+1} > T_{\rm pred}$ and all future data points and, therefore, does not respect causality in real-time applications. Linear interpolations also have the added benefit over GPs of not over-smoothing and being less computationally intensive. With few and noisy observations characteristic of early real-time data, the GP may also be more prone to overfitting the light curve than linear interpolation.

Our method for obtaining linear interpolations with uncertainties is detailed as follows. For each data point of transient s in passband p at time t since trigger in a light curve, we used the observed flux $\hat{F}_{spt}$ and uncertainty $\hat{\sigma}_{F,spt}$ as the mean and standard deviation of a normal distribution to draw 100 random fluxes indexed by i,

$$\hat{F}_{spt,i} \sim \mathcal{N}(\hat{F}_{spt}, \hat{\sigma}_{F,spt}). \tag{3.2}$$

This gives 100 different replications of the observed light curve. We linearly interpolate each of these 100 generated light curves at 3-day intervals. We obtain the interpolated flux and flux uncertainty by computing the mean and standard deviation of these light curves draws, respectively:

$$D_{spt} = \frac{1}{100}\sum_{i=1}^{100}\hat{F}_{spt,i}, \tag{3.3}$$

$$\sigma_{D,spt} = \sqrt{\frac{1}{100}\sum_{i=1}^{100}\left(\hat{F}_{spt,i} - D_{spt}\right)^2} \tag{3.4}$$

We removed any data more than 150 days from the first observation after $t = -70$, and hence with the 3-day interpolations make a matrix of length $N_t = 50$, with each point for transient s, passband p, and interpolated time t having a flux and uncertainty as follows,

$$\boldsymbol{X}_{spt} = [D_{spt}, \sigma_{D,spt}] \tag{3.5}$$

We similarly define the output flux predictions of our models described in Section 3.4 as a vector of the predictive flux and uncertainty,

$$\boldsymbol{Y}_{spt} = [y_{spt}, \sigma_{y,spt}] \tag{3.6}$$

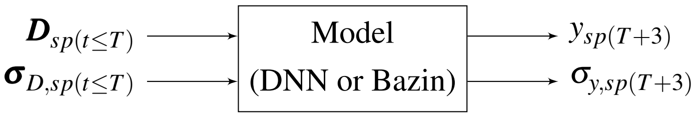
While not strictly necessary in our architecture, the fixed length vectors are useful for passing the data to the neural network framework. The final input matrix $\boldsymbol{X}_s$ for each transient s is a matrix of shape $N_t \times 2N_p$ where the rows are composed of the interpolated flux and flux uncertainty in each of the N_p passbands across the N_t time-steps. If contextual information such as the redshift or host galaxy properties were known, we could include this information as a extra columns in the input matrix. However, as this information is not always available without additional host spectra, we have not included it in this work, but note that our framework allows for an easy incorporation of contextual data. Future work should aim to use host galaxy information to improve transient identification. Studies by Foley & Mandel 2013 and Gagliano et al. 2021 have shown that using only host galaxy properties without any photometric data can achieve $\sim 70\%$ accuracy when classifying SNe Ia and CC SNe.

3.4 Models

Our methods for anomaly detection involve first developing an autoregressive sequence model of a transient class, and then using the model's ability to predict future fluxes as an anomaly score (or conversely, a goodness of fit score). We develop two methods for regressing over a transient. The first is a probabilistic deep neural network (DNN) approach using Temporal Convolutional Networks (TCNs) (described in Section 3.4.1), and the second is a Bayesian parametric approach using the flexible Bazin function Bazin et al. (2009) of transients (described in Section 3.4.2).

Each model aims to do real-time detection, and is hence causal, using only past values to predict future values. Specifically, our model is a function that predicts future fluxes in a time-series as well as the uncertainty of that prediction; it then compares the prediction with the observed data to obtain an anomaly score.

In the following two subsections (§3.4.1, §3.4.2), we describe our two approaches of developing a function that maps the interpolated fluxes up to time T onto flux predictions $y_{sp(T+3)}$ and predictive uncertainties $\sigma_{y,sp(T+3)}$ three days after a given set of observations:


$\boldsymbol{D}_{sp(t\leq T)}$
$\boldsymbol{\sigma}_{D,sp(t\leq T)}$
Model
(DNN or Bazin)
$y_{sp(T+3)}$
$\sigma_{y,sp(T+3)}$

In Section 3.4.3, we define an anomaly score metric that uses the discrepancy between the fluxes D and predictions y to quantify anomalies.

The DNN approach builds a neural network that effectively performs regression over past data in order to predict the flux 3 days in the future. On the other hand, the Bazin approach performs regression over time to predict the flux at any time. We then feed in partial light curves into the Bazin model and infer a prediction 3 days after given data to obtain anomaly scores comparable with the DNN.

3.4.1 Probabilistic Neural Network

3.4.1.1 Model definition

The DNN is an autoregressive mapping function that aims to map an input multi-passband light curve matrix, $\boldsymbol{X}_{s(t\leq T)}$, for transient s up to an interpolated time T, onto an output

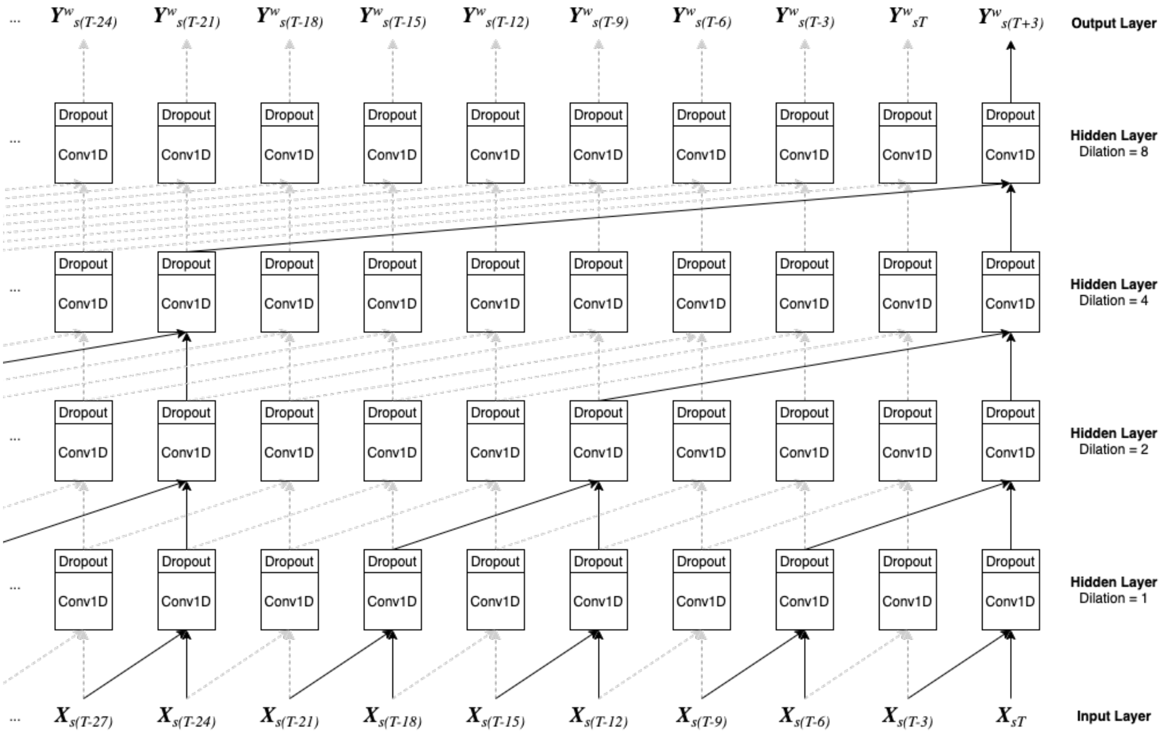


Fig. 3.1 Temporal Convolutional Neural Network architecture used in this work. Each column in the diagram is a subsequent time-step from left to right, and each time-step is 3-days after the previous one. The bottom row is the input light curve (from equation 3.5) where each input is a vector of the interpolated flux and flux uncertainty in all passbands for a transient s at a time t. The input fluxes and uncertainties of two adjacent time-steps are passed into a residual block consisting of a 1D convolutional neural network layer (Conv1D) with dropout. While not shown in the figure, the residual block also contains a second Conv1D layer with dropout. The outputs of these are then convolved with the outputs from some previous time-steps in the above hidden layers as shown in the diagram, until the final Output Layer is the predicted light curve at the following time-step (from equation 3.6). The solid arrows show how the prediction $\boldsymbol{Y}^w_{s(T+3)}$ is made, and the gray dashed arrows show the neural network layers that lead to all other predictions. The network is causal, whereby new predictions only use information from previous time-steps in the light curve. We set the dropout rate to 20% for all layers in the network. We build this model using the `Keras` and `TensorFlow Probability` libraries after adapting the TCN model from Bai et al. (2018) and their code in https://github.com/philipperemy/keras-tcn.

multi-passband flux vector at the next time-step $T+3$ (where we recall that each time-step is 3 days after the previous interpolated time),

$$\boldsymbol{Y}^{w}_{s(T+3)} = \boldsymbol{f}_T(\boldsymbol{X}_{s(t\leq T)}; \boldsymbol{w}) \tag{3.7}$$

where $\boldsymbol{w}$ are the parameters (i.e. weights and biases) of the network. We define $\boldsymbol{X}_{s(t\leq T)}$ as the matrix $\boldsymbol{X}_s$ but up to a time T in each of its passbands. The model output prediction $\boldsymbol{Y}^{w}_{s(T+3)}$ is a $1\times 2N_p$ vector consisting of the predicted mean flux $\tilde{y}_{sp(T+3)}(\boldsymbol{w})$ and intrinsic uncertainty $\tilde{\sigma}_{\mathrm{int},sp(T+3)}(\boldsymbol{w})$ in the g and r passbands at the next time-step for a particular set of network weights (these outputs are explained in Section 3.4.1.3). The model $\boldsymbol{f}_T(\boldsymbol{X}_{s(t\leq T)}; \boldsymbol{w})$ in equation 3.7 is represented by the complex DNN architecture illustrated in Fig. 3.1, and the details of the architecture are described in the following subsection.

3.4.1.2 Model architecture

We developed a deep neural network architecture as our first approach for the autoregressive sequence model that learns the function described in equation 3.7. The problem of time-series prediction falls in the wider machine-learning area of *sequence learning*, whereby the input size of our data is not required to be fixed. Recurrent Neural Networks (RNNs) such as Long Short Term Memory (LSTM, Hochreiter & Schmidhuber 1997) Networks and Gated Recurrent Units (GRU, Cho et al. 2014) are considered the default starting point for sequence modelling tasks in the machine learning community after they were shown to achieve state-of-the-art performance in many benchmark time-series and sequential data applications (e.g. Bahdanau et al., 2014; Che et al., 2018; Chung et al., 2014; Jozefowicz et al., 2015; Pascanu et al., 2013; Sutskever et al., 2014; Zhang et al., 2015). RNN's ability to retain an internal memory of long-term temporal dependencies of variable length observations made it well suited for time-series applications, and it has been shown to be successful in light curve classification (e.g. Charnock & Moss, 2017; Jamal & Bloom, 2020; Martínez-Palomera et al., 2020; Möller & de Boissière, 2020; Moss, 2018; Muthukrishna et al., 2019a).

However, RNNs suffer from a few drawbacks not present in Convolutional Neural Network (CNN) approaches that have been so successful in image analysis and a range of other groundbreaking problems. Most notably, RNNs are notoriously slow and difficult to train using standard stochastic gradient descent (SGD) algorithms (Bai et al., 2018; Pascanu et al., 2013). In the past couple of years, Temporal Convolutional Networks (TCNs, first proposed in Lea et al. 2016) have risen as a powerful alternative to RNNs. A thorough systematic empirical evaluation of RNNs and TCNs conducted by Bai et al. (2018) suggest that TCNs are able to convincingly outperform LSTMs and GRUs across a broad range of sequence modelling tasks. In particular, Bai et al. (2018) demonstrates that TCNs exhibit a substantially longer memory of sequential data (being able to capture a longer history of data in the model), have a more flexible receptive field size (being able to control how many historic data points to remember), are much faster to train because of their parallelism (where RNNs need to wait for preceding blocks to complete but convolutions can be done in parallel since the same filter is used in each layer), are less memory intensive, and are able to capture local information through convolutions along with temporal information (Bai et al., 2018; Kalchbrenner et al., 2016; Lea et al., 2016). Furthermore, as TCNs are much simpler and clearer than RNNs, we have used this architecture in favour of RNNs in this chapter. In practice, we found that the TCNs were much faster to train, but we did not notice significant differences in the performance of our TCN architecture when we compared it to a similar LSTM/GRU architecture that was used in Muthukrishna et al. (2019a).

The TCN architecture used in this work is illustrated in Figure 3.1, and is based on the model developed by Bai et al. (2018)[2]. We used the high level `Python` API, `Keras` (Chollet et al., 2015), and the `TensorFlow Probability` library that are built on the efficient `TensorFlow` machine learning system (Abadi et al., 2016) to develop our deep probabilistic neural network model. We describe the architecture in detail here.

***Input*:** The input at each time-step is the vector $\boldsymbol{X}_{st}$ for transient s at time t. Each $\boldsymbol{X}_{st}$ input has shape $1 \times 2N_p$ containing the interpolated flux and flux uncertainty for each of the N_p passbands.

Residual block: Each residual block performs a 1D convolution on two vector inputs using a sigmoid activation function on the neurons, and then applies dropout to each layer. While not shown in Figure 3.1, each residual block also contains a second 1D convolution and dropout layer. We ensure that the convolutions are dilated and causal, whereby each output only uses information from preceding time-steps. We set the dilations to 1, 2, 4, and 8 such that the total receptive field includes $2 \times 8 = 16$ time-steps (equivalent to 48 days as we've set each time-step to be 3 days apart). Such a receptive field was considered a sufficient light curve history to make a prediction.

[2] The TCN code was adapted from https://github.com/philipperemy/keras-tcn

Conv1D: A 1D convolution is applied to the two vector inputs that each have a shape $1 \times 2N_p$. The convolutional kernel size is 2, applied to the two preceding time-steps in the input layer, and dilated to other time-steps in the hidden layers as illustrated in the Figure.

Dropout: We also implement dropout regularization to each layer of the neural network to reduce overfitting during training. This is an important step that effectively ignores randomly selected neurons such that their contribution to the network is temporarily removed. This process causes other neurons to more robustly handle the representation required to make predictions for the missing neurons, making the network less sensitive to the specific weights of any individual neuron. We set the dropout rate to 20% of the neurons present in the previous layer. This effectively means that each neuron's weight has a 20% probability of being set to zero. While this approach is ubiquitously used for regularisation during training, we also apply dropout during test time to obtain model uncertainties (see Section 3.4.1.3).

Neurons: The output of each neuron in a neural network layer is expressed as a function of the weighted sum of the connections to it from the previous layer.

Activation function: As with any neural network, each neuron applies an activation function to bring non-linearity to the network and hence help it to learn complex patterns in the data. We use a sigmoid activation function for the 1D convolutional layers as, after some testing, it appeared to have more stability while training the network when compared to a ReLU function. This bounded the outputs of the neurons to values between 0 and 1 and ensured that the weights did not become too large.

Masking Layer: TCNs have the advantage over standard CNNs of allowing variable length input sequences. However, the `Python` API requires a fixed length input for ease of computation. To employ the TCN's flexibility, we make an $N_t = 50$ length input matrix for each light curve, but set the time-steps where data does not exist to an arbitrary value. We then use `Keras`'s Masking Layer to mask this value and hence ensure that time-steps where data are not available are not used in the model.

3.4.1.3 Capturing uncertainties in the model

Standard deep learning tools for regression and classification do not capture model uncertainty. Nevertheless, the power and success of neural networks at a wide range of benchmark problems has led to their widespread use in science. They are particularly useful when the underlying physical processes that generated the data is not well-understood. However, gaining an intuitive understanding of the neural network's high-dimensional model is difficult and often impossible. In fact, a common and significant issue in deep learning is its overconfident predictions on unseen data (e.g. Guo et al., 2017). Getting a neural network to say that it "does not know" and to state its confidence in a prediction is imperative for its use in science. The softmax probability output in many neural network classification problems is often erroneously interpreted as model confidence in spite of it being infamously falsely overconfident (e.g. Gal & Ghahramani, 2015a; Goodfellow et al., 2015; Szegedy et al., 2014). Obtaining uncertainties on predictions is important to help overcome these issues and for the continued use of deep learning in science.

To characterise the intrinsic uncertainty of our network's ability to represent a light curve, we build a probabilistic neural network using the `Tensorflow Probability` `Python` library. While typical neural networks model the output as a point estimate, a probabilistic neural network allows us to easily model the output of the DNN as a distribution. In this work, we model the output as a Normal distribution parameterised by the predictive mean $\tilde{y}_{spt}(\boldsymbol{w})$ and standard deviation $\tilde{\sigma}_{\mathrm{int},spt}(\boldsymbol{w})$ for a particular set of network weights w. The predictions $\tilde{y}_{spt}(\boldsymbol{w})$ and $\tilde{\sigma}_{\mathrm{int},spt}(\boldsymbol{w})$ are components of the vector $\boldsymbol{Y}_{spt}^{w}$ from equation 3.7. We interpret the learned uncertainty $\tilde{\sigma}_{\mathrm{int},spt}(\boldsymbol{w})$ as being intrinsic to problem. We include it because we know that our DNN model is not a perfect representation of a light curve, and even if we had no measurement error and had an infinite training set, there would still be some discrepancy between our DNN predictions and the observed light curves.

A Bayesian neural network enables us to also quantify the uncertainty in our model's predictions of the outputs, $\tilde{y}_{spt}(\boldsymbol{w})$ and $\tilde{\sigma}_{\mathrm{int},spt}(\boldsymbol{w})$. The key advantage of a Bayesian neural network over a standard neural network, is that we are able to sample over a posterior distribution of network parameters (i.e. the weights and biases). The ideal way to perform a Bayesian inference over the neural network model would be to sample over the model parameters with a method such as Markov Chain Monte Carlo (MCMC). However, the huge number of parameters in a deep neural network make this a computationally intractable problem. Instead, an approach called Monte Carlo (MC) dropout sampling that places a Bernoulli distribution over the network weights using the commonly used dropout regularisation technique has become the popular approach for implementing approximate Bayesian neural networks (see Gal & Ghahramani (2015a) for an explanation of how MC dropout approximates a Bayesian NN). The method is significantly simpler to implement than standard neural network Variational Inference (VI) approaches (such as *Bayes by Backprop* (Blundell et al., 2015), the *Flipout* *estimator* (Wen et al., 2018), and the *Reparameterization estimator* (Kingma & Welling, 2013)) - simply requiring dropout to be applied to all the network weights during validation (instead of just training). It has the further advantage over standard approaches to VI in neural networks of not increasing training time or reducing test accuracy. Throughout this work, we use MC dropout with our probabilistic neural network to estimate the predictive uncertainty. We do this by collecting the results of stochastic forward passes through the network as approximate posterior draws of $\boldsymbol{Y}^{w}_{spt}$, and use the mean and standard deviation of these draws as our marginal predictive mean y_{spt} and predictive uncertainty $\sigma_{y,spt}$.

3.4.1.4 Model loss function

Before defining the loss function, we first develop a generative model of the latent flux of a transient s in passband p 3 days in the future at time $T+3$. We aim to model the underlying latent flux with the neural network as follows,

$$F_{sp(T+3)}(\boldsymbol{w}) = \tilde{y}_{sp(T+3)}(\boldsymbol{w}) + \varepsilon_{\text{int},sp(T+3)}(\boldsymbol{w}), \tag{3.8}$$

where the error $\varepsilon_{\text{int},sp(T+3)}(\boldsymbol{w}) \sim \mathcal{N}\left(0, \tilde{\sigma}^2_{\text{int},sp(T+3)}(\boldsymbol{w})\right)$ is a zero-mean Gaussian random variable with variance $\tilde{\sigma}^2_{\text{int},sp(T+3)}(\boldsymbol{w})$. Thus, we write the predictive distribution of the latent flux as follows,

$$\mathcal{P}(\boldsymbol{F}_{s(T+3)}|\boldsymbol{X}_{s(t\leq T)}, \boldsymbol{w}) = \prod_{p=1}^{N_p} \mathcal{N}\left(F_{sp(T+3)}(\boldsymbol{w}) \mid \tilde{y}_{sp(T+3)}(\boldsymbol{w}), \tilde{\sigma}^2_{\text{int},sp(T+3)}(\boldsymbol{w})\right). \tag{3.9}$$

Next, a generative model of the observed flux is derived by adding a measurement error to the latent flux as follows,

$$D_{sp(T+3)} = F_{sp(T+3)}(\boldsymbol{w}) + \varepsilon_{D,sp(T+3)}, \tag{3.10}$$

where we assume that the measurement error $\varepsilon_{D,sp(T+3)} \sim \mathcal{N}(0, \sigma^2_{D,sp(T+3)})$ is a zero-mean Gaussian random variable with variance $\sigma^2_{D,sp(T+3)}$.

Typically, researchers will not use the uncertainty in the data within the loss function (e.g. Jamal & Bloom 2020; Villar et al. 2020b, however, work by Naul et al. 2018 included data uncertainty without model uncertainty). In this work, we construct our loss function to include both predictive uncertainties $\tilde{\sigma}_{\text{int},sp(T+3)}(\boldsymbol{w})$ and flux uncertainties $\sigma_{D,sp(T+3)}$. Given equations 3.8 and 3.10, we write the likelihood function of the probabilistic DNN as follows,

$$\begin{aligned}\mathcal{P}(\boldsymbol{D}_{s(T+3)}|\boldsymbol{X}_{s(t\leq T)},\boldsymbol{w}) &= \prod_{p=1}^{N_p} \mathcal{N}\left(D_{sp(T+3)} \mid \tilde{y}_{sp(T+3)}(\boldsymbol{w}), \tilde{\sigma}^2_{\text{int},sp(T+3)}(\boldsymbol{w}) + \sigma^2_{D,sp(T+3)}\right) \\ &= \prod_{p=1}^{N_p} \left(2\pi(\tilde{\sigma}^2_{\text{int},sp(T+3)}(\boldsymbol{w}) + \sigma^2_{D,sp(T+3)})\right)^{-0.5} \\ &\times \exp\left(-0.5\frac{(\tilde{y}_{sp(T+3)}(\boldsymbol{w}) - D_{sp(T+3)})^2}{\tilde{\sigma}^2_{\text{int},sp(T+3)}(\boldsymbol{w}) + \sigma^2_{D,sp(T+3)}}\right).\end{aligned} \tag{3.11}$$

Following Gal & Ghahramani (2015a), we define the prior over the weights as a zero-mean Normal distribution,

$$\mathcal{P}(\boldsymbol{w}) = \mathcal{N}(\boldsymbol{w} \mid 0, \boldsymbol{I}/l^2), \tag{3.12}$$

where $\boldsymbol{I}$ is the identity matrix and l is the prior length-scale that regularises how large the weights can be. The posterior over the weights is given by the product of the prior distribution and the likelihood distribution over all N_s transients at all N_t time-steps,

$$\mathcal{P}(\boldsymbol{w}|\boldsymbol{X}) \propto \mathcal{P}(\boldsymbol{w}) \prod_{s=1}^{N_s} \prod_{T=-70}^{80} \mathcal{P}(\boldsymbol{D}_{s(T+3)}|\boldsymbol{X}_{s(t\leq T)},\boldsymbol{w}), \tag{3.13}$$

where we ignore the Bayesian evidence as a scaling constant that is unnecessary for this work. We would ideally like to sample the negative log posterior while training our DNN, and so we derive the log prior from equation 3.12 as $\log \mathcal{P}(\boldsymbol{w}) = \text{constant} - l^2||\boldsymbol{w}||_2^2/2$. We can ignore the additive constant not necessary for our optimisation and follow Gal & Ghahramani (2015a) to implement the log prior by including an L_2 regularisation term $\lambda||\boldsymbol{w}||_2^2$ weighted by some weight decay that averages over the number of transients N_s and time-steps N_t,

$$\lambda = \frac{l^2(1-d)}{2N_sN_t}, \tag{3.14}$$

where d is the dropout rate (set to 0.2 in this work), and we set $l = 0.2$ consistent with work by Gal & Ghahramani (2015a)[3]. Here, we have included the $(1-d)$ term to account for the dropout regularisation used in our work. With this term, the L_2 regularisation term $\lambda||\boldsymbol{w}||_2^2$ matches the log prior, $\log \mathcal{P}(\boldsymbol{w})$, averaged over the number of transients and time-steps.

[3] See Section 4.2 of Gal & Ghahramani (2015b) for a detailed explanation of this prior and https://github.com/yaringal/DropoutUncertaintyExps/blob/master/net/net.py for an example implementation of this L_2 regularisation by Yarin Gal.

However, we point out that our λ slightly differs from equation 18 of Gal & Ghahramani (2015b) because we use the negative log-likelihood instead of a squared loss as the cost function of our DNN. Furthermore, we also add the caveat that because of this difference of loss functions and our inclusion of a probabilistic neural network that models the output as a Normal distribution instead of a point estimate, it is not clear that the demonstration of MC dropout as an approximation to Bayesian neural networks in Gal & Ghahramani (2015a) necessarily holds true in our work. Future machine learning research should check the validity of MC dropout as a Bayesian approximation in a broader range of neural network architectures.

Since we use dropout regularisation, we define a dropout objective function over all time-steps and over all transients that we aim to minimise while training the neural network model as follows,

$$\text{obj}(\boldsymbol{w}) = \sum_{s=1}^{N_s} \sum_{T=-70}^{80} \left[-\log \mathcal{P}(\boldsymbol{D}_{s(T+3)} | \boldsymbol{X}_{s(t \leq T)}, \boldsymbol{w}) + \lambda ||\boldsymbol{w}||_2^2 \right] \tag{3.15}$$

where we sum the log-likelihood and L_2 regularisation term over all N_t time-steps (between -70 and 80 days) and N_s transients in the training set. To train the DNN and determine optimal values of its parameters $\hat{\boldsymbol{w}}$, we minimise the dropout objective function with the sophisticated and commonly used `Adam` gradient descent optimiser (Kingma & Ba, 2015).

To make predictions, we evaluate the predictive distribution of the latent flux defined as follows,

$$\mathcal{P}(\boldsymbol{F}_{s(T+3)} | \boldsymbol{X}_{s(t \leq T)}) = \int \mathcal{P}(\boldsymbol{F}_{s(T+3)} | \boldsymbol{X}_{s(t \leq T)}, \boldsymbol{w}) \mathcal{P}(\boldsymbol{w} | \boldsymbol{X}) d\boldsymbol{w}, \tag{3.16}$$

where we are marginalising over the weights of the network by integrating the product of the predictive distribution of the latent flux given the network weights (first term in the integrand and defined in equation 3.9) and the posterior distribution over the network weights (second term in the integrand and defined in equation 3.13). The integral is intractable, and so we approximate it by using Monte Carlo dropout at inference time to sample the posterior distribution, as described in Gal & Ghahramani (2015a). We draw 100 samples from the posterior $\mathcal{P}(\boldsymbol{w}|\boldsymbol{X})$ by running 100 forward passes of the neural network for a given input. Since we are using a probabilistic neural network and have modelled the output as a Normal distribution (as shown by our model of the latent flux in equation 3.8), each run of the neural network outputs both a mean $\tilde{y}_{sp(T+3)}(\boldsymbol{w}_{\text{draw}})$ and standard deviation $\tilde{\sigma}_{\text{int},sp(T+3)}(\boldsymbol{w}_{\text{draw}})$. To include the variance of each draw in the marginal predictive uncertainty $\sigma_{y,sp(T+3)}$, we compute $F_{sp(T+3)}(\boldsymbol{w}_{\text{draw}}) \sim \mathcal{N}\left(\tilde{y}_{sp(T+3)}(\boldsymbol{w}_{\text{draw}}), \tilde{\sigma}^2_{\text{int},sp(T+3)}(\boldsymbol{w}_{\text{draw}})\right)$. We estimate the marginal predictive mean and uncertainty as the sample mean and standard deviation of the 100 values of $F_{sp(T+3)}(\boldsymbol{w}_{\text{draw}})$ taken from the 100 forward passes of the neural network, respectively:

$$y_{sp(T+3)} = \frac{1}{100} \sum_{\text{draw}=1}^{100} F_{sp(T+3)}(\boldsymbol{w}_{\text{draw}}), \tag{3.17}$$

$$\sigma_{y,sp(T+3)} = \sqrt{\frac{1}{100} \sum_{\text{draw}=1}^{100} \left(F_{sp(T+3)}(\boldsymbol{w}_{\text{draw}}) - y_{sp(T+3)}\right)^2}. \tag{3.18}$$

We use these to compute the anomaly scores discussed in Section 3.4.3.

3.4.2 Parametric Bayesian Bazin function

3.4.2.1 Model Definition

The fast inference speed of the DNN model makes it scaleable for the enormous data streams expected from surveys such as LSST. However, the large number of parameters in the model makes it difficult to ascertain how the model makes decisions. Thus, as a comparison, we have built a Bayesian model of each transient light curve based on the widely used phenomenological Bazin function from Bazin et al. (2009). This method has the advantage of not requiring regularly sampled input data. However, to make it comparable to the DNN method, we use the interpolated fluxes $\boldsymbol{D}_{spt}$ and uncertainties $\boldsymbol{\sigma}_{D,spt}$ as the input data.

We begin by augmenting the standard Bazin function with an additional error term, $\varepsilon_{\text{int}}(t)$. We include this error because we know that the Bazin function is not a perfect representation of a light curve, and even if we had no measurement error, there would still be some discrepancy between the Bazin model and the observed light curve. Hence, we define a generative model of a transient's luminosity as follows,

$$L(t) = L_0 \left(\frac{e^{-(t-t_0)/\tau_{\text{fall}}}}{1 + e^{-(t-t_0)/\tau_{\text{rise}}}} + \varepsilon_{\text{int}}(t) \right) \tag{3.19}$$

where $L(t)$ is the luminosity as a function of time t in days since trigger, L_0, t_0, τ_{fall}, τ_{rise} are free parameters of the model, and $\varepsilon_{\text{int}}(t) \sim \mathcal{N}(0, \sigma^2_{\text{int}})$ is a zero-mean Gaussian random variable with intrinsic variance σ^2_{int}.

The Bazin model has the advantage of being much more interpretable than a DNN as the parameters can be intuitively understood with respect to the shape of a light curve, whereby,

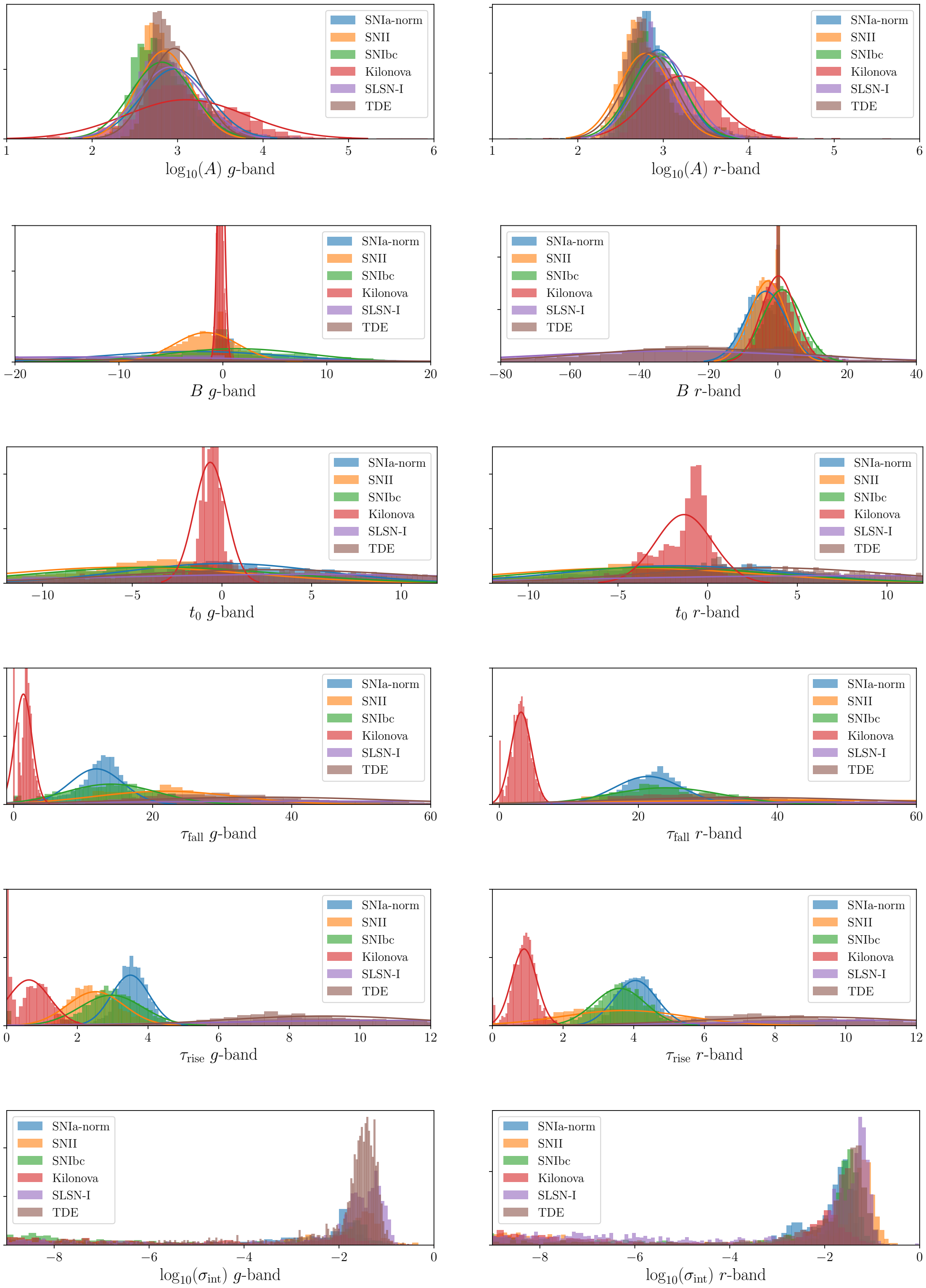


Fig. 3.2 The histograms of the best fit Bazin parameters for the population of light curves in each transient class and passband. We ignored any transient light curves that did not have 10 data points on each side of trigger. We modelled the population distribution as a multivariate Gaussian, and show the one-dimensional slices as the solid lines. We used this multivariate Gaussian as the priors for the Bazin parametric model defined in equation 3.24.

L_0 can be interpreted as the intrinsic luminosity of the transient, τ_{fall}, τ_{rise} relate to the sloping rise and fall times of the light curve, and t_0 relates to the time of peak brightness (see figure 3.3). We then model the latent flux of a transient s in passband p by first dividing equation 3.19 by $4\pi d^2$, where d is the distance to the transient object, and then adding a term for the measured background flux B,

$$F_{spt}(\boldsymbol{\theta}) = A\frac{e^{-(t-t_0)/\tau_{\text{fall}}}}{1+e^{-(t-t_0)/\tau_{\text{rise}}}} + B + A\varepsilon_{\text{int}}(t), \tag{3.20}$$

where $A = L_0/4\pi d^2$. Then, a generative model of the input flux is derived by adding a measurement error $\varepsilon_{D,spt}$, as follows,

$$D_{spt} = A\frac{e^{-(t-t_0)/\tau_{\text{fall}}}}{1+e^{-(t-t_0)/\tau_{\text{rise}}}} + B + A\varepsilon_{\text{int}}(t) + \varepsilon_{D,spt}, \tag{3.21}$$

where we assume that the measurement error $\varepsilon_{D_{spt}} \sim \mathcal{N}(0, \sigma^2_{D,spt})$ is a zero-mean Gaussian random variable with variance $\sigma^2_{D,spt}$. The mean of the model in equation 3.21 is the Bazin function described in Bazin et al. (2009) with free parameters $\boldsymbol{\theta} = [\log_{10}(A), B, t_0, \tau_{\text{fall}}, \tau_{\text{rise}}, \log_{10}(\sigma_{\text{int}})]$.

In Figure 3.3, we plot equation 3.20, and explore the effect of each of the free parameters on the model light curve shape.

To model a partial light curve from -70 days before trigger or when observations begin up to time T, and given equations 3.20 and 3.21, we write the likelihood function as follows,

$$\begin{aligned} \mathcal{P}(\boldsymbol{D}_{sp}|\boldsymbol{t},\boldsymbol{\theta}) &= \prod_{t=-70}^{T} \mathcal{N}\left(D_{spt} \mid F_{spt}(\boldsymbol{\theta}), A^2\sigma^2_{\text{int}} + \sigma^2_{D,spt}\right) \\ &= \prod_{t=-70}^{T} \left(2\pi(A^2\sigma^2_{\text{int}} + \sigma^2_{D,spt})\right)^{-0.5} \exp\left(-0.5\frac{(F_{spt}(\boldsymbol{\theta}) - D_{spt})^2}{A^2\sigma^2_{\text{int}} + \sigma^2_{D,spt}}\right) \end{aligned} \tag{3.22}$$

3.4.2.2 Bayesian model and prior

We define a Bayesian model to fit each transient light curve in a particular passband as follows,

$$\mathcal{P}(\boldsymbol{\theta}|\boldsymbol{D}_{sp},\boldsymbol{t}) \propto \mathcal{P}(\boldsymbol{D}_{sp}|\boldsymbol{t},\boldsymbol{\theta})\mathcal{P}(\boldsymbol{\theta}) \tag{3.23}$$

where $\mathcal{P}(\boldsymbol{\theta}|\boldsymbol{D}_{sp},\boldsymbol{t})$ is the posterior distribution, $\mathcal{P}(\boldsymbol{D}_{sp}|\boldsymbol{t},\boldsymbol{\theta})$ is the likelihood function from equation 3.22, and $\mathcal{P}(\boldsymbol{\theta})$ is the prior distribution of each transient class. We have ignored

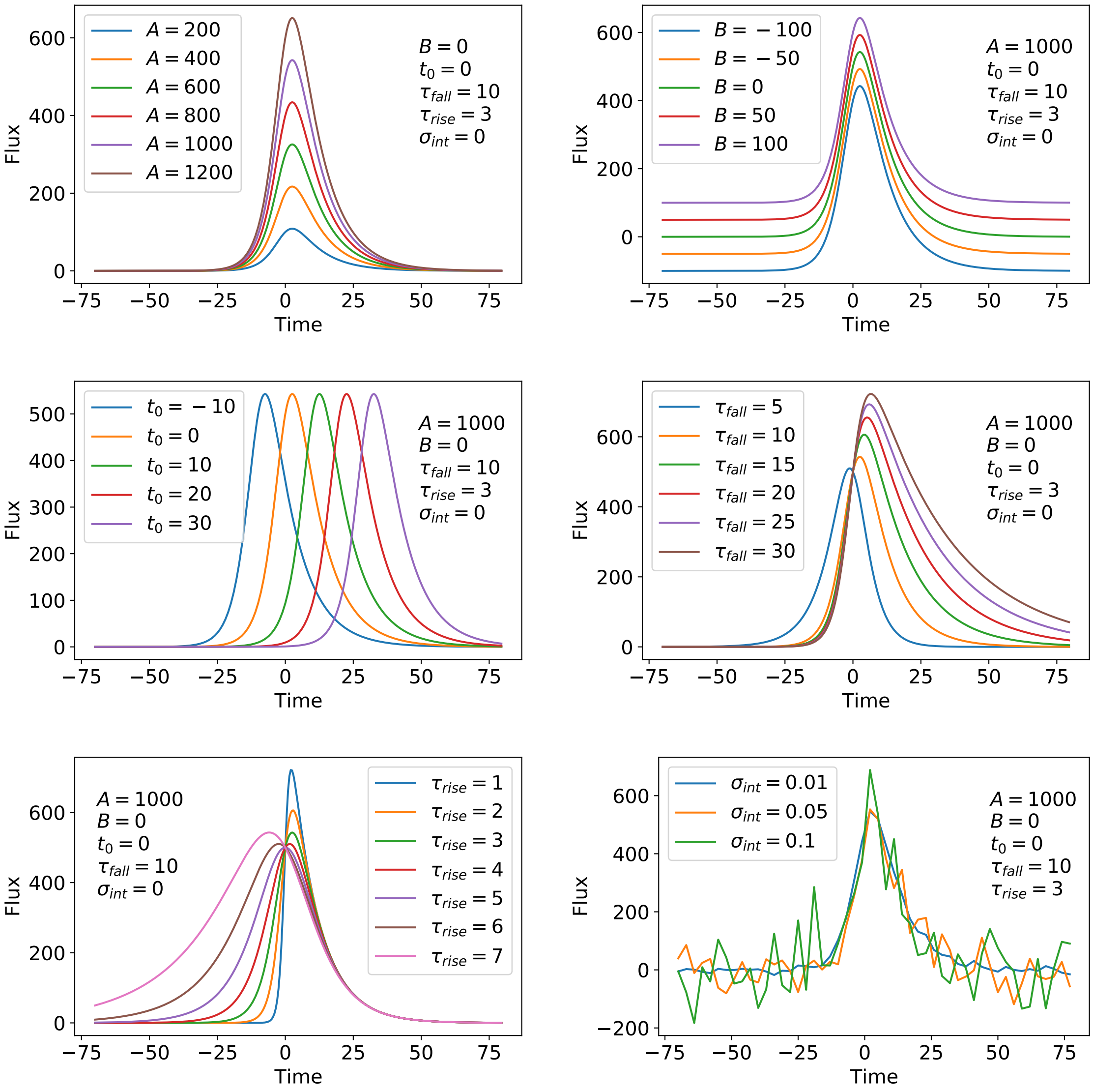


Fig. 3.3 The effect that changing different Bazin parameters has on the modelled light curve shape. We plot equation 3.20 using the baseline parameters, $A = 1000$, $B = 0$, $t_0 = 0$, $\tau_{\text{fall}} = 10$, $\tau_{\text{rise}} = 3$, and $\sigma_{\text{int}} = 0$. Each plot then varies one of these parameters to illustrate its effect on the light curve shape. A affects the light curve amplitude, B dictates the vertical shift, t_0 affects the horizontal shift, τ_{fall} affects the duration of the light curve decline, τ_{rise} affects the duration of the light curve rising to maximum, and σ_{int} affects the modelled noise on the light curve.

the evidence $\mathcal{P}(\boldsymbol{D}_{sp})$ as a scaling factor that is unnecessary for our optimisation and is not computed in this work.

We have chosen to base our prior on the distribution of fits of the transient population in each passband and class. We first compute the population distribution by fitting each light curve in the training set with the Bazin function in equation 3.20 and maximising the negative log likelihood function in equation 3.22. We selectively fit every light curve that had at least ten data points on either side of trigger so that the τ_{fall} and τ_{rise} parameters were not biased by light curves that did not cover the full transient phase. The one-dimensional histograms of the best fit parameters are shown in Figure 3.2. We computed the mean $\boldsymbol{\mu}_{\text{pop}}$ and covariance $\boldsymbol{\Sigma}_{\text{pop}}$ of the set of best fit parameters and hence modelled the population distributions as multivariate Gaussians for each passband and each transient class. We decided to use this multivariate Gaussian as the prior distribution,

$$\mathcal{P}(\boldsymbol{\theta}) = \mathcal{N}\left(\boldsymbol{\theta}|\boldsymbol{\mu}_{\text{pop}}, \boldsymbol{\Sigma}_{\text{pop}}\right). \tag{3.24}$$

The distribution of the parameters A and σ_{int} have distributions that appear right-skewed. To make these distributions more Gaussian, we instead optimised over $\log_{10}(A)$ and $\log_{10}(\sigma_{\text{int}})$ and use these reparameterisations in the multivariate Gaussian prior.

3.4.2.3 Optimisation and fitting routine

To fit each light curve, we ideally want to first sample the posterior $\mathcal{P}(\boldsymbol{\theta}|\boldsymbol{D}_{sp}, \boldsymbol{t})$. Initially, we fit and sampled the posterior of the 6-parameter model with an MCMC routine. However, this proved far too computationally slow to run over the thousands of light curves in our training set. Instead, we used the Laplace approximation to approximate the posterior as a multivariate Gaussian centred on the mode. To apply this, we first optimised the objective function with the Nelder-Mead optimisation routine (available in the `scipy` optimisation library) after setting the starting parameter values to the median of the histograms shown in Figure 3.2. We then computed the Hessian matrix of the negative log posterior, using the autodifferentiation package `autograd`, and evaluated it at the optimal parameter values,

$$\mathrm{H}_{qr} = -\left.\frac{\partial^2 \log \mathcal{P}(\boldsymbol{\theta}|\boldsymbol{D}_{sp}, \boldsymbol{t})}{\partial\theta_q \partial\theta_r}\right|_{\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}} \tag{3.25}$$

where $\hat{\boldsymbol{\theta}}$ is the optimal parameter values, and q and r run across the six parameters in the model. The inverse of this Hessian is the covariance matrix,

$$\boldsymbol{\Sigma}_{\text{optimal}} = \mathrm{H}^{-1} \tag{3.26}$$

To ensure that the Laplace approximation was a good approximation of the posterior, we compare it with a fit using MCMC. We show the covariance contours from the Hessian matrix and the MCMC samples of an example SNIa g-band light curve fit in Appendix Figure 3.18. While the Hessian approximates most parameter covariances well, it is a poor approximation of $\log_{10}(\sigma_{\text{int}})$. The MCMC samples show that the model is non-Gaussian over $\log_{10}(\sigma_{\text{int}})$ and prefers negative values. Since the Hessian matrix is inherently symmetrical, it does not approximate this behaviour well. The large values of $\log_{10}(\sigma_{\text{int}})$ can lead to unphysically large estimations of the predicted flux. To account for this poor approximation, we define $\boldsymbol{\theta'} = [\log_{10}(A), B, t_0, \tau_{\text{fall}}, \tau_{\text{rise}}]$ and $\boldsymbol{\Sigma}'_{\text{optimal}}$ to not include $\log_{10}(\sigma_{\text{int}})$.

Then, to obtain posterior fits, we drew 100 samples from a multivariate Gaussian with mean $\hat{\boldsymbol{\theta}}'$ and covariance $\boldsymbol{\Sigma}'_{\text{optimal}}$ and set $\log_{10}(\sigma_{\text{int}})$ to the optimal fit for all draws. We evaluated equation 3.20 with these sets of parameters to obtain posterior fits. We noticed that some posterior samples in the Bazin parameter space produced unrealistic light curves that had wildly large flux values that did not fit past data well. These unrealistic parameter values are an artefact of the Laplace Approximation not being a good enough approximation to the true posterior (we discuss this in detail Appendix 3.8). To account for this behaviour, we ignored posterior fits that deviated from data previous to the present time T by $\chi^2 > 10$, leaving K posterior samples. We note that usually, less than 5% of the samples resulted in spurious fits, and so K was not much less than 100.

Unlike the DNN that performs regression over past data to give a flux predictions at a single time-step, our Bazin method performs regression over time to give flux predictions at all times. Therefore, to compare the predictive power of this method with the DNN, we use the interpolated fluxes as input, use data up to time T, and record the prediction 3 days later $(T+3)$. We do this for each time-step, to obtain a sequence of predictions one time-step in the future from given data.

In practice, for anomaly detection, we need to make predictions of D at new times T (where this T corresponds to the interpolated times used in the DNN for easy comparison).

This requires that we evaluate the predictive distribution defined by

$$\mathcal{P}(F_{sp(T+3)}|\boldsymbol{D}_{sp(t\leq T)}) = \int \mathcal{P}(F_{sp(T+3)}|\boldsymbol{\theta})\mathcal{P}(\boldsymbol{\theta}|\boldsymbol{D}_{sp(t\leq T)})d\boldsymbol{\theta}. \tag{3.27}$$

This distribution can be compared to the predictive distribution for the DNN described in equation 3.16. The integral on the RHS cannot be computed analytically, and so we approximate it by sampling. We draw K sample parameters $\boldsymbol{\theta}_{\text{draw}}$ of the posterior (second term in the integrand) and compute the flux predictions $F_{sp(T+3)}(\boldsymbol{\theta}_{\text{draw}})$ for each set of parameters (first term in the integrand) with equation 3.20. The LHS of equation 3.27 is approximated by the sampled probability distribution function, and we estimate the marginal predictive mean and uncertainty as the sample mean and standard deviation of the fluxes computed from the posterior draws, respectively:

$$y_{sp(T+3)} = \frac{1}{K}\sum_{\text{draw}=1}^{K} F_{sp(T+3)}(\boldsymbol{\theta}_{\text{draw}}) \tag{3.28}$$

$$\sigma_{y,sp(T+3)} = \sqrt{\frac{1}{K}\sum_{\text{draw}=1}^{K}\left(F_{sp(T+3)}(\boldsymbol{\theta}_{\text{draw}}) - y_{sp(T+3)}\right)^2}. \tag{3.29}$$

We use these to compute the anomaly scores discussed in Section 3.4.3. We also plot the $K \approx 100$ posterior fits (excluding the unrealistic spurious fits) and the median of these in the respective plots throughout this chapter.

3.4.3 Anomaly score definition

To quantify a potential anomaly, we first define the instantaneous anomaly score as a χ^2 metric to compute the discrepancy between the observed flux at time t and the predictions of a model based on previous data. This χ^2 is weighted by the total variance including the predictive uncertainty and measurement error.

$$\chi^2_{st} = \frac{1}{N_p}\sum_{p=1}^{N_p}\frac{(y_{spt} - D_{spt})^2}{c^2\sigma^2_{y,spt} + \sigma^2_{D,spt}}. \tag{3.30}$$

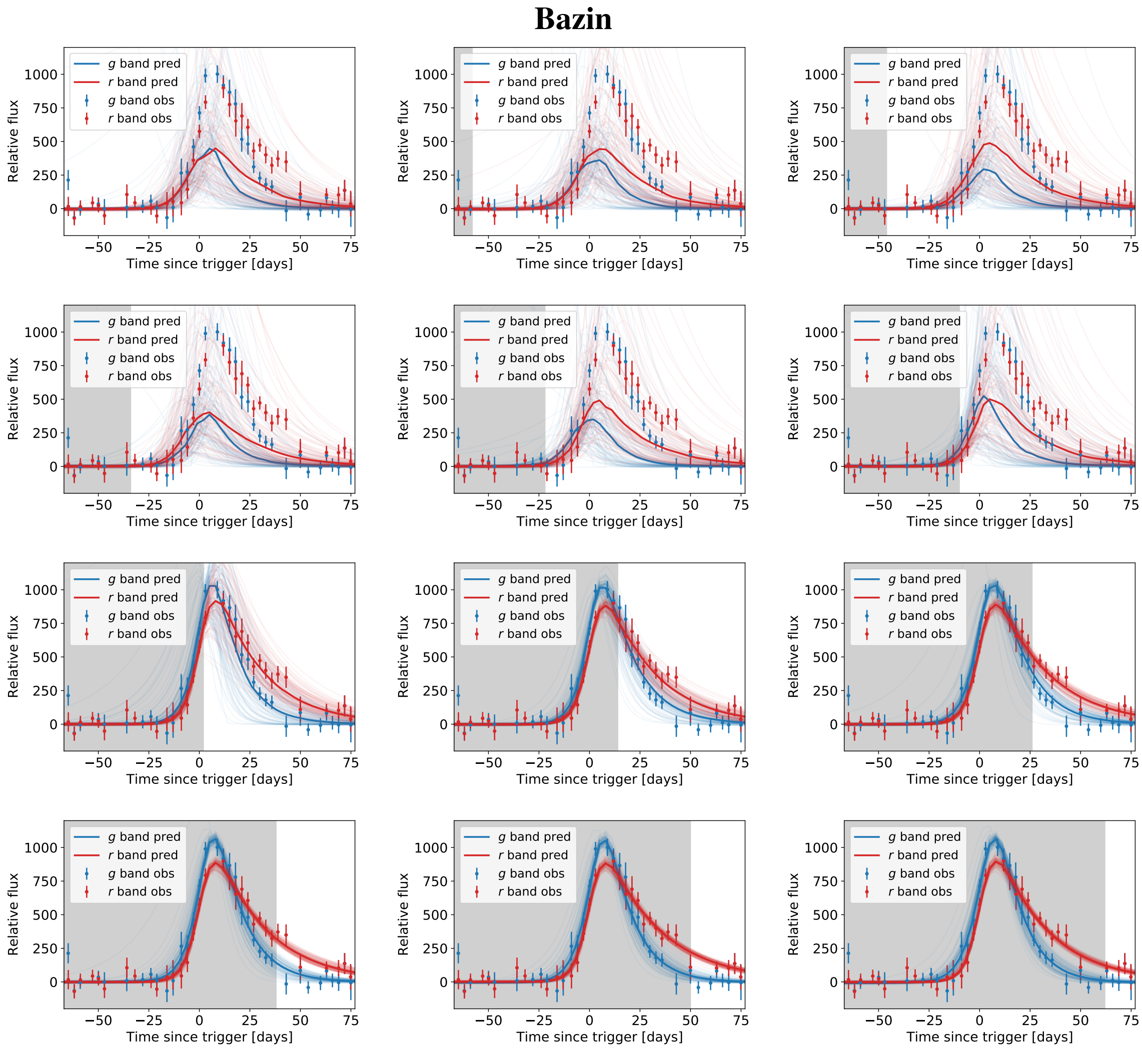


Fig. 3.4 The Bazin parametric method being used as a generative model of a SNIa given a partial light curve. The grey shaded region is the region of data that the model was fit with, while the observations in the white region was not used to fit the model. The trace lines show the normally distributed posterior sample fits from the Laplace approximation. The bold solid line is the median of the posterior fits. The first panel does not use any data from the light curve and thus illustrates the fits from the prior distribution. The following panels use data up to times -58, -46, -34, -22, -10, 2, 14, 26, 38, 50, and 62 days from trigger, respectively. The plots show a fit to an example simulated SNIa.

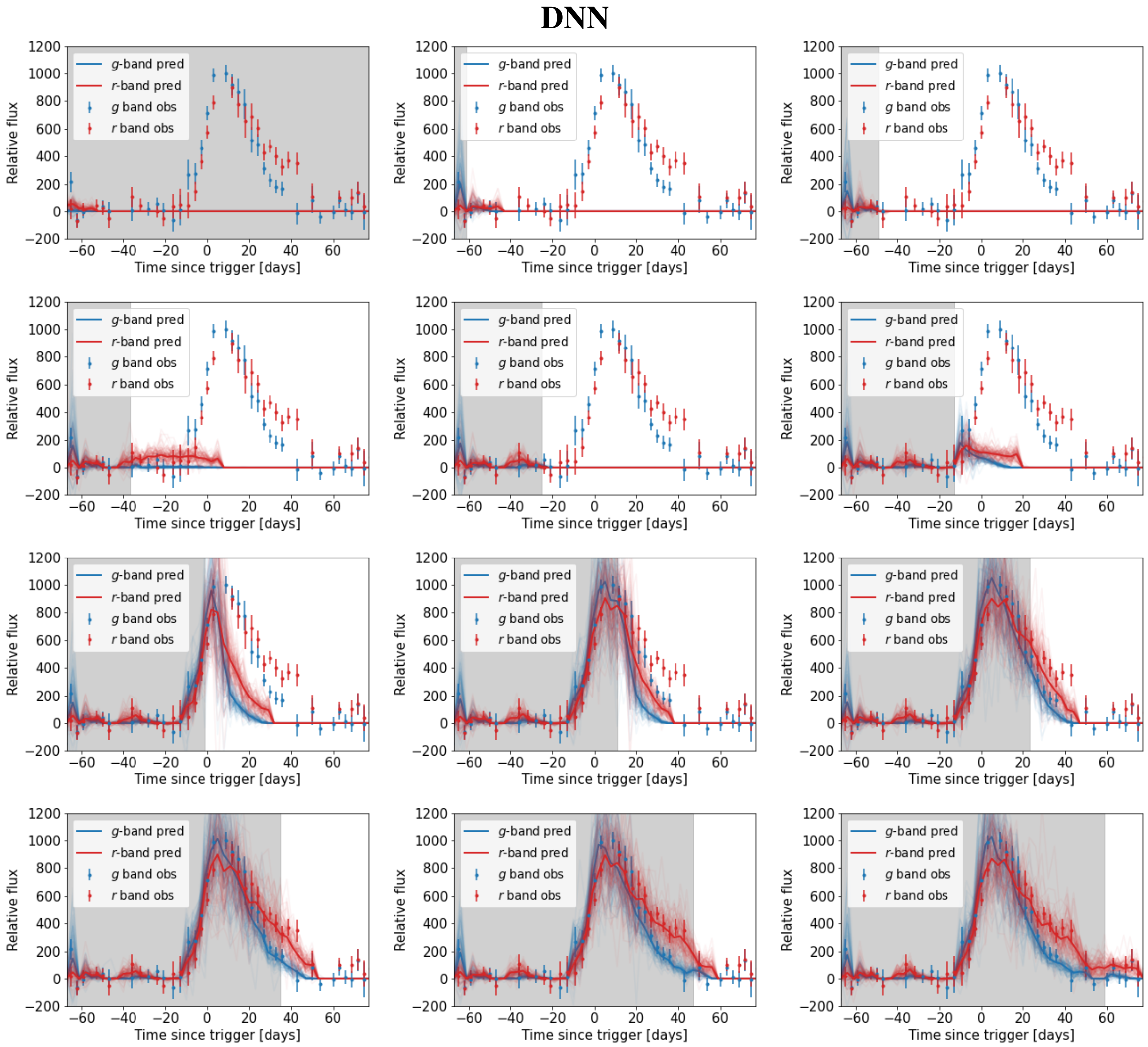


Fig. 3.5 The DNN being used as a generative model of a SNIa given a partial light curve. The grey shaded region is the region of data that was used to make a prediction, while the observations in the white region was not used to make predictions. As the DNN was specifically designed to only predict one time-step in the future and was not designed to generate an entire light curve, it cannot be expected to perform well, but acts as a good comparison to Figure 3.4. To obtain a sequence of predictions, we feed in the predicted values back into the DNN as if they were part of the observations. The trace lines illustrate the posterior sample predictions and the bold solid line is the median of the posterior predictions. The panels sequentially show the predicted light curve given increasing amounts of observational data, and each panel uses observations up to times -58, -46, -34, -22, -10, 2, 14, 26, 38, 50, and 62 days from trigger. The plots show prediction on an example simulated SNIa.

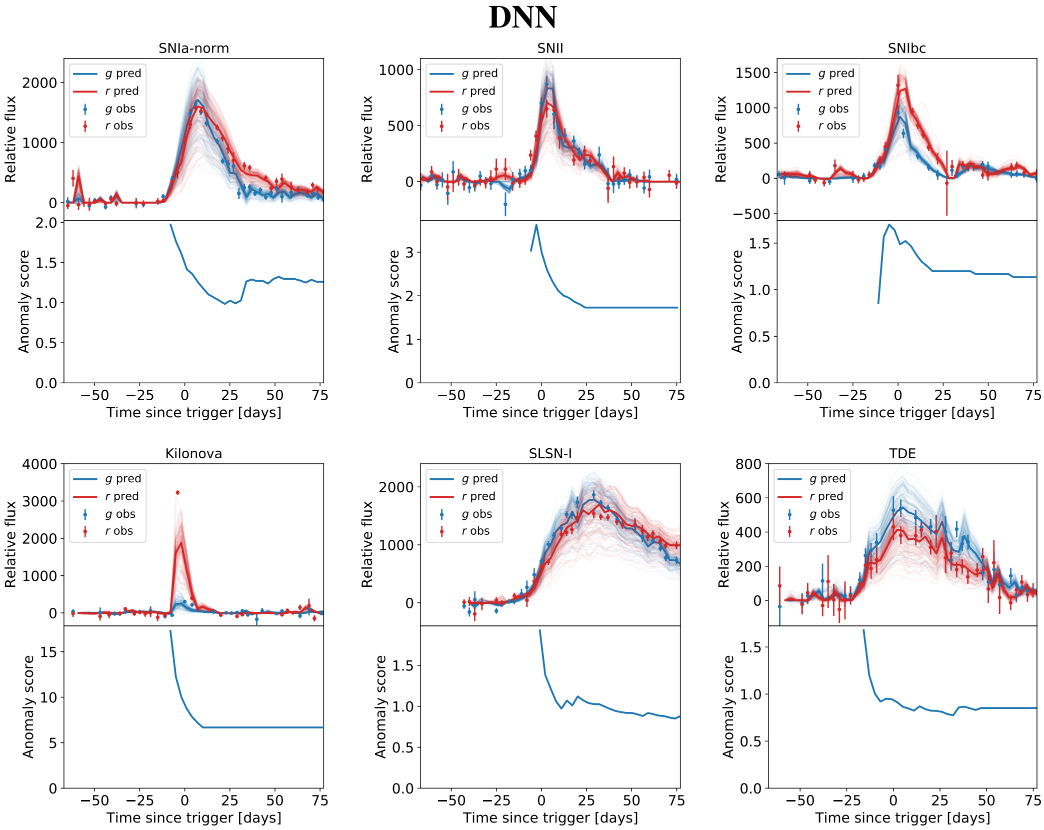


Fig. 3.6 The DNN being used as a predictive sequence model. Each plot uses a different one of the six trained models and applies it to an example transient from the same class. The trace lines show the posterior predictions and the bold solid line is the median of the posterior predictions. The SNIa plot (first plot) is made up of each of the predictions 3 days after the grey shaded regions in Figure 3.5. The bottom panels in each plot show the anomaly scores (computed using equation 3.31) as a function of time. We expect the anomaly scores to be low since these plots show example objects from the same class the models were trained on.

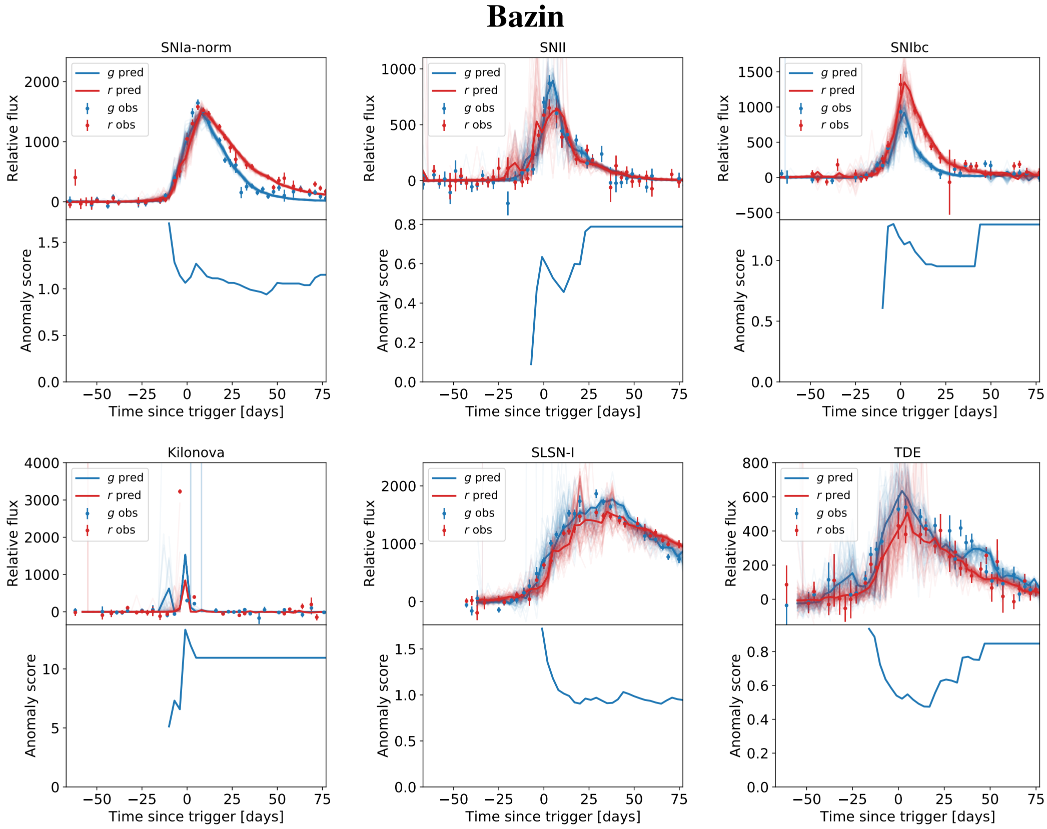


Fig. 3.7 A sequence of predictions only 3 days in the future of a given partial light curve made using the Bazin function. Each plot uses a different one of the six trained models and applies it to an example transient from the same class. The trace lines show the posterior predictions and the bold solid line is the median of the posterior predictions. The SNIa plot (first plot) is made up of each of the predictions 3 days after the grey shaded regions in Figure 3.4. The bottom panels in each plot show the anomaly scores (computed using equation 3.31) as a function of time. We expect the anomaly scores to be low since these plots show example objects from the same class the models were trained on.

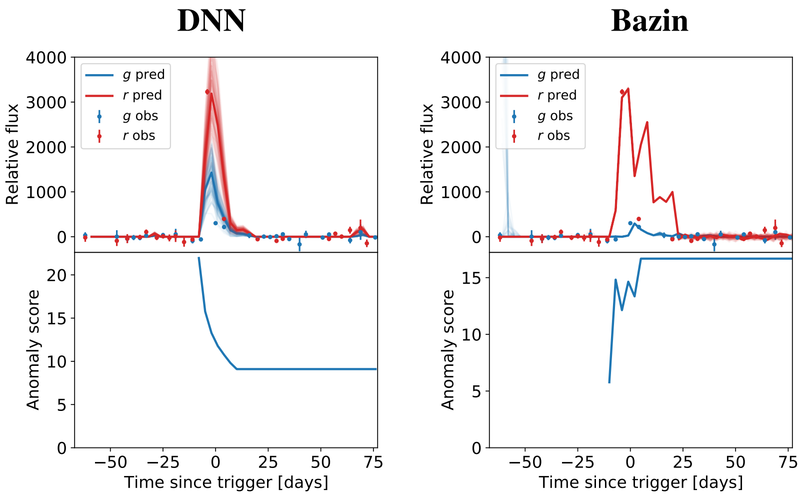


Fig. 3.8 Predictions of an example kilonova modelled with the DNN (left) and Bazin (right) SNIa models.The plots show a sequence of predictions only 3 days in the future of the partial light curves. The trace lines show the posterior predictions and the bold solid line is the median of the posterior predictions. The bottom panels in each plot show the anomaly scores (computed using equation 3.31) as a function of time.

Next, we define the *Anomaly score*, $\tilde{\chi}$, used throughout this chapter, as the square root of the time-averaged χ^2 up to the present time T,

$$\tilde{\chi}_{sT} = \sqrt{\frac{1}{N_{\text{avg}}} \sum_{\{t \leq T : (\text{S/N})_t > 5\}} \chi^2_{st}}, \qquad N_{\text{avg}} = |\{(\text{S/N})_{\text{sT}} > 5\}| \tag{3.31}$$

where N_{avg} is the number of time-steps with signal-to-noise greater than 5 up to the time T, and k runs across that index. This metric is effectively the time-averaged reduced χ^2 up to time T. After some analysis (see the detailed discussion in Appendix 3.9), we identified that the DNN overestimated the predictive uncertainty. To account for this, we scale the predictive uncertainty with a factor $c = 0.2$ for the DNN. We found that the Bazin model's predictive uncertainties were already well-calibrated to actual predictive performance and so use use $c = 1$ for the Bazin model.

This metric is used as our real-time anomaly score. Higher values indicate that the regressive model was less able to predict future data given past data, while lower scores indicate that the model was able to effectively predict future data.

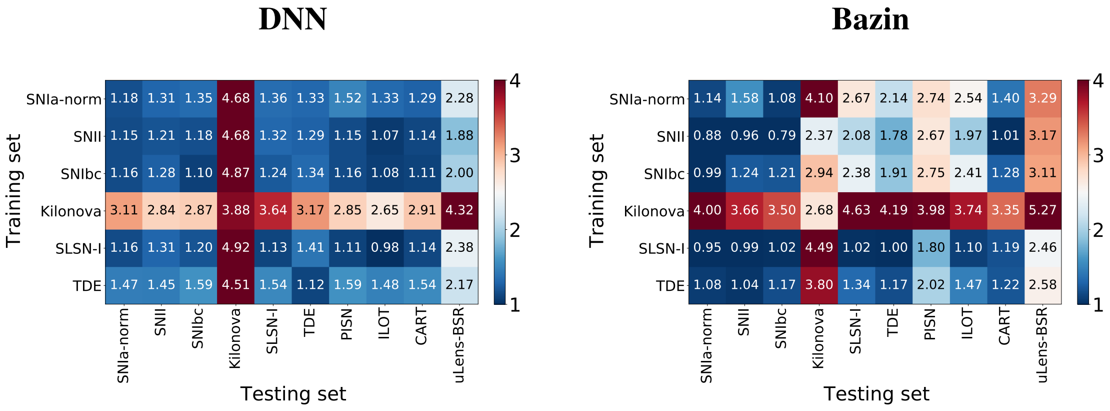


Fig. 3.9 The matrix illustrates the similarity of different transient classes, with lower numbers being more similar (less anomalous), and higher numbers being less similar (more anomalous). The vertical axis shows six trained models, and the horizontal axis are transients from a range of classes. Each transient in our dataset is fit with the six models, and the anomaly score over the full light curve is recorded. The median of the distribution of anomaly scores for each class are the numbers shown.

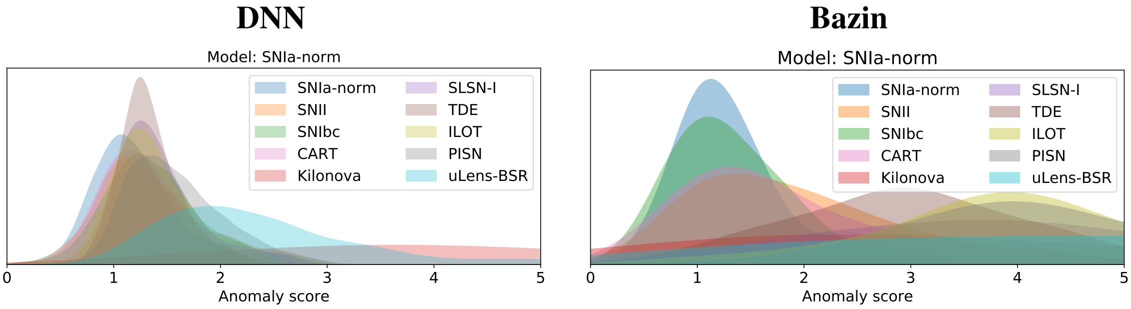


Fig. 3.10 Anomaly score distribution recorded over the full light curve for the SNIa model tested on the transient population of ten different classes. For clarity, we have smoothed the histograms with a Kernel Density Estimate (KDE) fit. Classes that are dissimilar to SNIa have higher anomaly scores, while similar classes have lower anomaly score distributions. The Bazin plot (right) shows a larger separation of the distributions of the SNIa and anomalous classes than the DNN (right). Similar plots can be made of the other five trained models, but are not shown for brevity.

DNN

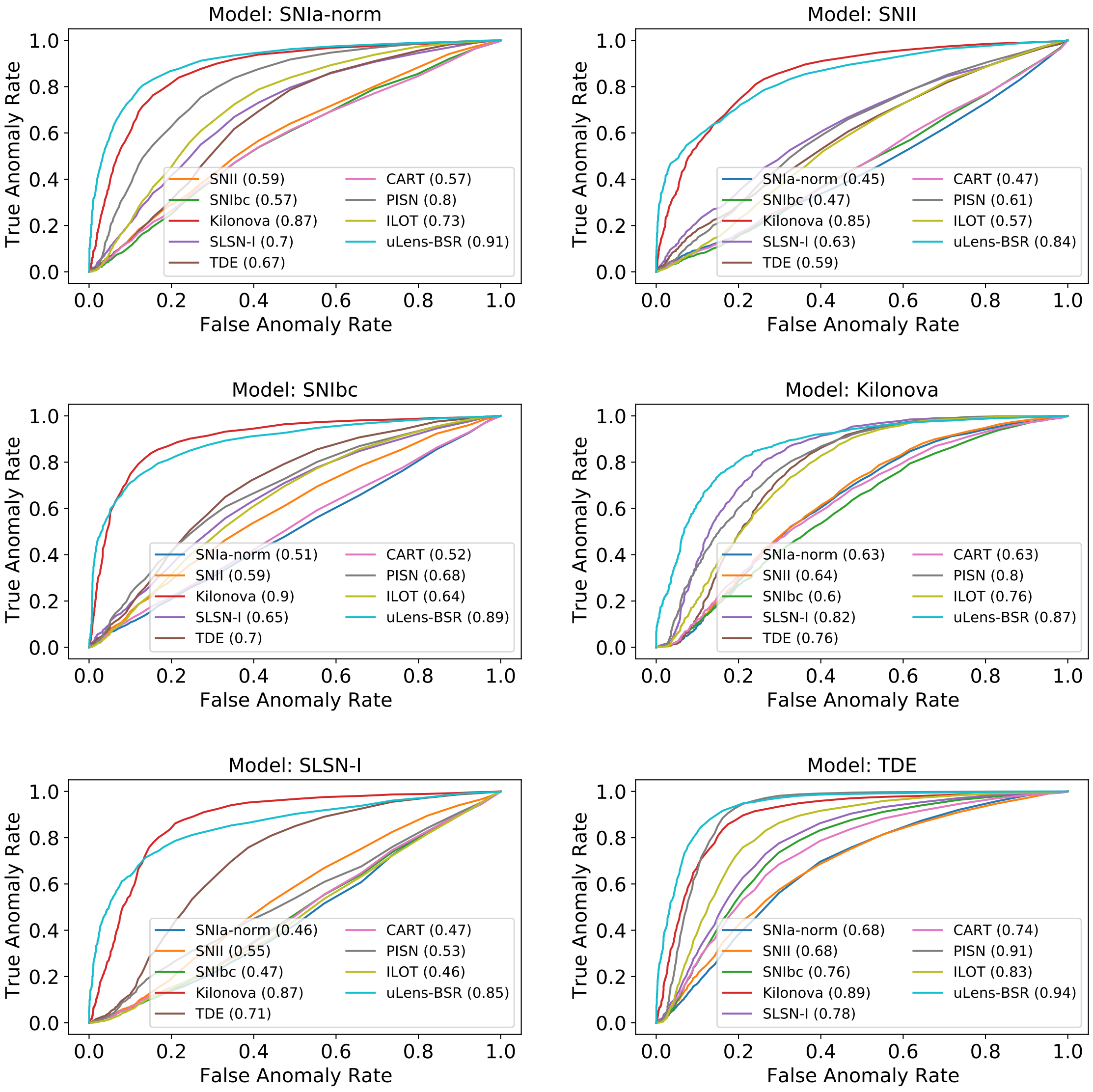


Fig. 3.11 Receiver Operating Characteristic (ROC) curve for each trained DNN model against nine other transient classes. In each subfigure, we use the Model class as the reference class and the anomalous classes as the ones denoted in the legend. The Area under the curves (AUCs) are shown in the brackets in the legends. The plots are made by plotting the True Anomaly Rate against the False Anomaly Rate for a range of different threshold anomaly scores. We use the anomaly scores over the full light curves to make these plots.

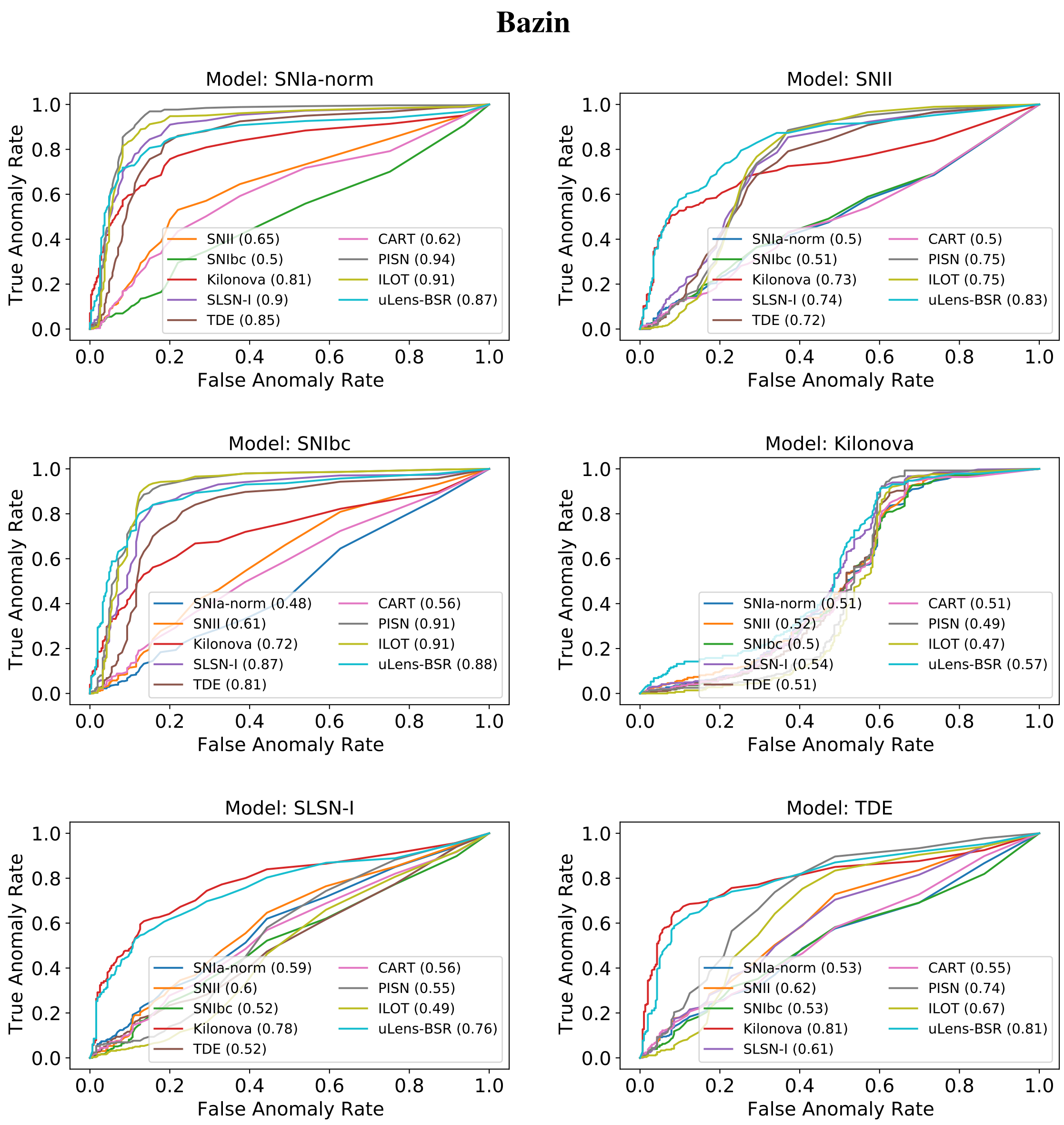


Fig. 3.12 Receiver Operating Characteristic (ROC) curve for each Bazin model against nine other transient classes. In each subfigure, we use the Model class as the reference class and the anomalous classes as the ones denoted in the legend. The Area under the curves (AUCs) are shown in the brackets in the legends. The plots are made by plotting the True Anomaly Rate against the False Anomaly Rate for a range of different threshold anomaly scores. We use the anomaly scores over the full light curves to make these plots.

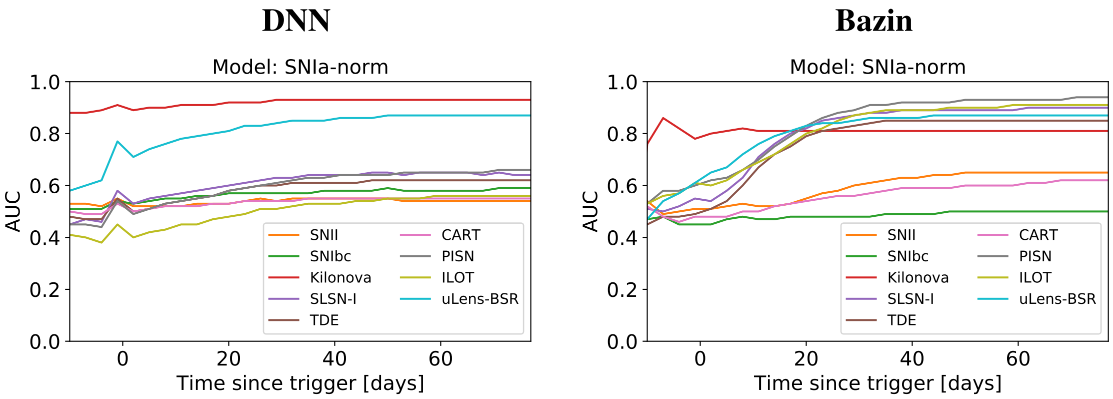


Fig. 3.13 Area Under the Curve (AUC) vs time since trigger assuming the SNIa model as the reference class and the anomalous classes denoted in the legend. These are made by reproducing the SNIa-norm ROC curves in Figures 3.11 and 3.12 at all time steps since trigger (instead of only over the full light curve) and recording the AUCs.

3.5 Results

In this section, we explore the performance of the DNN and the Bazin parametric approach at predicting light curves and identifying anomalies on a simulated ZTF dataset. For the DNN, we trained six autoregressive models one for each transient class: SNIa, SNII, SNIbc, Kilonova, SLSN, TDE. And similarly, for the Bazin function, we defined a prior distribution for each class. Each training set consisted of ∼8000 light curves and we tested the performance of the models on ∼2000 light curves from each transient class.

3.5.1 Generating light curves

The DNN was designed to predict one time-step in the future given a light curve up to a specified time. On the other hand, the Bazin model fits an entire past light curve and can make predictions for any set of future times. However, the primary purpose of this chapter is to predict just one time-step in the future to compare that prediction to observed fluxes (and hence evaluate an anomaly score). Before delving into the anomaly detection results, we first show the power of these two methods at building a generative model of a transient class.

In Figure 3.4, we illustrate the use of our Bazin parametric approach as a generative model of an example SNIa. Each panel fits only a partial light curve (shown in the grey region) and generates the rest of the light curve from this information. In the first panel, where no observations are being used, we are effectively plotting the prior distribution. As more observations are included in the fit, predictions improve, and once the peak of the light curve has been observed the predictions are much more accurate.

In Figure 3.5, we show the power of our DNN as a generative model of a full light curve to compare it against the parametric approach. Of course, we note that an autoencoder (specifically designed to fit an entire light curve) would perform much better at this task, it is interesting to see how the network's predictions evolve over time. As the DNN was specifically designed to only predict one time-step (3 days) in the future and was not designed to generate an entire light curve, we can only obtain a sequence of predictions by feeding in the predicted values back into the DNN and iteratively predicting each consecutive time-step in the light curve. While the predictions in the first few time-steps are accurate, the small inaccuracies quickly compound before the predictions reduce down to the zero-flux background prediction.

These two plots (Figures 3.4 and 3.5) illustrate how the two models differ in their approach to the problem - one being a fitting function regressing fluxes over time and the other being a predicting algorithm regressing future fluxes over past data. The Bazin model which was designed as a generative model of a light curve, obviously produces much more realistic light curves than the DNN. Forcing the DNN to generate a full light curve by iteratively inputting predicted values back into the model produces poor predictions.

In the rest of the plots in this chapter, we only use the predictions one time-step (3 days) in the future of a partial light curve. We emphasise that a key difference between Figures 3.4 and 3.5 and Figures 3.7 and 3.6, is that the former two figures generate an entire light curve given some partial light curve, while the latter figures iteratively predict only the next time-step given a partial light curve. Hence, to make the first panels of Figures 3.7 and 3.6, we use all the subplots of Figures 3.4 and 3.5, respectively, by recording the predictions one time-step after each panel's grey shaded region.

3.5.2 Fit many models

Following the methods outlined in Section 3.4, we trained a separate DNN and defined a separate Bazin prior for each of the six transient classes. In Figures 3.6 and 3.7, we illustrate the performance of these six models on example light curves in the testing sets of each class. The top panel of each subfigure shows the example light curve with uncertainties and posterior draws of the predictions, with the bold line showing the median of these draws. Each prediction is causal and hence only uses data from the previous time-steps. The bottom panel of each subfigure plots the anomaly score defined in equation 3.31. In most figures the final anomaly score is close to $\chi = 1$ indicating that the models are effective at predicting future transients fluxes from their own class.

The Kilonova models (for both the DNN and Bazin), however, show large anomaly scores, hinting that these are not good autoregressive models. Given the cadence of ZTF and the time-scale of kilonova events, most light curves only have one or two significant flux detections with a $\mathrm{S/N} > 5$. Thus, at this cadence, it is not surprising that our Bazin model and DNN, which are both causal models (that only use data prior to a given time), struggle to learn the shape of kilonova light curves.

The DNN and Bazin model's different approach to the regression problem causes slight differences in the anomaly score plots. The Bazin model regresses fluxes over time to learn the shape of each light curve as a function of time since trigger; while the DNN, on the other hand, regresses future fluxes over past data in a light curve and does not learn anything about time since trigger. This causes the DNN plots to have larger anomaly scores near the explosion time of each transient because the DNN's expectation of observing background flux is abruptly disrupted by the transient phase of an event. Furthermore, the Bazin model tends to have prediction light curves that are less smooth than the DNN, with some posterior samples having large deviations from the mean. This behaviour is because the Bazin model plots are produced from several independent fits to partial light curves.

3.5.3 Identifying anomalies

In Figure 3.8, we illustrate an example simulated kilonova with observations predicted using the SNIa models. The poor predictions and high anomaly scores indicate that this transient is flagged as anomalous with respect to the SNIa model - showing a first-order success in our method. We note that most kilonovae in the dataset were similarly flagged as highly anomalous at a similar epoch. As illustrated in Figure 3.2, the distribution of the τ_{fall} and τ_{rise} Bazin parameters of kilonovae is much lower than the mean of the SNIa prior distribution. Thus, the SNIa Bazin model struggles to model kilonova light curves, and in Figure 3.8 it s dominated by this prior whereby the predicted light curves have a much larger τ_{fall} than the data suggests, which causes a very high anomaly score. The DNN model appears to be more flexible, and is better at predicting the example kilonova light curve, but still has a very high anomaly score.

To compare the anomaly scores of all transients in our data set against our trained models, we have modelled every transient in the testing sets with each of the regressive models. Since we obtain anomaly scores as a function of time, we record the anomaly score of each transient over a full light curve, and report the median of all scores in Figure 3.9.

The plot highlights the similarity of each trained class to every other class, and acts as a similarity matrix for the shown transient classes. Higher numbers indicate classes that are more dissimilar, and lower number indicate classes that are more similar. For each trained model, the lowest number generally corresponds to the same class, which confirms effective training of our models, showing that each model can predict the future fluxes of transient light curves from its own class well. The model trained on SNe Ia also has low scores for Core-collapse SNe (SNIbc, SNII) and CARTs which highlights their similarity to SNe Ia. Kilonovae stand out as very anomalous for every trained model, indicating that the short lifetime and low luminosity of these classes cannot be well-predicted with the trained models. The Bazin matrix shows starker differences between the model class and the other testing classes, hinting that it may be better at identifying anomalies than the DNN. Overall, Figure 3.9 highlights some interesting similarities between transient classes, and confirms what may already be known about their general behaviour. It highlights the overall performance of our method on the testing sets, and shows that we are able to identify anomalous classes with this method.

However, Figure 3.9 only represents the median of the anomaly scores across the testing sets. In Figure 3.10, we plot the histograms of the full light curve anomaly scores of the SNIa model predicting the light curves from ten different testing classes. The plot shows that the DNN SNIa model cannot easily differentiate classes, other than the Kilonova and uLens-BSR classes. The Bazin plot, on the other hand, shows that the anomalous classes (Kilonova, SLSN, TDE, ILOT, PISN and uLens-BSR) all have histograms that separate well from the SNIa class, while the classes that are known to look similar to SNIa, (SNII, SNIbc, and CARTs) do not separate as well, consistent with expectations.

We refer to anomalies as all classes that are not from the reference class. To identify anomalies, a threshold anomaly score would need to be chosen such that the reference class was not flagged as anomalous but all other transient classes were flagged as anomalous. This threshold score would need to be chosen to have the highest possible True Anomaly Rate (TAR) while having the lowest possible False Anomaly Rate (FAR). That is, we would choose a threshold anomaly score that correctly identified most of the transients from non-reference classes as anomalous, while not identifying many of the transients from the reference class

	Actual Anomaly	Actual Not Anomaly
Predict Anomaly	TA	FA
Predict Not Anomaly	FNA	TNA

Table 3.1 Definitions of True Anomalies (TA), False Anomalies (FA), False Not Anomalies (FNA), and True Not Anomalies (TNA).

as anomalous. A Receiver Operating Characteristic (ROC) Curve that plots the TAR against the FAR for a range of thresholds is a good way to visualise this trade-off.

For the purposes of explanation in this paragraph, we use a SNIa as an example of a reference class transient and a kilonova as an example of a non-reference class transient. We define a True Anomaly (TA) as a transient from a non-reference class that was predicted as being an anomaly (e.g. correctly identifying a kilonova as anomalous), a False Anomaly (FA) as a transient from the reference class that was predicted as being an anomaly (e.g. incorrectly identifying a SNIa as anomalous), a True Not Anomaly (TNA) as a transient from the reference class that was predicted as being non-anomalous (e.g. correctly identifying a SNIa as non-anomalous), and a False Not Anomaly (FNA) as a transient from the non-reference class that was predicted as being non-anomalous (e.g. incorrectly identifying a kilonova as non-anomalous). These definitions are summarised in Table 3.1.

We then define the TAR as the ratio of the number of True Anomalies to all anomalies, and the FAR as the ratio of the number of False Anomalies to all non-anomalous transients, as follows,

$$\mathrm{TAR} = \frac{\mathrm{TA}}{\mathrm{TA}+\mathrm{FNA}}, \tag{3.32}$$

$$\mathrm{FAR} = \frac{\mathrm{FA}}{\mathrm{FA}+\mathrm{TNA}}. \tag{3.33}$$

We plot the ROC curves for each of the six trained models compared to each other class in Figures 3.11 and 3.12. The first subfigure of Figure 3.12 illustrates that the Bazin model of a SNIa is very effective at identifying all anomalous classes assuming SNIa as the reference class (except for core-collapse SNe and CARTs which are known to look broadly similar to SNe Ia) while having low FARs. The Area Under the Curves of these anomalous classes are above 0.8. Choosing a threshold score along these curves near the inflection on the top left point such that the FAR is low and the TAR is high will be a good choice to identifying anomalies with respect to SNe Ia. The performance of the Bazin SNIbc model is similar and the Bazin SNII model is only slightly worse, but similarly predicts most classes except for common SN types as anomalous. The other Bazin models are much poorer at identifying anomalies (assuming the model as the reference class), with the Kilonova model being nearly as bad as random guessing. Given that SNIa, SNII, and SNIbc are the most common types, developing an algorithm that identifies all classes that are not these common types would be an effective anomaly detection algorithm for most astronomers. The DNN models of the TDE and Kilonova classes are better than the Bazin models, but are worse than the Bazin models for the common SN classes.

The plots so far show that our method is able to identify anomalies relative to *common* SN classes when using full light curves. However, what is often more important for large scale surveys is identifying anomalies in real-time so that we can prioritise which transients should receive follow-up observations. We have made similar ROC curves to figures 3.11 and 3.12 for every time-step since trigger instead of over the full light curve. We summarise these for the SNIa models as a plot of the Area Under the Curve of each class as a function of time since trigger in Figure 3.13. We can see that the AUC increases with time since trigger and plateaus around 25 days since trigger, which is often close to the end of the transient phase of most SNe Ia. The Bazin model clearly performs much better than the DNN model of SNIa at identifying anomalies (except for kilonovae) at all times, where we assume SNe Ia as the reference class.

We have decisively shown that the Bazin models of the common SN classes are significantly better at identifying anomalies than the DNN models. In Appendix 3.10.2, we highlight that the poor performance of the DNN compared to the Bazin model is because it is too flexible at predicting light curves; and after being trained on one class, it is still able to accurately predict fluxes in a different class of transients. The DNN model is actually better at predicting the future fluxes of transients within a trained class, but is also able to predict the future fluxes of transients from different classes well. While this flexibility allows for good flux predictions, it is not good for anomaly detection. Future work should look at developing a better DNN model that penalises flux predictions from anomalous transients while rewarding flux predictions from the trained class. The remaining plots in this Section and Section 3.16 use the Bazin framework instead of the DNN.

3.5.4 Identifying anomalies against common classes

With this many models of each transient, the question of which transient model should ideally be used to identify anomalies remains. We have developed a framework for identifying anomalies with respect to particular model classes. However, often astronomers are interested

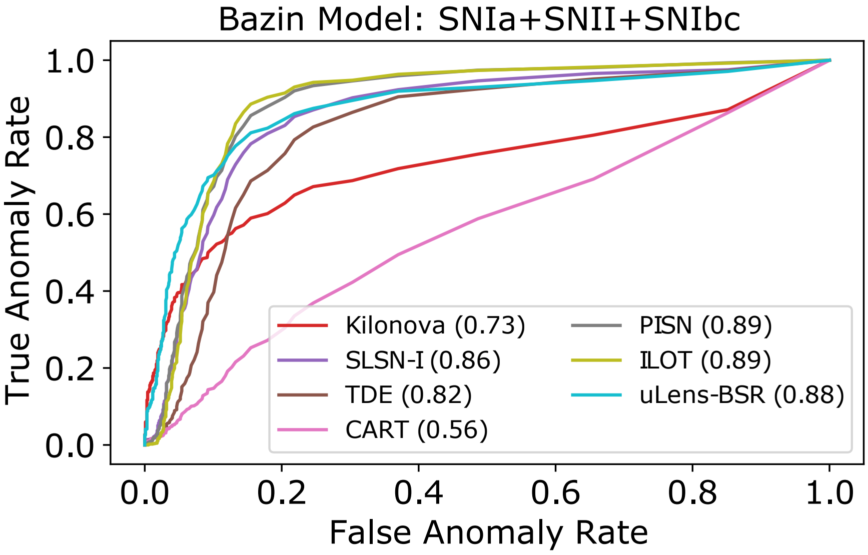


Fig. 3.14 ROC curves assuming the combination of the Bazin SNIa, SNII and SNIbc models are the reference classes and the anomalous classes are denoted in the legend. The area under the curves (AUCs) are shown in brackets in the legend. We use the anomaly scores over the full light curves to make these ROC curves.

in identifying anomalies with respect to common classes. For this case, we suggest a simple addition of the anomaly scores from the models that would be considered "common". For example, a good choice, might be to define the total anomaly score as the mean of the common SNIa, SNII, and SNIbc classes, as follows,

$$\chi_{\text{total}} = \frac{1}{3}\left(\chi_{\text{SNIa}} + \chi_{\text{SNII}} + \chi_{\text{SNIbc}}\right). \tag{3.34}$$

This enables us to identify anomalies with respect to all three of these common classes.

We combine the results of the Bazin models of the SNIa, SNII, and SNIbc classes and plot the resultant ROC curve in Figure 3.14 and the AUC vs time in Figure 3.15. The combined model performs very well at distinguishing all classes except for CARTs with high AUCs. As noted in Muthukrishna et al. (2019a), CARTs are difficult to distinguish from common SNe based only on the light curves.

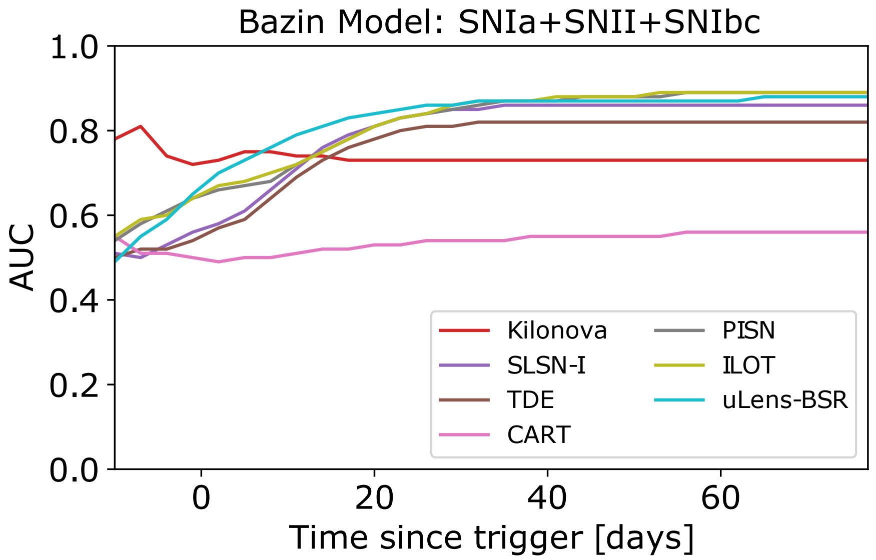


Fig. 3.15 The area under the curve (AUC) vs time since trigger assuming the combination of the Bazin SNIa, SNII and SNIbc models are the reference classes and the anomalous classes are denoted in the legend. These are made by reproducing the ROC curve in Figure 3.14 at all time steps since trigger and recording the AUCs.

3.6 Application to ZTF Observational data

In this section we illustrate our method being applied to real observations from the public ZTF MSIP survey instead of simulations.

The vast majority of observed transient phenomena are Type Ia Supernovae (SNe Ia), and it is therefore difficult to build a training set for other classes. Nonetheless, to obtain a labelled dataset of ZTF transients, we searched the Open Supernova Catalog[4] (Guillochon et al., 2017) for objects with ZTF aliases. We collected 11278 SNe with ZTF aliases, but after removing all unclassified events (i.e. objects classified as "Candidate", "Other", "removed", or "NT", comprising 8413 objects), and objects with invalid or multiple conflicting classification labels (comprising 26 objects), we were left with 2839 labeled SNe. The number of events under each classification label were as follows: Ia: 1993, Ia-91T: 78, Ia-91bg: 14, Ia-pec: 21, Ia-csm: 3, Ia-x: 10, II: 351, IIP: 56, IIL: 1, II-pec: 2, IIn: 87, IIb: 37, Ib: 40, Ibn: 6, Ic: 55, Ic-BL: 31, Ibc: 9, CC: 6, SLSN: 1, SLSN-I: 23, SLSN-II: 16.

[4] We searched the catalog, https://sne.space/, on 22 Oct 2020

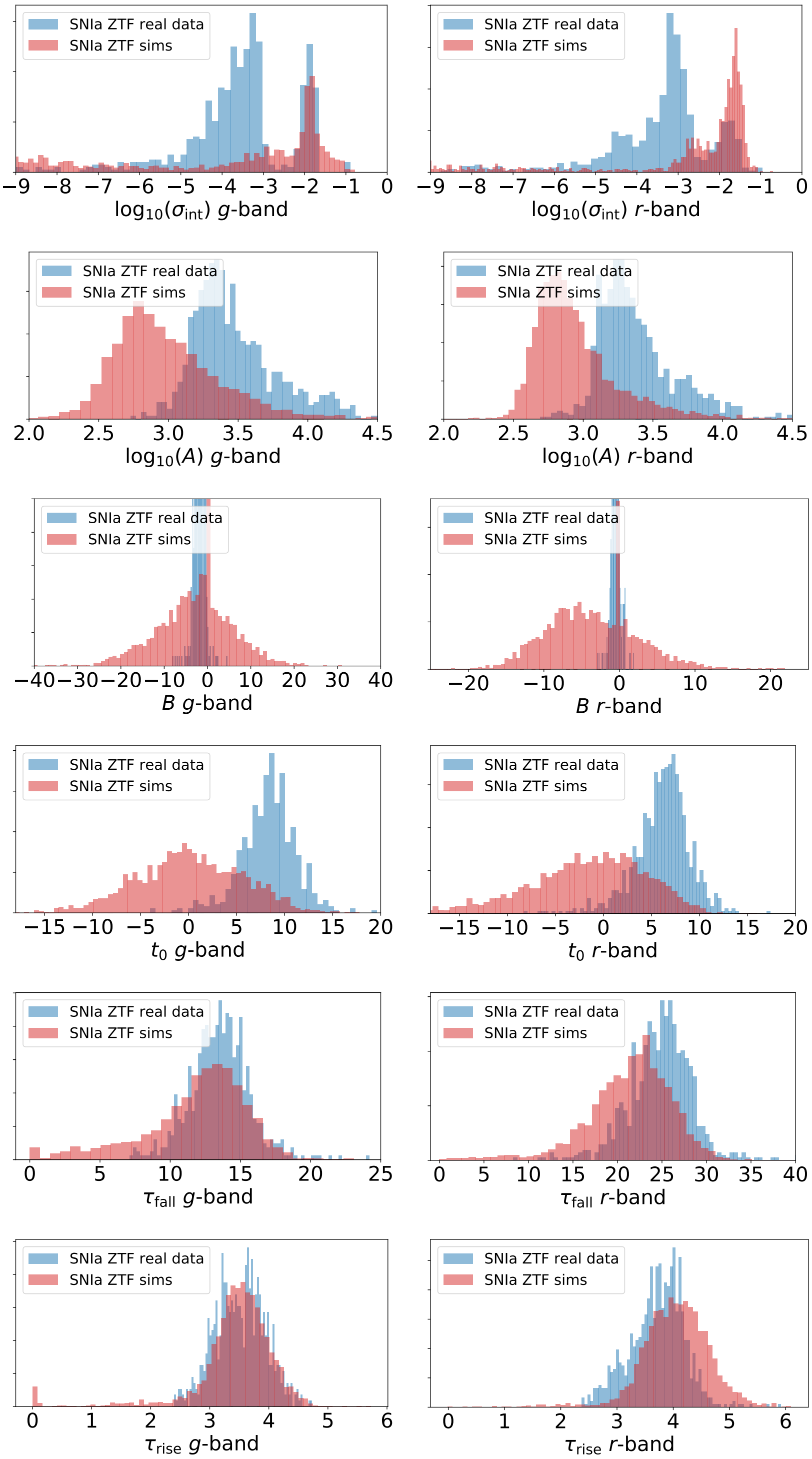


Fig. 3.16 Comparison of the distributions of the best fit Bazin parameters for the population of simulated ZTF light curves and the collection of real observations taken from the ZTF data stream. We only show the parameter fits of light curves that have at least nine observations and at least one observation before peak.

Unlike the simulations, the real data from the ZTF MSIP data stream are distributed in alert packets that contain magnitudes instead of flux units. We convert these magnitudes and magnitude uncertainties to flux counts and uncertainties as follows,

$$F = 10^{-0.4(\mathrm{mag}-26.2)} \tag{3.35}$$

$$\sigma_F = |F\sigma_{\mathrm{mag}} \times 0.4 \log 10| \tag{3.36}$$

where F is the flux, mag is the magnitude in the ZTF alert packet, σ_F is the uncertainty in the flux, and σ_{mag} is the magnitude uncertainty. We have selected a zeropoint of 26.2 to scale the observations such that the flux and flux uncertainty distributions match the simulations.

We performed the same processing methods detailed in Sections 3.3 and 3.4. To compare our simulations to the real observations, we plot the Bazin parameter distributions when fit to the SNIa light curves from our simulations and our collected ZTF transient populations in Figure 3.16. The SNIa simulation distributions are the same as that shown in Figure 3.2, and the real data distributions were made by optimising the likelihood for all real SNIa light curves that had at least nine data points and at least one point before peak. The τ_{fall} and τ_{rise} distributions match the simulations reasonably well. However, there are a large fraction of real light curves that have sparse data that are not well observed well before peak (to constrain τ_{rise}) and well after peak (to constrain τ_{fall}), and hence cause τ_{fall} to be slightly overestimated for the real data. As in Section 3.4, we choose to use the population parameter distributions as the prior for the Bazin model. However, we have used the τ_{fall} and τ_{rise} prior from the simulations because the parameter histograms look similar and because the missing observations in the real light curves could lead to slight overestimations of these parameters. The real data also has much brighter peak fluxes as indicated by the larger values of $\log_{10}(A)$. This is most probably due to selection effects where brighter SNIa are more likely to have been found and classified. The distribution on t_0 is also slightly offset, mainly because the real data do not have any pre-trigger observations as there is no available forced photometry to get reliable non-detection fluxes.

In Figure 3.17, we illustrate the ROC curve of our method trained on real SNIa data and tested on the other classes we have collected, where we assume SNe Ia as the reference class. The SNII class includes all transients labelled on the OSC as II, IIP, or IIn; the SNIbc class includes all transients labelled as IIb, Ib, Ibn, Ic, or IcBL; and the SLSN class includes all transients labelled as SLSN-I or SLSN-II. The AUC for the SNII and SLSN classes assuming SNIa as the reference class are 0.84 and 0.87, respectively, indicating that our approach was very effective at identifying SNe II and SLSNe as anomalous. However, it was not

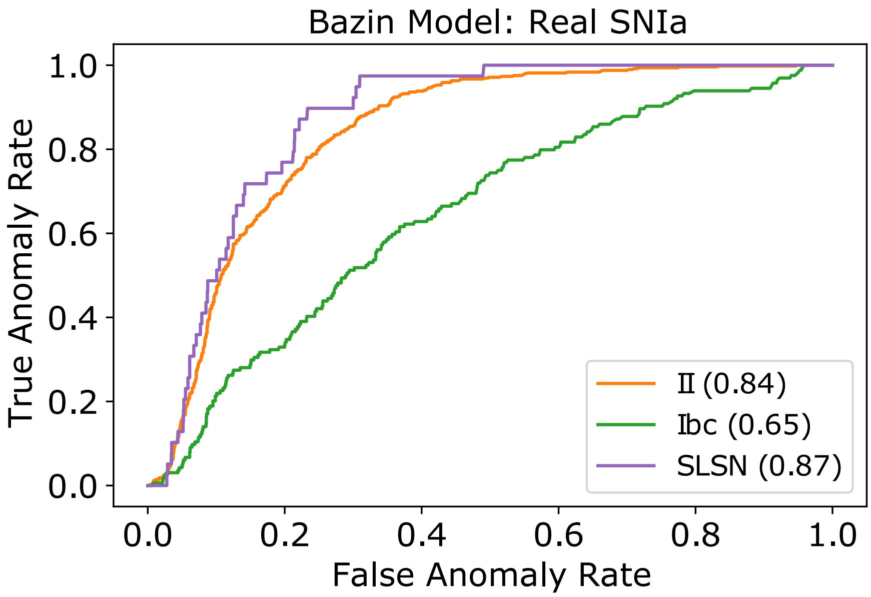


Fig. 3.17 Receiver Operating Characteristic (ROC) curve based on a Bazin model for real ZTF SNIa observations where we assume the SNIa class as the reference class and the anomalous classes are the ones denoted in the legend. We test the performance of this model on real observations of SNII, SNIbc, and SLSN classes. We use the anomaly scores over the full light curves to make the ROC curves.

as proficient at identifying SNe Ibc. This section has highlighted that our framework for identifying anomalies might be reasonably effectively when applied to a collection of real data without the use of simulations.

3.7 Conclusions

Upcoming wide-field surveys of the transient universe will probe deeper, wider, and faster than ever before, providing an opportunity for the discovery of entirely new classes of transient phenomena. However, discovery in astronomy has often been driven by serendipity, whereby identifying new phenomena has fortuitously occurred after human eyes sifted through data. With the huge amounts of data from surveys such as the LSST (expected to observe over 10 million transient alerts each night), a methodology aimed at automating the discovery of new transients through dedicated anomaly detection algorithms has become necessary.

Standard supervised classification approaches are unable to deal with the scope for new discovery offered by the wealth of data from upcoming surveys because they can only identify transients that they have been specifically trained on. Anomaly detection algorithms enable an opportunity to automatically flag unusual and interesting transients for further follow-up. In this chapter, we have detailed the development of a real-time anomaly detection framework for identifying unusual transients in large-scale transient surveys. We have built two separate frameworks. The first is a probabilistic deep neural network (DNN) built using Temporal Convolutional Networks aimed at predicting the next data point in a light curve. And the second is based on a parametric fit to a partial light curve using the Bazin function (Bazin et al., 2009), where we extrapolate a prediction 3 days after each partial light curve fit to compare it to the DNN approach. Each of the approaches can be well optimised to deal with the millions of alerts that ongoing and upcoming wide-field surveys such as ZTF and LSST will produce.

Our two methods allow us to identify anomalies as a function of time, and we have demonstrated its performances on both ZTF-like simulations and real ZTF light curves from the public MSIP survey. In particular, we have demonstrated that we are able to identify anomalies with respect to common supernova classes (SNIa, SNII, SNIbc) with low False Anomaly Rates and high True Anomaly rates culminating in Area Under the ROC Curve (AUC) scores well above 0.8 for most rare classes with the Bazin approach. Our ability to identify anomalies improves over the lifetime of the light curves. Based on the anomaly scores and the epoch of discovery, our framework enables a prioritised follow-up of unusual transients.

Both the DNN and Bazin approaches are very fast and will be easily scaleable to surveys as large as LSST, but the DNN is considerably faster at inference time. However, while we have shown that our DNN approach is very good at the prediction of fluxes, we have also noted that it is too flexible to act as a good anomaly detector when compared with our Bazin approach. The DNN method trained on a particular supernova class is able to accurately predict supernova within that class, but is so flexible, that it makes reasonable predictions of transients in other classes too. This flexibility means that it is not good at detecting anomalies. On the other hand, the Bazin approach is very effective at identifying transients outside the modeled supernova class, making it an effective anomaly detector.

In future work, we hope to apply our method on a ZTF transient broker to gauge our success at identifying real anomalous transients. We think that applying an anomaly detection framework in conjunction with a transient classifier will provide more valuable information on whether a newly discovered transient is interesting enough for further follow-up observations. An issue with this work, is that there has been no distinction between anomalies and *interesting* anomalies. It is possible that without good *real-bogus* cuts on a data stream, our approach may flag unusual transient phenomena that don't align with our trained supernova classes but are uninteresting to most astronomers. To deal with this, future work should apply Active Learning frameworks that use methods such as *Human-in-the-loop learning* that focus on specifically targeting what users define as interesting phenomena (recent work by Ishida et al. 2019a; Lochner & Bassett 2020 have begun working on Active Learning for anomaly detection).

Overall, this chapter presents a novel and effective method at identifying anomalous transients in real-time surveys. Anomaly detection coupled with other classification approaches enables astronomers to prioritise follow-up candidates. Building from this work and other recent approaches to anomaly detection is going to be hugely important for discovery in the new era of large-scale transient surveys.

3.8 Appendix: Laplace approximation

In Section 3.4.2, we detail how the light curves are fit using a Bayesian parametric function. We would ideally obtain distributions over the parameters for each light curve fit using

MCMC. However, because of the how computationally intensive this would be for the many light curves in our training set (and in large scale surveys), we resort to optimising the fit, and approximating the posterior with the Laplace approximation.

In Figure 3.18, we compare our fits to an example SNIa light curve using MCMC and our Laplace approximation. The parameter distributions appear well-approximated by the Laplace approximation for all parameters except for $\log_{10}(\sigma_{\rm int})$. The MCMC samples disfavour high values of $\log_{10}(\sigma_{\rm int})$, and have a sharp cut-off near zero. However, because the Hessian matrix is inherently symmetrical, it does not approximate this distribution well. The very large values of $\log_{10}(\sigma_{\rm int})$ lead to unrealistically large estimates of the predicted flux. Thus, we use the mode of this parameter instead of sampling over it for all Bazin models in this chapter.

3.9 Appendix: Analysis of predictive uncertainty

We performed the following analysis to assess the computed predictive uncertainties. We first defined the Total-Uncertainty-Scaled Prediction Error as follows,

$$\mathrm{TUSPE}_{spt} = \frac{(y_{spt} - D_{spt})}{\sqrt{c^2\sigma^2_{y,spt} + \sigma^2_{D,spt}}}. \tag{3.37}$$

We note that the instantaneous anomaly score in equation 3.31 is just the average over passbands of the squared scaled error. We plotted the distribution of the scaled error for the SNIa model and recorded the mean and root mean square (rms) at each time-step. For an unbiased model, the mean of the scaled error should be close to 0, and for a model that correctly estimates the predictive uncertainty, the rms should be close to 1. We have plotted the scaled error as a function of time since trigger for the SNIa DNN and Bazin model in Figure 3.19. Around the early phase of the transient (before trigger) the rms and bias is large because the models are not effective until more of the light curve has been observed. In this work, we are mainly interested in observations after trigger, and so, we recognise that a good model would have a rms near 1 after trigger. While the rms is good for the Bazin model, it is too small for the DNN, indicating that the predictive uncertainty is overestimated. Our DNN poorly estimates the predictive uncertainty, and to calibrate it, we scale it by a factor of $c = 0.2$ to obtain the green plot in Figure 3.19. We made similar plots for the other five

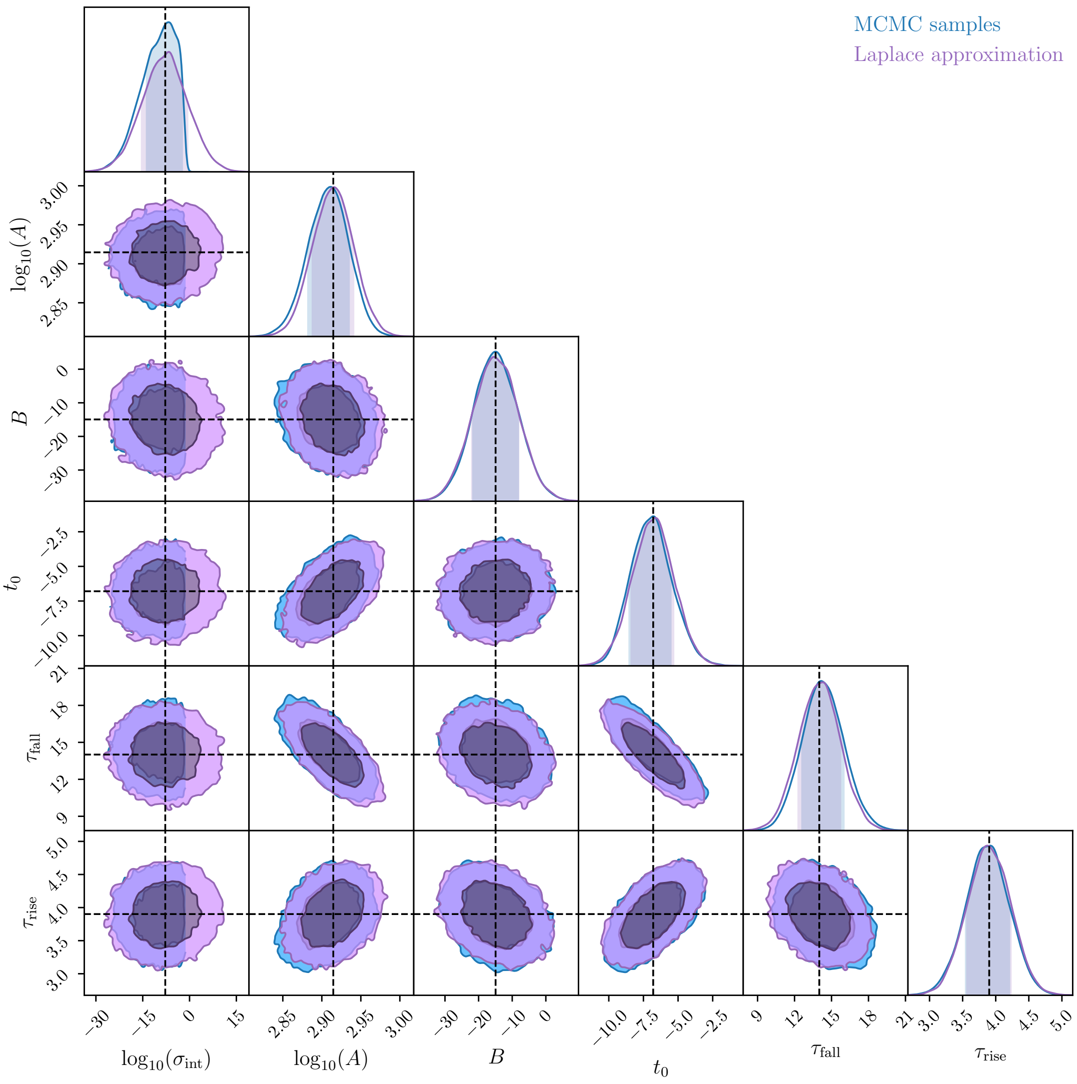


Fig. 3.18 Example Bazin fit parameter distributions of example SNIa light curve using MCMC samples and the Laplace approximation. The dashed lines shows the optimal fit from the Nelder-Mead optimisation routine.

transient models (not shown for brevity), and determined that the optimal factor to correct all the DNN models was close to $c = 0.2$.

3.10 Appendix: Comparison of DNN and Bazin predictive power

We performed the following analysis to evaluate why the DNN model was less effective at identifying anomalies than the Bazin model. To compare the models on an even scale, we defined the Measurement-Uncertainty-Scaled Prediction Error as follows,

$$\mathrm{MUSPE}_{spt} = \frac{(y_{spt} - D_{spt})}{\sigma_{D,spt}}. \tag{3.38}$$

We note that this differs from equation 3.37 because we are normalising the flux prediction error by the square-root of the measurement variance instead of the square-root of the total variance. The scaled error in equation 3.37 would not let us easily compare between the methods because the predictive variance differs for the DNN and Bazin model, and thus the denominators would be different for each model. Equation 3.38, on the other hand, is just the flux prediction error in units of the measurement error, and thus allows us to compare the DNN and Bazin models on an even scale.

3.10.1 DNN overfitting evaluation

In Figure 3.20, we plot the distribution of prediction errors on all light curves in the DNN SNIa training set (orange lines) and testing set (green lines). The prediction error distributions are very similar, and add further confirmation that our DNN SNIa model has not overfit the training set. We made similar plots and conclusions for the SNIbc, SNII, Kilonova, SLSN and TDE models, but have not shown them here for brevity.

The blue line in Figure 3.20 shows the prediction error distributions for the Bazin model, and it appears that they perform slightly worse than the DNN, with a slghtly wider prediction error distribution. Hence, we conclude that the DNN is slightly better at modelling SNIa than the Bazin model. However, this does not explain why the DNN is worse at identifying anomalies.

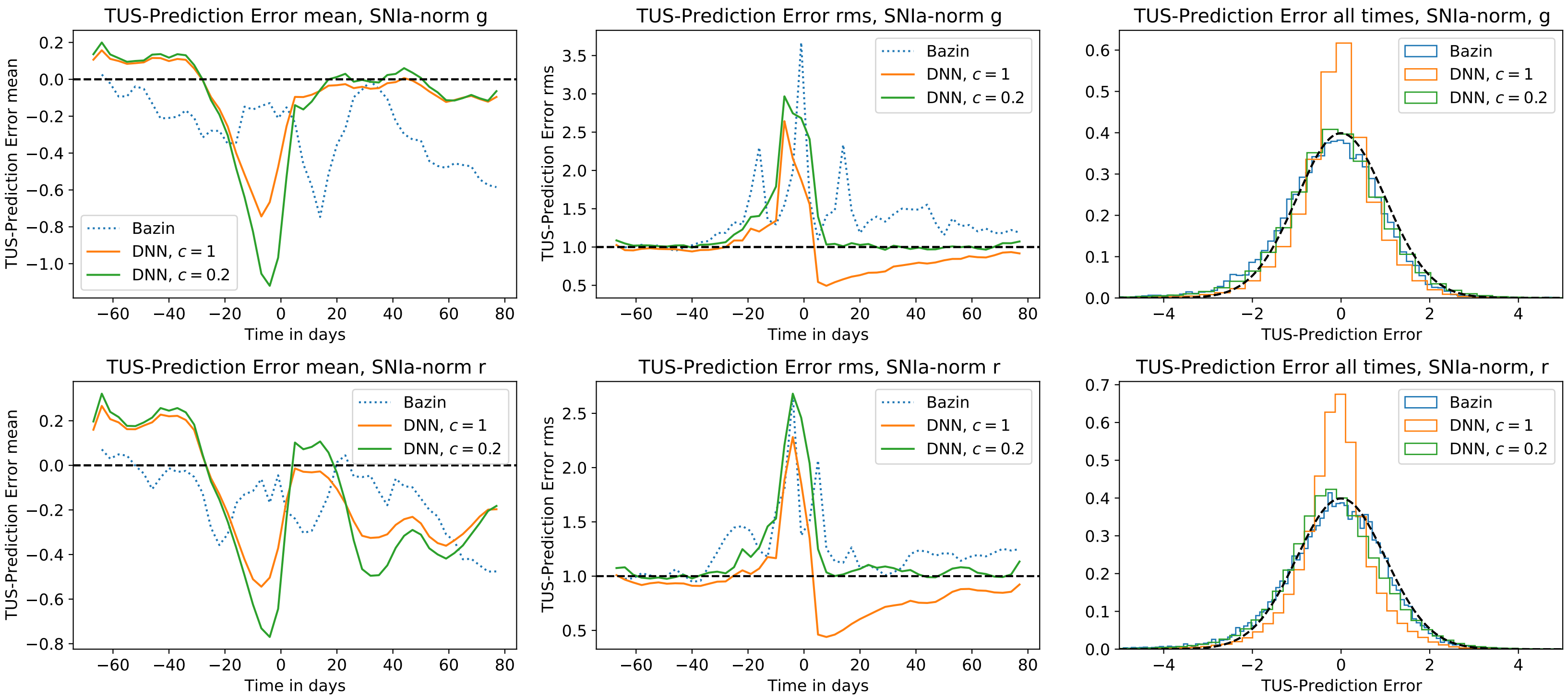


Fig. 3.19 Distribution of the Total-Uncertainty-Scaled Prediction Error (TUSPE) for the SNIa model at different times. Equation 3.37 is computed at each time-step for all SNIa in the testing set. We show the mean and root mean square (rms) for each scaled error distribution at each time-step in the first three panels, with the *g* band shown in the top row of panels, and the *r* band shown in the bottom row of panels. In the last column, we plot the distribution of scaled errors across all times. After trigger, a scaling factor of $c = 0.2$ on the predictive uncertainty for the DNN improves the scaled error. Similar plots were made for the SNIbc, SNII, Kilonova, SLSN and TDE models, but are not shown here for brevity. We plot a unit Gaussian (which is an ideal Scaled Error distribution) as a black dashed line to help guide the eye.

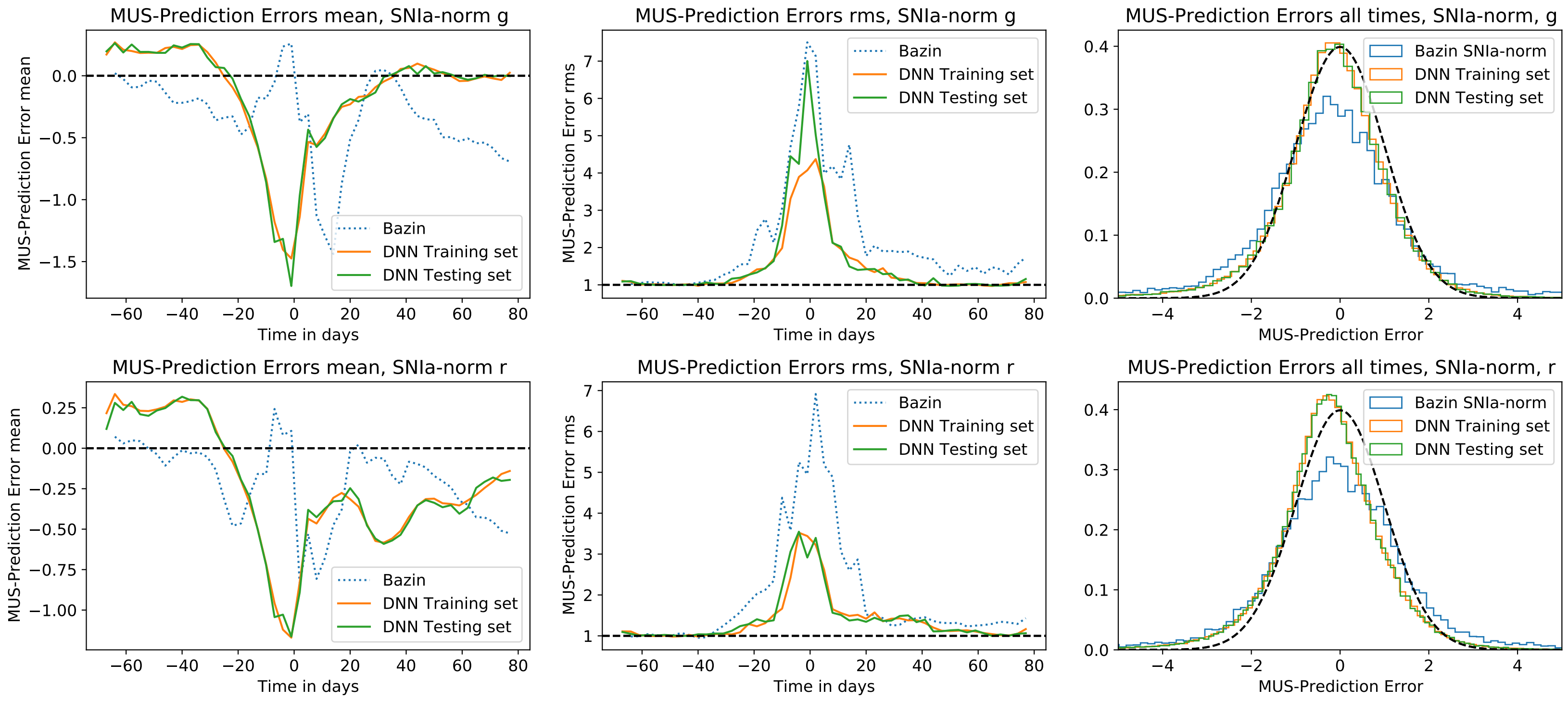


Fig. 3.20 Distribution of the Measurement-Uncertainty-Scaled Prediction Errors (MUSPE) for the SNIa model at different times. Equation 3.38 is computed at each time-step for all SNIa in DNN training set (orange lines), testing sets (green lines) and for the Bazin training set (blue dotted line). We show the mean, and root mean square (rms) for each prediction error distribution at each time-step in the first three panels, with the g band shown in the top row of panels, and the r band shown in the bottom row of panels. In the last column, we plot the distribution of scaled errors across all times. Similar plots were made for the SNIbc, SNII, Kilonova, SLSN and TDE models, but are not shown here for brevity. We plot a unit Gaussian as a black dashed line to help guide the eye.

3.10.2 DNN vs Bazin

To analyse why the SNIa Bazin model is better than than the DNN at identifying anomalies, we plot the prediction errors for the SNIa models applied to each of the other transient classes for Bazin and DNN in Figures 3.21 and 3.22, respectively. We expect that the SNIa models should predict the SNIa light curves best, and indeed, we see that these blue lines for the SNIa have nearly the best prediction error distributions. In Figure 3.21, the prediction errors are significantly worse for the more anomalous classes (SLSNe, TDEs, PISNe, ILOTs) with deviations ranging up to 5 sigma. However, the prediction errors for these classes in the DNN are much smaller, not much more than 1 sigma deviations. This indicates, that the Bazin model is much worse at predicting these anomalous classes than the DNN, and hence is better at identifying them as anomalies, despite Figure 3.20 highlighting that the SNIa DNN model is better able to predict SNIa.

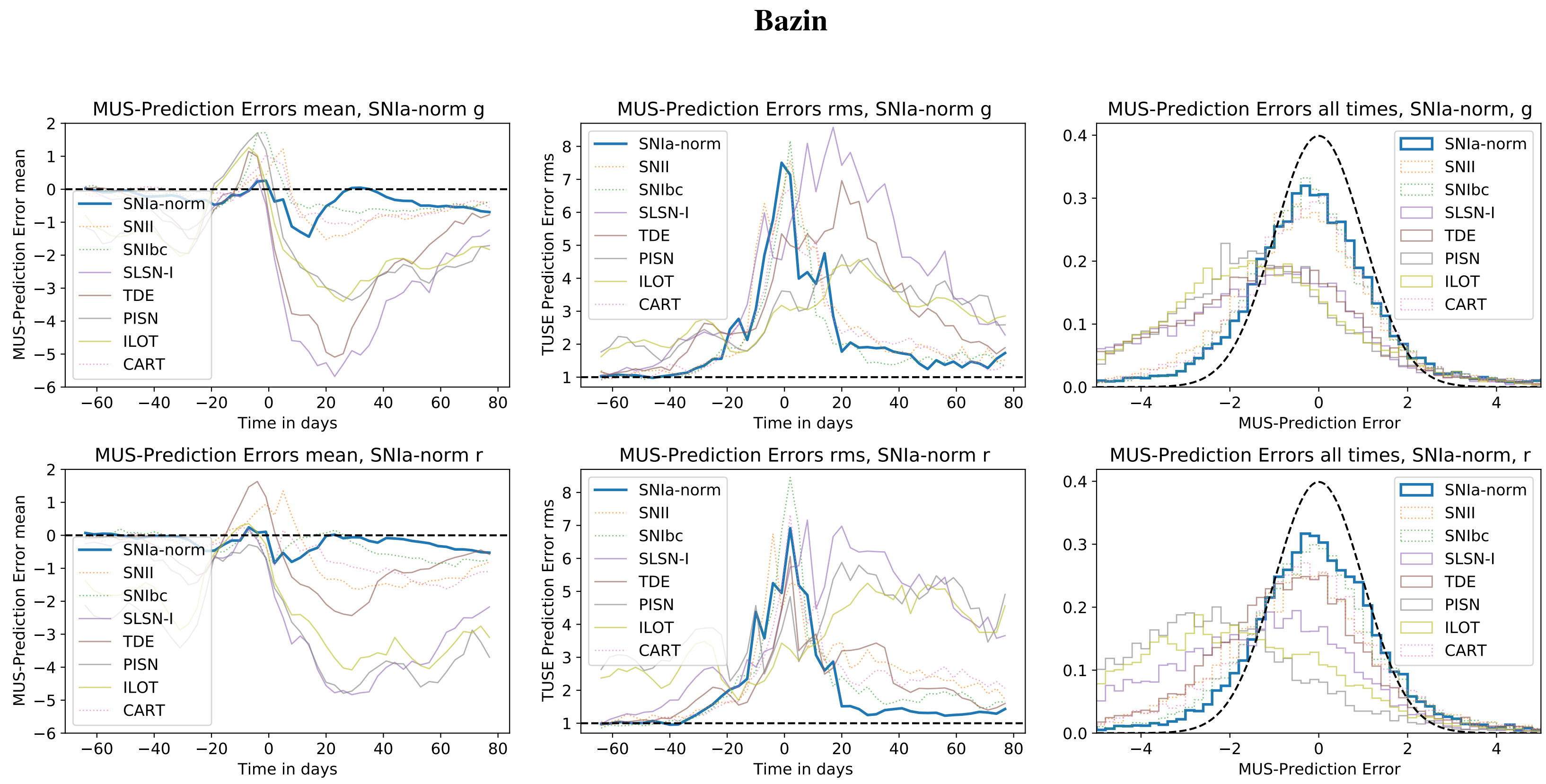


Fig. 3.21 Distribution of the Measurement-Uncertainty-Scaled Prediction Errors (MUSPE) for the Bazin SNIa model at different times. We apply the Bazin SNIa model to each other class' datasets and compute Equation 3.38 at each time-step. We have not plotted the Kilonova or uLens-BSR classes here because they show significant deviations, indicating that the SNIa model is very bad at predicting these classes, and hence easily identifies them as anomalies. We show the mean and root mean square (rms) for each prediction error distribution at each time-step in the first three panels, with the *g* band shown in the top row of panels, and the *r* band shown in the bottom row of panels. In the last column, we plot the distribution of scaled errors across all times. Similar plots were made for the SNIbc, SNII, Kilonova, SLSN and TDE Bazin models, but are not shown here for brevity. We plot a unit Gaussian as a black dashed line to help guide the eye.

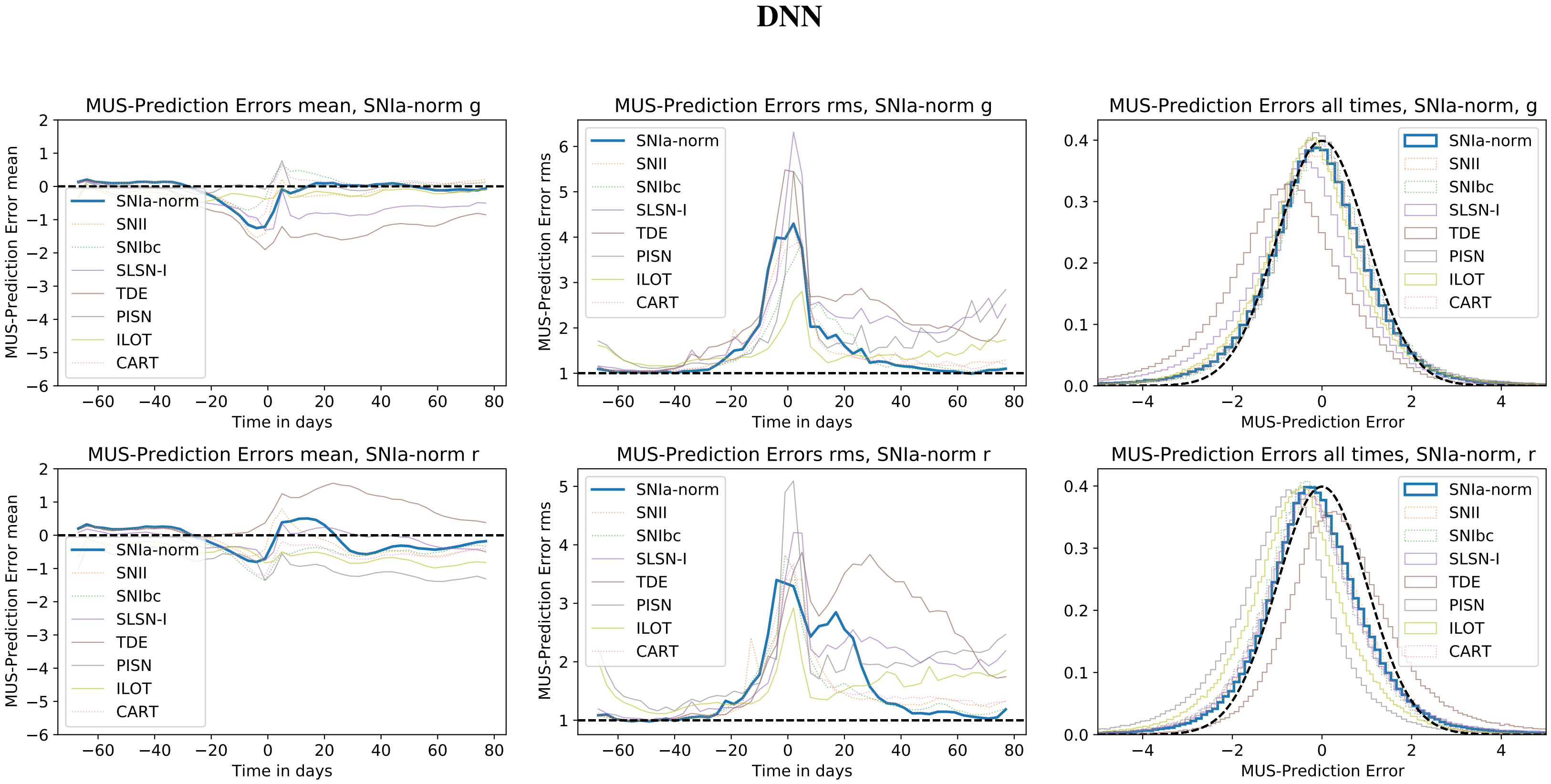


Fig. 3.22 Distribution of the Measurement-Uncertainty-Scaled Prediction Errors (MUSPE) for the DNN SNIa model at different times. We apply the DNN SNIa model to each other class' datasets and compute Equation 3.38 at each time-step. We have not plotted the Kilonova or uLens-BSR classes here because they show significant deviations, indicating that the SNIa model is very bad at predicting these classes, and hence easily identifies them as anomalies. We show the mean and root mean square (rms) for each prediction error distribution at each time-step in the first three panels, with the *g* band shown in the top row of panels, and the *r* band shown in the bottom row of panels. In the last column, we plot the distribution of scaled errors across all times. Similar plots were made for the SNIbc, SNII, Kilonova, SLSN and TDE Bazin models, but are not shown here for brevity. We plot a unit Gaussian as a black dashed line to help guide the eye.

Chapter 4

Classifying photometric light curves on real data

4.1 Overview

In Chapter 2, I developed a novel method of performing real-time classification of photometric light curves for upcoming and ongoing transient surveys. The results predominantly showed its application to simulations of the Zwicky Transient Facility (ZTF). In this chapter, I present the first real-time transient classifier that is designed for real data applications. I apply it to observations from the Pan-STARRS1 (PS1) photometric survey, the public ZTF MSIP survey, improved ZTF simulations (taking care of some issues with the original PLAsTiCC simulations), and a cosmological sample from the Foundation survey. I improve upon the previous RAPID methodology with the lessons learned in Chapters 2 and 3. In particular, I use a neural network architecture based on Temporal Convolutional Networks and use new data augmentation methods, developed by Boone 2019, that use Gaussian Processes to simulate new light curves. This new method helps to train classifiers on real datasets that are smaller than the previously used simulations.

I trained three classifiers on each of the following datasets:

1. PS1 spectroscopically-confirmed supernovae from Villar et al. (2020a),
2. Our collection of real transients from the ZTF MSIP public survey, and
3. PLAsTiCC-like ZTF simulations (improved from those used in Chapter 2).

We obtained excellent success on our PS1 classifier, obtaining average AUC scores of 0.98 only a few days after transient detection, which is a significant improvement over previous research. Applying our PS1 classifier to the 180 SNe Ia used in the Foundation Supernova Survey's cosmological sample (Foley et al., 2018b), we found that we could obtain a 100% pure sample of SNe Ia with a completeness of 91% by setting RAPID's SNIa acceptance threshold probability to 0.57. Our results use only the photometry and impressively shows that future cosmological surveys may be able to rely on photometry for a high-precision sample of SNe Ia.

Our ZTF classifier trained on real observations from the public survey achieved good performance, obtaining average AUC scores of 0.86 and 0.95 at detection and over the full light curve, respectively. Our last classifier trained on ZTF simulations improved upon the performance shown in Chapter 2. In particular, our use of the new, more realistic core-collapse supernova (CCSN) models from Vincenzi et al. (2019) resulted in better classifications of CCSNe. On the simulated testing set, we achieved excellent early classification performance with an average AUC score above 0.97 shortly after trigger. However, despite the improvements in our simulations, we found that the performance of the simulated ZTF classifier on our collection of real ZTF observations was poor, achieving an overall accuracy of 64% over the full light curves. We conclude that until simulations improve further, classifiers should be trained on real data if they are to have success on real observations.

We end the chapter by analysing the effect of observing strategy and in particular colour information on classification performance. We demonstrate that the inclusion of i-band in addition to the gr passbands in Pan-STARRS significantly improves classification performance, especially for SNe Ia. We also show that a classifier trained on $griz$ passbands is only marginally better than a classifier trained on only gri passbands, indicating that the additional z-band information is not very important for classification. This last section should prove informative for future surveys' decisions on which passbands should be included and for ongoing debates within the LSST community on optimal observing strategies.

4.2 Introduction

Current and next generation astronomical surveys focused on time-domain observations will produce unprecedented amounts of data. Pan-STARRS has already released the first ever Petabyte-sized data release (Flewelling et al., 2020), the ZTF is currently generating up to 400,000 transient alerts per night (Graham et al., 2019), and the upcoming LSST is expected to produce up to 100 times more (Ivezić et al., 2019). To process these data and distribute them in real-time to the astronomical community, several transient brokers have emerged to assist the community in identifying interesting transients and to selectively trigger follow-up observations (see Section 1.5 on transient brokers). The UK-based Lasair (Smith et al., 2019), USA-based ANTARES (Saha et al., 2016), and Chile-based ALerCE (Förster et al., 2020) are three of the largest and well-funded brokers dedicating efforts to preparing for the huge data volume of LSST. They are each currently processing alerts from ongoing surveys such as ZTF and Pan-STARRS.

Pivotal to these transient brokers is an effective real-time classifier capable of identifying potentially interesting transient alerts and distributing these to the community to enable a prioritised follow-up of transients. RAPID (detailed in Chapter 2 and Muthukrishna et al. 2019a) was one of the first such attempts at developing a real-time classifier capable of dealing with these huge data volumes. However, RAPID and other transient classifiers have only demonstrated their performance on simulations, rather than real data streams. Recent work such as Hosseinzadeh et al. (2020) and Villar et al. (2019) have pointed to this as a major issue in the literature of transient classifiers.

In fact, nearly all transient classifiers built to date (e.g Boone, 2019; Charnock & Moss, 2017; Ishida et al., 2019a; Möller & de Boissière, 2020; Muthukrishna et al., 2019a; Richards et al., 2012) have relied on simulated data for their training sets. However, the use of simulations implies that we understand the underlying population of each transient class or understand the generative model producing explosive transient light curves. But, with the absence of large datasets, it may not be a reasonable assumption that our simulations are a good representation of the diversity of transient light curves that new surveys will observe, or that our simulations are able to account for the diversity caused by actual survey conditions. Throughout this chapter, we highlight how we have tried to make our simulations as realistic as possible, and also some issues where our simulated light curves fall short of being good representations of real transient light curves.

There are several reasons for why most classifiers have relied on simulations for their training sets. One of the biggest is that we have not observed enough transients to get a large enough training set. The only class of transients where we might have a large enough sample of light curves that might resemble the true underlying diversity of the population is Type Ia Supernovae (SNe Ia). While SNe Ia are not as intrinsically common throughout the universe as Core-collapse Supernovae (CCSNe), they are observed in much higher quantities because of how bright they are in the optical wavelengths, and because of the many cosmological surveys that have dedicated efforts to finding large samples of SNe Ia (e.g. The Dark Energy Survey (DES, Dark Energy Survey Collaboration et al., 2016), The Foundation Supernova Survey (Foley et al., 2018b), The Supernova Legacy Survey (SNLS, Davis et al., 2007), and ESSENCE (Astier et al., 2006)).

However, even our observations of SNe Ia lack data at early times in a light curve, and simulations struggle to accurately model the early explosion time of supernova light curves. As discussed in Chapter 1, the progenitor and explosion mechanism of SNe Ia remains poorly understood, despite the wealth of research into modelling these events and their use in cosmology. Recently, the Young Supernova Experiment (YSE, Jones et al., 2021b), has begun searching for early-phase transients using the Pan-STARRS telescopes to better understand SNe progenitors (see Section 1.4 for a summary of YSE). For this reason among others, it is ever more important that we have real-time classifiers that can find transients at early times.

The use of simulated datasets for classification has remained because it acts as a reliable way to assess the performance of new classifiers. Supernova classification challenges such as the Photometric LSST Astronomical Time-series Classification Challenge (PLAsTiCC, The PLAsTiCC team et al., 2018, see Section 1.6), and the Supernova Photometric Classification Challenge (SNPhotCC, Kessler et al., 2010b) have developed simulations of a range of transient classes. These well-built simulations and classification challenges have enabled their widespread use for classification so that the performance of new classifiers can be easily evaluated and assessed against the simulations and previous work. While these photometric classification challenges have enabled fruitful new research and the development of novel and advanced techniques, they have also had the unwanted consequence that very little research has been put toward building classifiers trained on real data. Instead, most research has (perhaps unwisely) assumed that the algorithms trained on simulations can easily be adapted for use on real data. However, very little work has gone on to adapt these classifiers for applications to real data, and these photometric challenges based on simulations have been a distraction from the more difficult challenge of classifying real data.

Consequently, few classifiers have demonstrated their performance on real surveys for this reason among others. Recently, Hosseinzadeh et al. (2020) and Villar et al. (2020c) (hereafter called H20 and V20, respectively) have built classifiers trained on real Pan-STARRS data. Both H20 and V20 use a Random Forest classifier similar to the one we developed in section 2.7. However, they differ in how they extract features from the light curves to use in the Random Forest. H20 first fits each light curve with the parametric function from Villar et al. (2019) (which is an improved version of the Bazin function used in Chapter 2), and extracts the parameters from the model in each passband as features. They apply Principal Component Analysis (PCA) to their collection of model parameters, and use the first six principal components as features for their Random Forest classifier. On the other hand, V20 interpolates each light curve with a Gaussian Process (GP), before training a Recurrent Variational Autoencoder Neural Network (RAENN) on the GP light curves (similar to the classification work on variable stars by Naul et al. (2018)). The advantage of the RAENN is that it is unsupervised, and so they use both an unlabelled set of photometric supernovae as well as a labelled set of spectroscopically-confirmed supernovae to train the RAENN. V20 then extracts features from the encoded layer of the RAENN for the set of labelled supernovae, and trains the Random Forest classifier on these sets of features. These classifiers are the first to train on real supernovae from the Pan-STARRS survey. However, they are limited because they are retrospective classifiers, in that they require the full light curve to be observed before providing reliable classifications of transients.

In this Chapter, we improve RAPID and develop real-time classifiers that, for the first time, are aimed at performing well on real data streams. We build classifiers for Pan-STARRS and ZTF, and compare their performance when trained on simulations and real observations. In Section 4.3, we describe the data our classifiers are built from. In Section 4.4, we improve the RAPID architecture from the lessons learned in the first two chapters, and outline our new architecture and data augmentation method for classifying real data. In Section 4.5, we detail the results of our classifier on real data, compare our PS1 classifier to other author's research, and illustrate how RAPID can be used to obtain high-precision samples of supernovae for cosmology without requiring spectra. In Section 4.6, we analyse the effect of survey observing strategy and the impact of different Pan-STARRS passbands on classification performance. We conclude and highlight future directions of this research in Section 4.7.

4.3 Data

4.3.1 Pan-STARRS data

The Panoramic Survey Telescope and Rapid Response System Telescope 1 (Pan-STARRS1 or PS1, Chambers et al., 2016) is a 1.8m diameter wide-field survey telescope located near the summit of Haleakala in Hawaii. It uses a 1.4 gigapixel camera with a 7.1 deg^2 field of view to survey the northern sky in five passband filters, *grizy*. The PS1 Medium Deep Survey

(MDS), one of several surveys running on Pan-STARRS, revisited 10 single-pointing fields covering a total sky-area of approximately 70 deg^2 in four passband filters *griz* from 2010 to 2014. The MDS fields consisted of about 25% of the overall survey observing time, and were observed with an approximately 3-day cadence in each passband with a 5σ limiting magnitude of ~ 23.3 per visit. Due to weather and other conditions, Scolnic et al. (2018) found that the average cadence was closer to 6-7 days in each passband. The survey typically observed the g and r passbands on the same night, followed by the i and z passbands on subsequent nights.

In this chapter, we use the supernova light curves collected and processed by V20 taken from the dataset available on Zenodo (Villar et al., 2020a). The dataset consists of 5243 SN-like objects from the PS1-MDS where only events with at least three observations with a S/N > 4 were used. V20 processed the PS1 data using `photpipe` (Rest et al., 2005, 2014; Scolnic et al., 2018), and performed image subtraction using `HOTPANTS` (Becker, 2015) before performing forced point-spread function (PSF) photometry on the centroid of the images. They determined the redshifts from spectra of the host galaxies of each transient. V20 then cut objects with poor redshift estimates, and ones that were variable over multiple seasons and were hence unlikely to be supernovae. The full processing method is detailed in Section 2 of their paper. After processing and selection cuts, the final sample includes 2885 transients with redshift estimates. Within this collection, 557 supernovae were spectroscopically-confirmed and observed in real-time throughout the survey.

H20 uses the 557 spectroscopically-confirmed supernovae as the training set for their supervised Random Forest classifier before testing their classifier's performance on 2315 transients. Similarly, V20 uses the 557 spectroscopically-confirmed supernovae as the initial labelled training set for their semi-supervised classification approach before using the unlabelled supernovae to help train the classifier. Both of these classifiers require the full phase coverage of transients before providing classifications, which is unlike RAPID that provides real-time classifications. Nevertheless, to compare our classifier to these works, we similarly only use the 557 spectroscopically-confirmed supernovae to train our algorithm before testing our classifier's performance on the remaining unlabelled transient light curves.

The class distribution of these 557 spectroscopically-confirmed supernovae is illustrated in Figure 4.1.

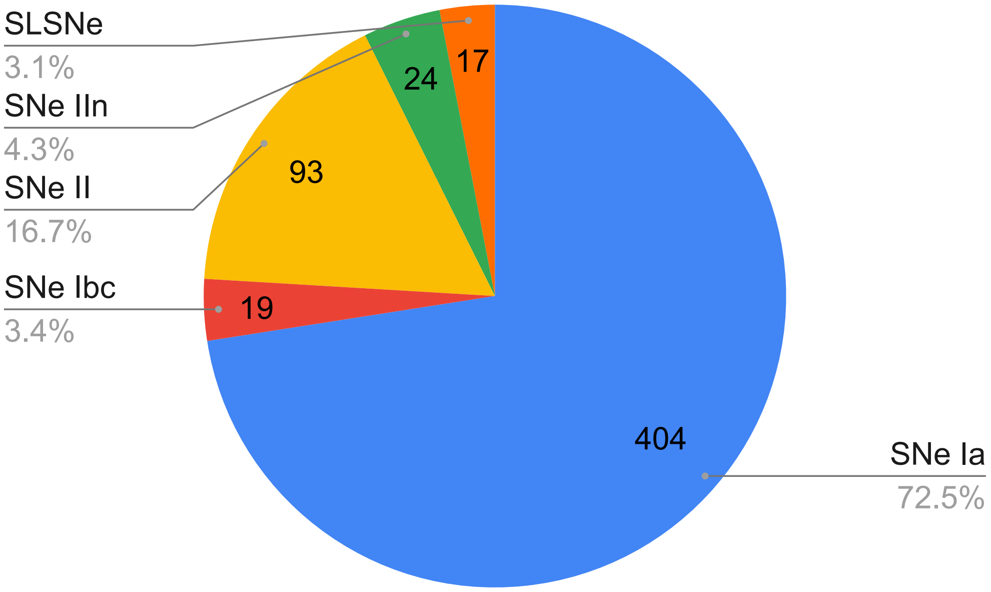


Fig. 4.1 The distribution of transients across the five labelled classes in our collection of PS1 spectroscopically-confirmed supernovae.

4.3.2 ZTF simulations data

The Zwicky Transient Facility (ZTF, Bellm et al., 2019) is a new optical synoptic survey on the 48-inch Schmidt telescope situated on the Palomar Mountain Range in California, USA. It has a very wide field of view camera (57 deg^2) that allows it to regularly scan the entire northern sky at a depth of 20.5-21 mag per visit (Graham et al., 2019). The ZTF Mid Scale Innovations Program (MSIP) is the public survey that streams up to one million transient alerts each night in two passband filters (gr) with an intended cadence of 3 days per filter.

In this chapter, we use improved versions of the PLAsTiCC-like ZTF simulations built for Chapters 2 and 3. Several changes have been made from the previous simulations to fix a range of issues, these changes are detailed in the following subsections.

4.3.2.1 Updated Core-collapse Supernova models

As described in Kessler et al. (2019), the core-collapse supernova models developed for the PLAsTiCC simulations were created using augmentations to observed supernova light curves and empirical models. A time-series of spectral energy distributions (SEDs) were developed using observed CCSN light curves from the Sloan Digital Sky Survey (SDSS) and the

Carnegie Supernova Project (CSP). Light curves were simulated from this SED to generate some of the CCSNe used in PLAsTiCC. Further light curves were developed by empirically simulating a time-series of SEDs using the `MOSFiT` software package (Guillochon et al., 2018b). `MOSFiT` takes in a number of physical parameters such as the supernova ejecta mass, ejecta velocity, and information about the circumstellar material surrounding the supernova. The parameters used to generate the SED simulations were taken from the theoretical models developed in Villar et al. (2017).

Problems with these CCSN simulations were noted after these simulations were published and used in PLAsTiCC and RAPID. In private communications with Dr Maryam Modjaz[1], she highlighted an inconsistency between the PLAsTiCC light curves and real observations of CCSNe. In particular, she noted that our simulations often have SNIbc light curves that are wider than SNII light curves. This issue is visible in the example light curve simulations plotted in Figure 2 of my RAPID paper (Muthukrishna et al., 2019a) (also reproduced in Figure 2.2 in Chapter 2). However, Dr Modjaz notes that all observations of CCSNe point to the opposite conclusion, and she has produced Figure 4.2 to exemplify this (published in Figure 2 of Modjaz et al. 2019). Figure 4.2 plots example SNII and SNIbc light curves in the V-band, illustrating the photometric diversity of CCSNe. In particular, the figure highlights the general trend that stripped-envelope supernovae (SNe Ibc) have narrower light curves than SNe II, contrary to light curves from the original PLAsTiCC models.

This issue might explain RAPID's poor performance on classifying CCSNe as illustrated by the RAPID performance metrics illustrated in Figures 2.6-2.11 from Chapter 2. In this chapter, we have replaced the CCSN models described above with new models from Vincenzi et al. (2019). These new models use a completely data-driven approach with no assumptions of any parametric form for light curves. They use photometric and spectroscopic data from 67 CCSNe to generate time-series spectral templates. These templates were integrated into `SNANA` and we have recently included them in the PLAsTiCC modelling code-base. Using these models, we have simulated a range of Type II, IIn, IIb, Ib, Ic, and Ic-BL supernovae to include in our training set.

4.3.2.2 Updated Observing logs

The PLAsTiCC code-base incorporates survey conditions, including the weather, cadence properties, seeing, clouds, sky brightness, and telescope maintenance downtime, to simulate

[1] Professor of Astrophysics at NYU, https://as.nyu.edu/faculty/maryam-modjaz.html

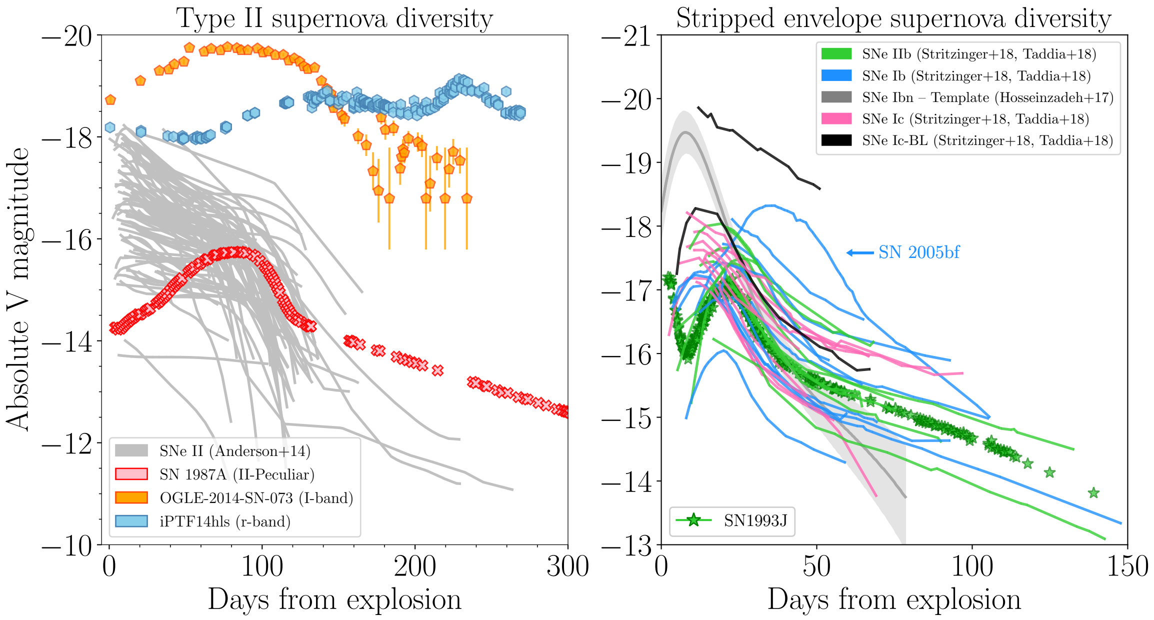


Fig. 4.2 Photometric diversity of SNII (left figure) and SNIbc (right figure) light curves in the V-band. The plots illustrate the general trend that stripped-envelope supernovae (SNe Ibc) have much narrower light curves than SNe II. This highlights an issue with the original PLAsTiCC models as highlighted by the opposite trend shown in Figure 2.2. Image is taken from private communications with Dr Maryam Modjaz and was published as Figure 2 of Modjaz et al. (2019).

light curves that more realistically match observations. When we first developed the ZTF simulations for RAPID in 2018, ZTF had only been running for just under 1-year, and so, we included the ZTF observing conditions from 1-year's worth of observations into the PLAsTiCC model. Our new simulations incorporate the observing conditions from nearly 3-years of ZTF and use an updated observational efficiency file, which should provide more realistic simulations that match the ZTF survey conditions. We obtained these files from Eric Bellm[2], and my collaborators, Patrick Aleo and Gautham Narayan[3], created these new simulations from the PLAsTiCC code-base.

4.3.2.3 Updated SNIa-91bg models

In Section 4.2.2 of Kessler et al. (2019), they note that after publishing the PLAsTiCC models, a mistake in the code meant that the SNe Ia-91bg were not modelled with enough variation. Specifically, only a single stretch value was used instead of a continuous range when simulating SNIa-91bg light curves. This meant that the simulated SNIa-91bg light curves only accounted for a small subset of the underlying population of SNe Ia-91bg expected in real data; thus meaning that if compared against real data, many observed SNe Ia-91bg would not match the simulated dataset.

In January 2021, this software bug was fixed (PLASTICC Team & PLASTICC Modelers, 2021), and we have incorporated these new PLAsTiCC models into our ZTF simulations used in this Chapter.

4.3.2.4 Updated SNIa-x models

A recent analysis of new supernova observations from the Dark Energy Survey, by Vincenzi et al. (2020), has compared supernova modelling parameters from simulations such as PLAsTiCC with observed data. Among many other trends, the analysis found that the $B-V$ colour distribution at B-band peak for SNe Iax has a much wider distribution than the original PLAsTiCC models (this is illustrated clearly in Figure 4 of Vincenzi et al. 2020). To fix this, the latest PLAsTiCC models (PLASTICC Team & PLASTICC Modelers, 2021) use dust extinction to introduce variation in the colour of the models. They use a reddening

[2] Survey scientist for ZTF and Professor of Astronomy at the University of Washington https://faculty.washington.edu/ecbellm/

[3] Graduate student and Professor of Astronomy at the University of Illinois Urbana-Champaign, respectively https://astro.illinois.edu/directory/profile/paleo2

$E(B-V) = 0.09$ with $R_V = 3.1$ and use the extinction law from Cardelli et al. (1989) to correct the simulated SNe Iax for dust extinction and thus reproduce simulations that more closely match the colour diversity of observed SNe Iax.

4.3.2.5 Updated Kilonova models

The original PLAsTiCC models (presented in Kessler et al. (2019) and used to simulate the ZTF transients in the previous chapters) simulated kilonovae using a set of SED time-series models of Binary Neutron Star mergers from Kasen et al. (2017). Since there has only been one kilonova event observed (GW170817, Abbott et al. 2017c), the diversity of kilonovae expected to be observed from new surveys is unknown. The models from Kasen et al. (2017) provide a theoretical model based on three parameters: the ejecta mass, ejecta velocity and lanthanide fraction of kilonovae. The original PLAsTiCC models used a uniform distribution of parameters close to the parameters of the observed GW170817 event, to simulate a diverse range of kilonovae.

In this chapter, we use the latest PLAsTiCC models (PLASTICC Team & PLASTICC Modelers, 2021) that use newer kilonova models from Bulla (2019) as well as the previous models from Kasen et al. (2017). The models from Bulla (2019) use a different modelling approach that uses four parameters: the ejecta mass, the temperature one day after the neutron star merger, the observer viewing angle, and the opening angle of the lanthanide-rich component. These two sets of models will ideally provide a more diverse range of kilonovae that encompass the expected diversity of observed light curves from upcoming surveys.

4.3.2.6 Simulated dataset

After including the new PLAsTiCC models in our ZTF-like simulation code-base, we simulated approximately $\sim 10,000$ ZTF transients in each of the following classes: SNIa, SNII, SNIbc, SNIa-91bg, SNIa-x, Kilonova, and SLSN-I. These simulations once again mimicked the observing properties of the ZTF MSIP survey with a target cadence of 3-days in the g and r passbands. As described in the previous chapters, each simulated transient includes the host-galaxy redshift, sky position, Milky Way dust reddening, and a time-series in the g and r ZTF passbands of the observed fluxes, flux uncertainties, and a flag indicating whether the flux was a detection or non-detection. We do not include a few of the rarer classes (point-Ia, PISN, ILOT, and CART) included in the first version of RAPID, because we have observed too few of these transients to trust their light curve models, they are likely to be very rare in ZTF, and we think that anomaly detection frameworks such as the ones detailed in Chapter 3 are better suited to identifying such objects.

4.3.3 ZTF real data

The Zwicky Transient Facility has been observing transients since 2017, and has successfully identified several thousand supernovae. When RAPID was developed, ZTF had not yet identified many supernovae, and thus it was not feasible to use real observations to train a classification algorithm. To obtain a labelled set of ZTF transients, we opted to use the wealth of transients labelled by the astronomical community. We searched the Open Supernova Catalog (OSC)[4] (Guillochon et al., 2017) for objects with ZTF aliases, and after removing all events without listed supernova classifications, we obtained 2839 labelled supernovae. The vast majority of the observed transients were labelled as a SNIa or a SNIa subtype and consisted of 75% of the labelled transients. The distribution of the transients in four broad classes is illustrated in Figure 4.3, and the distribution of transients for each labelled transient class (except for those labelled 'Ia' which comprise 1993 objects) is illustrated in Figure 4.4.

We downloaded the light curves and transient data from the LASAIR alert broker (Smith et al., 2019). Unlike the ZTF simulations, the real data from the ZTF MSIP data stream are distributed in alert packets that contain magnitudes instead of flux units. We convert these magnitudes and magnitude uncertainties to flux counts and uncertainties as follows,

$$\begin{aligned} F &= 10^{-0.4(\mathrm{mag}-\mathrm{zpt})} \\ \sigma_F &= |F(10^{0.4\sigma_{\mathrm{mag}}}-1)| \end{aligned} \tag{4.1}$$

where F is the flux, mag is the calibrated PSF-magnitude in the ZTF alert packet, σ_F is the uncertainty in the flux, σ_{mag} is the magnitude uncertainty, and zpt is the zeropoint that we use to scale all fluxes.

In the previous applications of RAPID to real ZTF observations, we misunderstood the calibration method in the ZTF alert packet and mistakenly used the associated zeropoint magnitude for zpt in equation 4.1. Since the mag in the alert packet is already calibrated, using the zeropoint from the alert packet effectively uncalibrates the flux measurement. Thus, our previous approach on applying RAPID to real data will produce erroneous results. This issue was pointed out to us in private communications with Prof Stephen Smartt and Michael

[4] We searched the catalog, https://sne.space/, on 22 Oct 2020

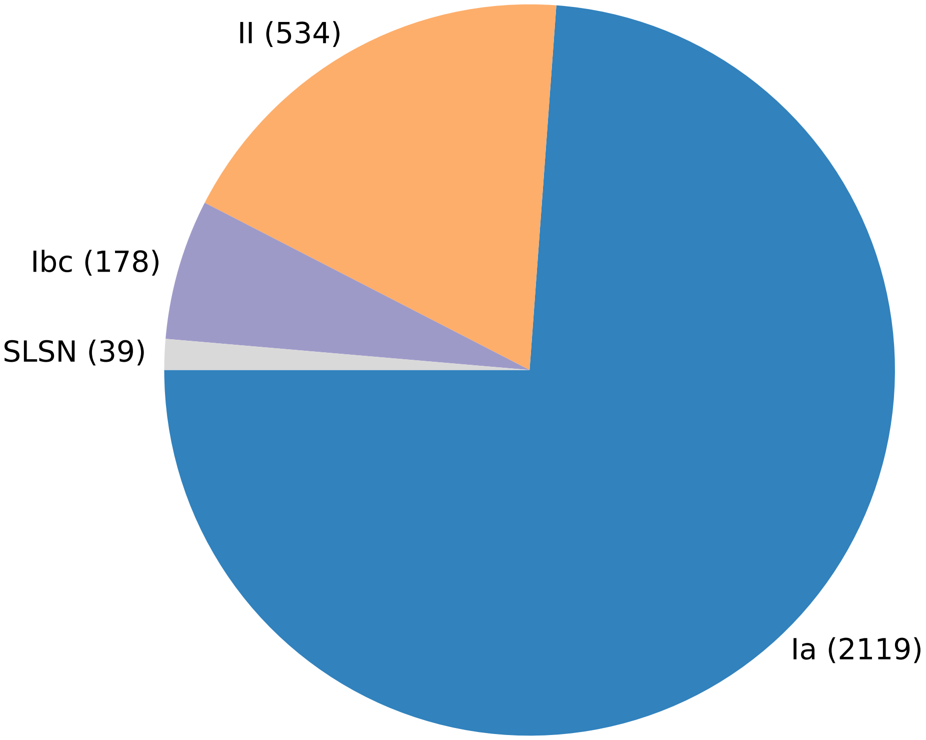


Fig. 4.3 Distribution of real ZTF transients with the light curves downloaded from the LASAIR alert broker (Smith et al., 2019) and the labels sourced from the Open Supernova Catalog (Guillochon et al., 2017). The transient labels are grouped into four broad types in the pie chart: Ia (including transients labelled as Ia, Ia91T, Ia91bg, Iacsm, Iapec, and Iax), II (including transients labelled as II, IIL, IIP, IIn, IIpec, and IIb), Ibc (including transients labelled as Ib, Ibn, IIb, Ic, IcBL, Ibc), and SLSN (including transients labelled as SLSNI and SLSNII).

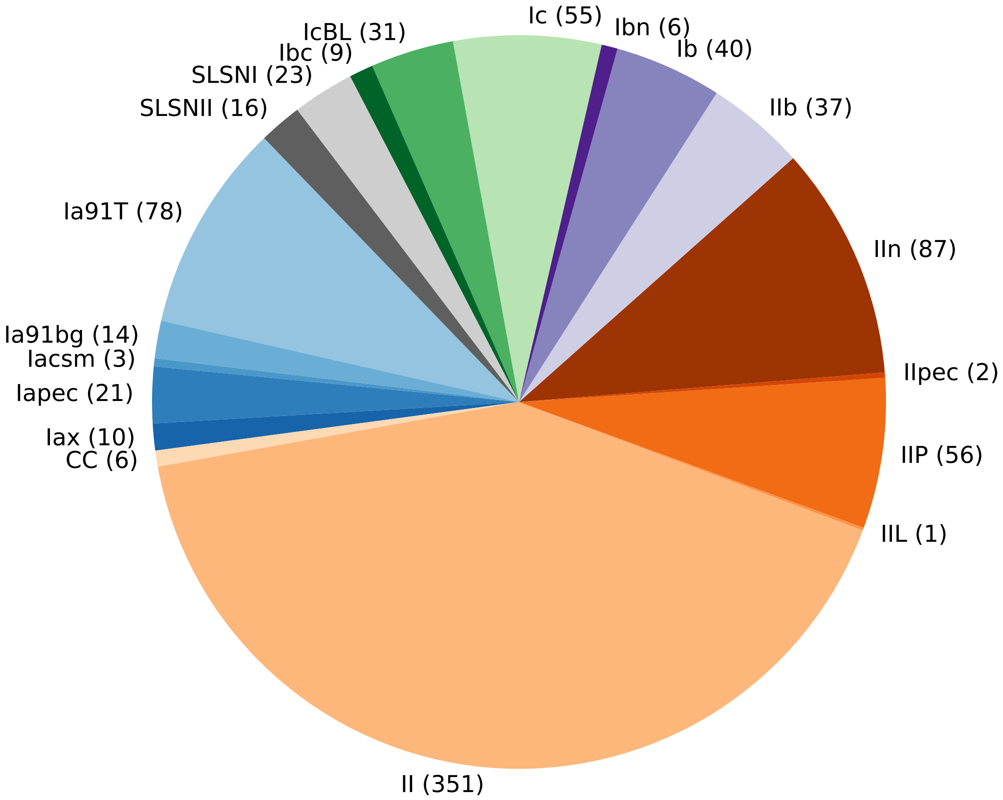


Fig. 4.4 Distribution of real ZTF transients with the light curves downloaded from the LASAIR alert broker (Smith et al., 2019) and the labels sourced from the Open Supernova Catalog (Guillochon et al., 2017). The SNe Ia are not shown in the pie chart because they comprise 1993 objects, account for 70% of the dataset, and would make it difficult to see the other classes. Mostly consistent with the colours from Figure 4.3, the Ia, II, Ib, Ic, and SLSN subclasses are shown in shades of blue, orange, purple, green, and grey respectively.

Fulton[5], researchers who have worked on implementing RAPID into the LASAIR transient broker. They ran an analysis on RAPID applied to real ZTF data, and highlighted some problems with RAPID's ability to classify real ZTF transients.

In this chapter, we retrain RAPID on improved simulations and retrain on real data to fix the issues pointed out in Prof Smartt and Michael Fulton's analysis. We improve upon the mistake in Chapter 2 when applying RAPID to real data, and note that the zeropoint selected for equation 4.1 is arbitrary and will just have a scaling effect that should not affect a classifier trained on real observations. However, if we wish to apply our classifier trained on ZTF simulations to this real dataset, then we should adjust the zeropoint to match the scaling of the simulated light curves. The following subsection details our method for selecting a zeropoint.

4.3.4 Comparison of ZTF SNIa simulations and real data

To match the scaling of the real data to the simulations, we analyse the SNIa flux distributions for the PLAsTiCC simulations detailed in Section 4.3.2 and the collection of real ZTF data. Since we do not have a large enough collection of real observations from the other transient classes, we only compare the SNIa distributions, and assume that the zeropoint scaling that is most effective for this class will be the same for the other classes.

In Figure 4.5, we plot the flux uncertainties against the fluxes for all the SNIa observations in our simulated dataset (blue lines) and our collected real data (orange lines). We varied the zeropoint of the real data observations in equation 4.1 until the distributions visually overlapped as much as possible. After a range of zeropoint selections, we determined that setting the zeropoint of the real SNIa observations to $\text{zpt} = 26.2$ led to the closest matching distributions in the flux uncertainty vs flux space in both the g and r passbands. We fit each distribution with a Kernel Density Estimation (KDE) so that the distributions were smoother and could be more easily compared.

While the population distributions overlap quite well, we notice that the median of the distributions in the real data are slightly brighter than the simulations in both the g and r bands. This bias is to be expected because brighter supernovae are more likely to be classified than fainter events in the collection of real observations. This trend is particularly obvious if instead of plotting the fluxes and flux uncertainties, we plot the peak flux and corresponding

[5] Professor of Astrophysics and PhD Student at Queen's University Belfast https://pure.qub.ac.uk/en/persons/stephen-smartt

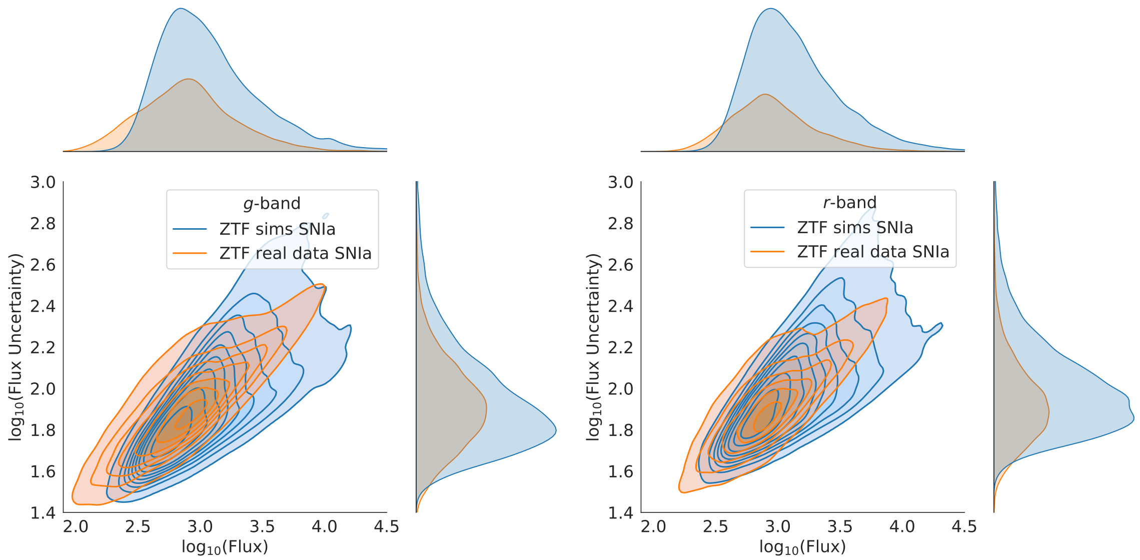


Fig. 4.5 Comparison of the flux and flux uncertainty distributions for the ZTF SNIa simulations and real data in the *g*-band (left) and *r*-band (right). We set the zeropoint of the real ZTF MSIP observations to zpt = 26.2. We have used a Kernel Density Estimation (KDE) to help visualise the distributions. The 1D marginal distributions of the flux and uncertainty are plotted above and to the right of the plot, respectively.

uncertainty for each SNIa light curve in our datasets as in Figure 4.6. Here, we see that most real SNe Ia have a peak flux that is brighter than the median of the simulated SNIa peak fluxes in both the *g* and *r* passbands. However, the real data contours are entirely within the simulated KDE contours, indicating that the brighter SNIa simulations are able to model light curve peak fluxes seen in real observations.

In Figure 4.7, we plot the distribution of peak fluxes against redshifts for the simulated and real SNe Ia. These plots make it clear that our collection of real ZTF SNe Ia have a much lower redshift distribution than what we expected from simulations. This might be explained by the previous point that our collection of real labelled transients are biased to be brighter objects with a higher signal-to-noise, since these are more easily classified. Lower redshift supernova typically have a higher signal-to-noise, and are more easily classified. However, if the simulations accurately represented the distribution expected from real data, we would expect that the real data contours would lie within the simulated contours. Instead, it appears that there are a sizeable collection of real SNe Ia at low redshift that have a lower peak flux than we expected from our simulations. This mismatch in distribution shape may explain some of the issues in classifying real supernova from our classifier trained on these

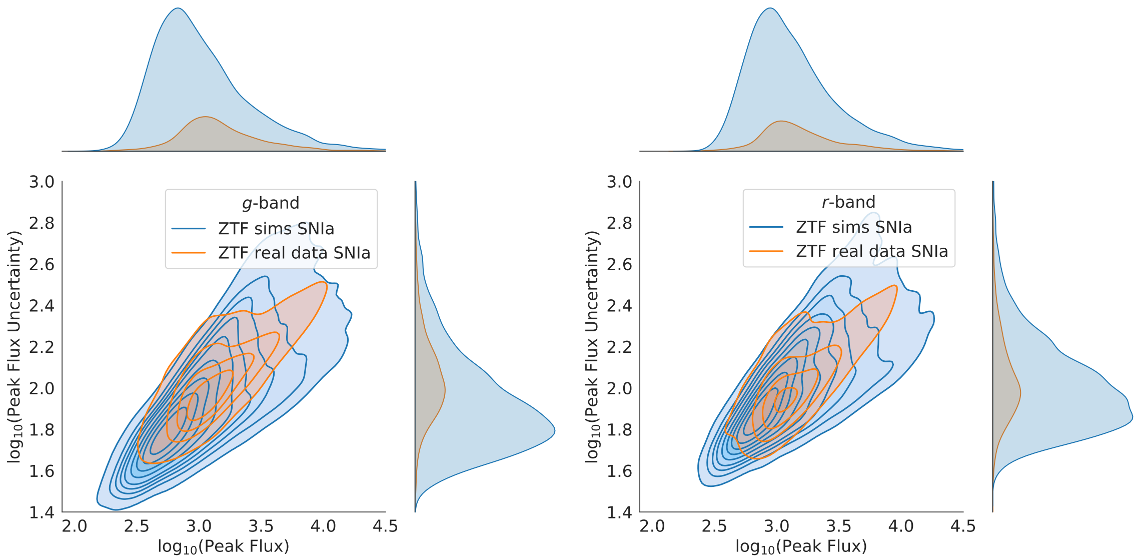


Fig. 4.6 Comparison of the peak flux and peak flux uncertainty distributions for the ZTF SNe Ia simulations and real data in the *g*-band (left) and *r*-band (right). We set the zeropoint of the real ZTF MSIP observations to zpt = 26.2. We have used a Kernel Density Estimation (KDE) to help visualise the distributions. The 1D marginal distributions of the peak flux and uncertainty are plotted above and to the right of the plot, respectively.

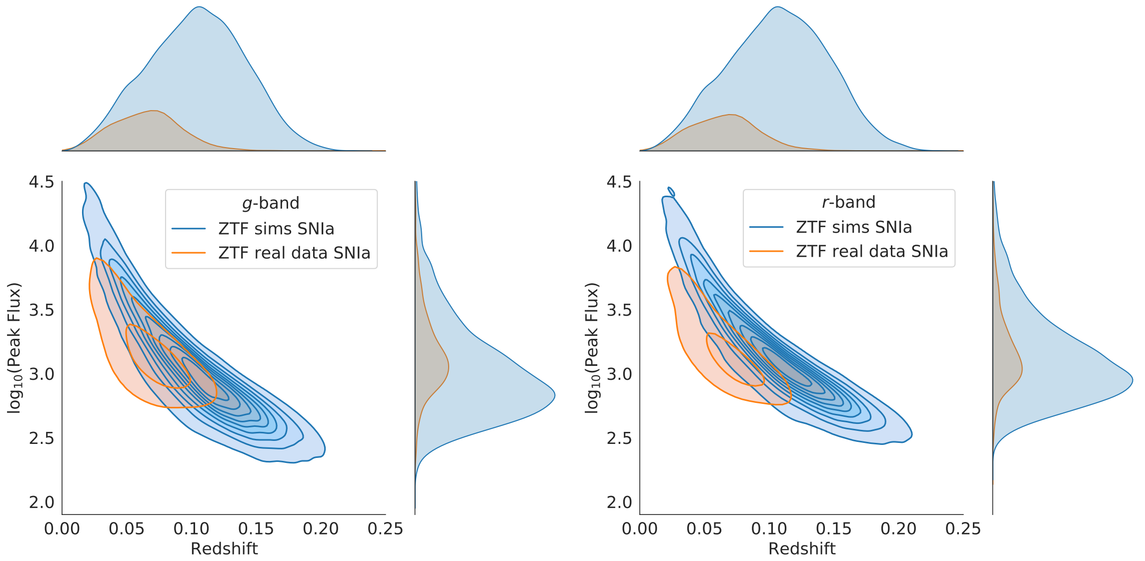


Fig. 4.7 Comparison of the peak flux and redshift distributions for the ZTF SNe Ia simulations and real data in the *g*-band (left) and *r*-band (right). We set the zeropoint of the real ZTF MSIP observations to zpt = 26.2. We have used a Kernel Density Estimation (KDE) to help visualise the distributions. The 1D marginal distributions of the peak flux and redshift are plotted above and to the right of the plot, respectively.

simulations later in this chapter. I have shared these results with the PLAsTiCC team, and in the upcoming PLAsTiCC v2.0, these issues will aim to be resolved.

4.4 Method

4.4.1 Data Augmentation

In Figures 4.1, 4.3, and 4.4, it is obvious that not only do we have a small collection of observed transients to train a classifier, but also, there are very few examples of some transient classes. The collection of transients in our real datasets are biased towards bright and low-redshift objects (this is evident in Figure 4.7). The reason for this bias is because brighter and lower redshift objects are typically easier to obtain spectra and classify.

Previous research has concluded that training photometric classifiers on a biased spectroscopically confirmed dataset performs very poorly when compared to training on data that is more representative of the variety of observations expected in photometric surveys (e.g. Lochner et al., 2016). Many authors (e.g. Karpenka et al., 2013; Lochner et al., 2016; Richards et al., 2012) have advocated that future spectroscopic surveys should prioritize faint objects for spectroscopic targeting in an attempt to make spectroscopically-confirmed datasets more representative of objects observed in photometric surveys. Recently, Ishida et al. (2019b) has proposed a framework that selects the most informative objects that should be followed up spectroscopically in order to optimise supernova photometric classification. They demonstrate that by spectroscopically targeting objects that a classifier is most uncertain about, and retraining photometric classifiers with these newly labelled objects, the performance of photometric classifiers can improve greatly. Thus, prioritised follow-up using active learning frameworks, such as that proposed by Ishida et al. (2019b), will improve the performance of classifiers in the future.

However, given the bias of current observed datasets and the small numbers of objects available, it is challenging to train a classifier that learns the underlying diversity of data in each transient class. While we may have enough SNe Ia to inform a good SNIa training set, there are likely not enough transients from the rarer classes to represent the variety of light curves expected in new observations. Previously, we have dealt with this by resorting to simulations of transients. However, in this chapter we aim to build and compare the performance of a classifier trained on only real data. We thus instead resort to data augmentation techniques that will help increase the size of our training set, and ideally make a training set that is more representative of the diversity of transients we expect from future observations.

The recent transient classification challenge, PLAsTiCC, had over 1000 classification entrants with a wide range of different algorithms (Hložek et al., 2020, see Section 1.6). The PLAsTiCC dataset had a class imbalance to represent the expected distribution of transient classes observed from LSST in the first 3-years of observations. Thus, central to the approaches that performed well at the challenge was good data augmentation techniques that helped train an algorithm to perform well on unseen test data. The winning algorithm was proposed by Kyle Boone and is described in Boone (2019). One of the key parts of his algorithm that enabled him to outperform the other teams was his unique light curve augmentation method. His algorithm's performance on unseen data demonstrated an ability to learn the diversity of data in the testing set despite a limited training set.

4.4.1.1 Gaussian Process Augmentation Method

In this chapter, we apply a similar data augmentation method described in Boone (2019) to our imbalanced real PS1 and ZTF training sets. We adapt the data augmentation methods in the open-source code, `Avocado`[6], released in Boone (2019) for our work. Pivotal to the augmentation method is to first fit a Gaussian Process (GP) to each light curve. Previously, Lochner et al. (2016) and Revsbech et al. (2018) introduced using GPs for astronomical transient classification. These methods evaluated separate GPs for each band of the light curve. Boone (2019) instead uses a GP in both time and wavelength to fit the light curve in all passbands simultaneously, and thus take cross-band information into account.

We use the `george` package (Ambikasaran et al., 2015) to construct a GP that is a smooth, continuous function that interpolates the discrete, irregularly sampled, and noisy data points of each light curve. We use a 2-dimensional Matern kernel and use maximum likelihood estimation to fit both the amplitude and time length scale parameters for each transient. As in Boone (2019), we fix the length scale in wavelength to 6000Å because with only a few passbands it is not possible to reliably fit the wavelength length scale. In Figure 4.8, we plot an example of our GP's interpolation of a SLSN light curve from our PS1 training set. We use the same kernel to fit a GP to all transients in our PS1 and real ZTF training sets.

We use the GP fits to generate several augmented versions of each light curve until we reach a balanced dataset with 5000 light curves in each class. We simulate each transient at

[6] https://github.com/kboone/avocado

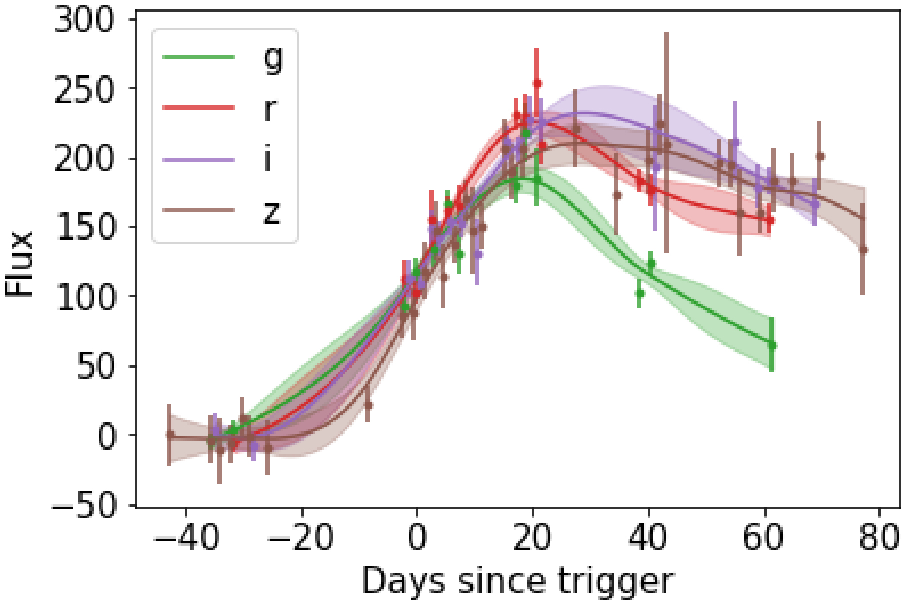


Fig. 4.8 Example Gaussian Process interpolations of a SLSN from the PS1 spectroscopically-confirmed dataset. The data is shown with uncertainties, the solid line is the mean of the GP flux predictions for each passband, and the shaded region shows the 1σ variations of the GP predictions.

different redshifts, by resampling its light curves at different times, and by adding noise to the measurements. Our complete augmentation method adapted from Avocado is detailed as follows.

We first choose a new random redshift z_{new} based on the observed redshift z_{original} from a log-uniform distribution in the range $0.95 z_{\text{original}} < z_{\text{new}} < 5 z_{\text{original}}$. Following Boone (2019), we also restrict the possible shift in wavelength to 50% so that the GP doesn't have to extrapolate too far from where data is available. We impose this as an additional upper bound on the redshift of $(1+z_{\text{new}}) < 1.5(1+z_{\text{original}})$. The lower bound on the redshift helps avoid making faint objects too bright and thus prevents the newly simulated observations from being dominated by large uncertainties. The upper bound on the redshift helps to avoid generating objects that would be too faint to be detected by a telescope.

Next, we select new observation times for the generated light curve. We scale the observed times to account for time dilation due to the newly selected redshift. Shifting to a higher redshift causes the light curve to have a lower density of observations. To account for this, we add observations to one passband of the light curve at the same times as existing observations in a different passband. We do not interpolate observations at new times because the GP flux predictions too far from the observed data have much large uncertainties that lead to unrealistic light curves. We then randomly drop 10% of the observations in this new light curve to introduce more variation in the generated light curves.

We use our GP fit of the original light curve to make flux predictions at these new set of observation times and wavelengths corresponding to the passbands shifted by the new redshift. We only use the mean GP flux predictions rather than drawing flux predictions from the GP so that we get more reliable interpolations. However, we include the 1σ uncertainty of the GP flux predictions when we simulate the observation noise in the new light curves.

To simulate observation noise, we first fit log-normal distributions to the collection of all flux uncertainties in the un-augmented training set for each passband. We then draw an uncertainty from the distribution and add it in quadrature to the uncertainty of the GP flux predictions. We use this as the simulated uncertainties of the observations in the generated light curves of the augmented training set.

Next, we correct the fluxes and flux uncertainties of the new light curve to account for the luminosity distance of the new redshift. We compute the luminosity distance assuming a fiducial flat ΛCDM cosmology with $\Omega_M = 0.3$, $\Omega_\Lambda = 0.7$ and $H_0 = 70$ km s^{-1} Mpc^{-1}.

Finally, we label all new observations with a $\mathrm{S/N} > 5$ as detections and other observations as non-detections; we use the first detection as the date of trigger. If the newly generated object has less than three detections in its light curves, then we re-simulate it using a newly drawn redshift. We repeat this process for every transient in our un-augmented training set until we have 5000 transients in each class.

4.4.2 Deep Neural Network

We use a very similar architecture to that used in the original `RAPID` described in Section 2.4. However, we replace the Gated Recurrent Units (GRUs) with a Temporal Convolutional Network (TCN). We described some of the benefits of TCNs over RNNs in Chapter 3 where we used a TCN architecture to predict future fluxes in a light curve (see section 3.4.1.2). Jamal & Bloom (2020) extensively compared the use of TCNs, GRUs, and LSTMs for the classification of variable stars and detailed the benefits of each. We have briefly compared the performance of all three of these architectures for our classification work, and note that there is very little difference in performance. However, similar to what we outlined in Section 3.1, the TCN architecture appears to be much faster to train (due to the parallelisable nature of Convolutional Neural Networks vs Recurrent Neural Networks), and seem to be much more stable during training time (as was also shown in Jamal & Bloom (2020)). We noticed that the RNN architectures had losses and accuracies that changed wildly during training, whereas the TCN was more stable when converging to the final loss and accuracy, further justifying our change in architecture.

In Figure 4.9, we illustrate our new TCN architecture for RAPID, where the input images $\boldsymbol{I}^{st}$ consisting of light curve fluxes in all passbands, the host galaxy redshift, and the Milky Way reddening, are passed into a TCN before classification probabilities $\boldsymbol{y}^{st}$ are extracted with a fully connected layer with a softmax regression activation function. Other than this change in architecture, the model is otherwise the same as described in the original RAPID work discussed in Chapter 2.4.

4.5 Results

In this section, we build three separate RAPID classifiers. One trained on a collection of PS1 data, another trained on a collection of real ZTF data, and a third trained on improved PLAsTiCC-like ZTF simulations. We first validate the performance of each classifier by evaluating its performance against a validation set (20% of the dataset that is not used for training). We then test the performance of our PS1 classifier on data from the Foundation Supernova Survey and compare the classifier's performance to results from other classification works. Finally, we compare the performance of our ZTF classifier built on simulations with the one built on real data.

4.5.1 Pan-STARRS classifier

As detailed in Section 4.3.1, we build a classifier using the collection of 557 spectroscopically-confirmed supernovae collected and processed by V20. We split the dataset by using 80% of the transients to train the classifier and 20% as the validation set to evaluate the classifier's performance. We augmented the training set using the method outlined in Section 4.4.1 to obtain approximately 5000 transients from each of the transient classes (SNII, SNIIn, SNIa, SNIbc, and SLSN). We prepared the training set using the same methods detailed in Sections 2.3.4-2.3.5. After first evaluating the performance of our classifier on the validation set, we then ran our classifier (trained on only the small spectroscopically-confirmed set) on the unlabelled photometric sample of 2315 transients. We then compare our performance on the

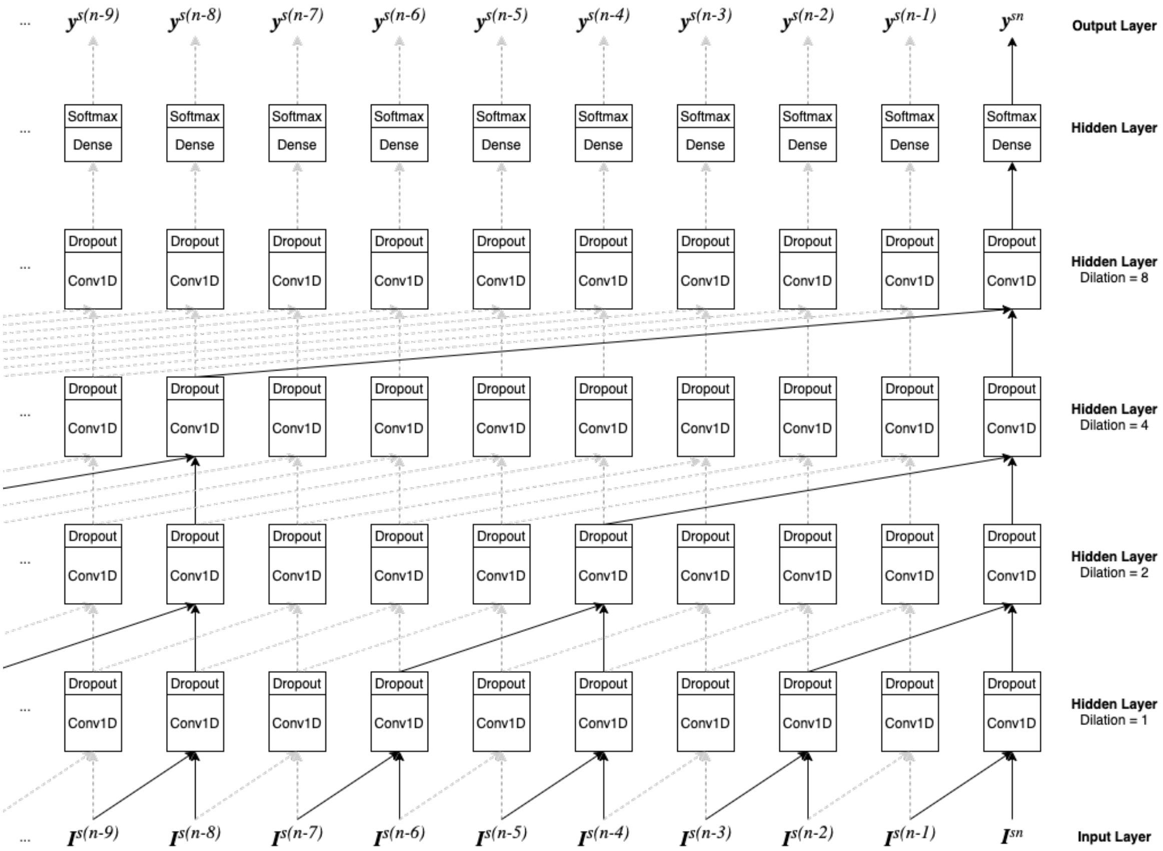


Fig. 4.9 Temporal Convolutional Neural Network architecture used for the new transient classifier and representing the model described in Section 2.4 and equation 2.5. Each column in the diagram is one of the n time-steps of the processed light curve with subsequent time-step from left to right. The bottom row is the input light curve matrix $\boldsymbol{I}$ for transient s up to a time-step t and consists of the light curve fluxes in all passbands, the host redshift, and the Milky Way reddening. The input light curve information of two adjacent time-steps are passed into a residual block consisting of a 1D Convolutional Neural Network Layer (Conv1D) with dropout. While not shown in the figure, the residual block also contains a second Conv1D layer with dropout. The outputs of these are then convolved with the outputs from some previous time-steps in the above hidden layers as shown in the diagram, until the Dense and Softmax layers. The fully-connected (dense) layer with a softmax regression activation function is applied at the final layer to compute the probability of each class at each time-step. The output layer is the predicted class probabilities $\boldsymbol{y}$ for transient s at time-step t. The solid arrows show how the predicted class probabilities $\boldsymbol{y}^{sn}$ are computed, and the gray dashed arrows show the neural network layers that lead to all other predictions. The network is causal, whereby new predictions only use information from previous time-steps in the light curve. We set the dropout rate to 20% for all layers in the network. We build this model using the `Keras` and `TensorFlow` libraries after adapting the TCN model from Bai et al. (2018) and their code in https://github.com/philipperemy/keras-tcn.

spectroscopic sample and the photometric sample with the performance of classifiers by V20 and H20.

4.5.1.1 Performance on PS1 spectroscopically-confirmed validation set

We tested the performance of our classifier on the small validation set consisting of 20% of the spectroscopically-confirmed PS1 dataset (which consisted of 111 transients across the five classes). In Figure 4.10 we plot the confusion matrices showing the accuracy of our classifications at early times just after trigger (1 day since trigger) and after the full light curve was observed (65 days since trigger). The classification performance is excellent, and correctly predicts the class of most supernovae in the validation set. Near trigger, most of the misclassifications of the II and IIn classes are to the other core-collapse classes. Thus, aggregating the core-collapse classes (II, IIn and Ibc), as if often done in other classification works, would provide nearly perfect classifications. At such early times, the SNIbc transients are often misclassified as Pre-explosion, but after a few more days of observations in the light curve, this quickly improves.

As discussed in chapter 2.5.4, the Receiver Operating Characteristic Curve (ROC) that plots the True Positive Rate against the False Positive Rate for a range of different threshold classification probabilities can often provide a more detailed understanding of the classifier's performance. In particular the area under the ROC curve (AUC) provides a good summary comparison statistic where AUC values close to 1 represent better classification performances. In Figure 4.12, we summarise the ROC curves at all times by plotting the AUC against time since trigger for each class. The Figure illustrates that the classifier has very good performance, with AUC values surpassing 0.95 for most classes, even at very early times. The SNIIn class has a slightly poorer classification performance because it is often mistaken for the SNII class as these classes look very similar. There are only very few SLSN transients in the testing set, and they are perfectly classified at all times after trigger. We think that the fact that the redshift distribution of the observed spectroscopically-confirmed SLSNe is higher than the other transients, means that using redshift information makes it very easy to distinguish SLSNe.

4.5.1.2 Comparison with other PS1 classifiers

Recent works by V20 and H20 have built classifiers trained on this same spectroscopic sample of PS1 transients and thus provide a very good benchmark to assess the performance

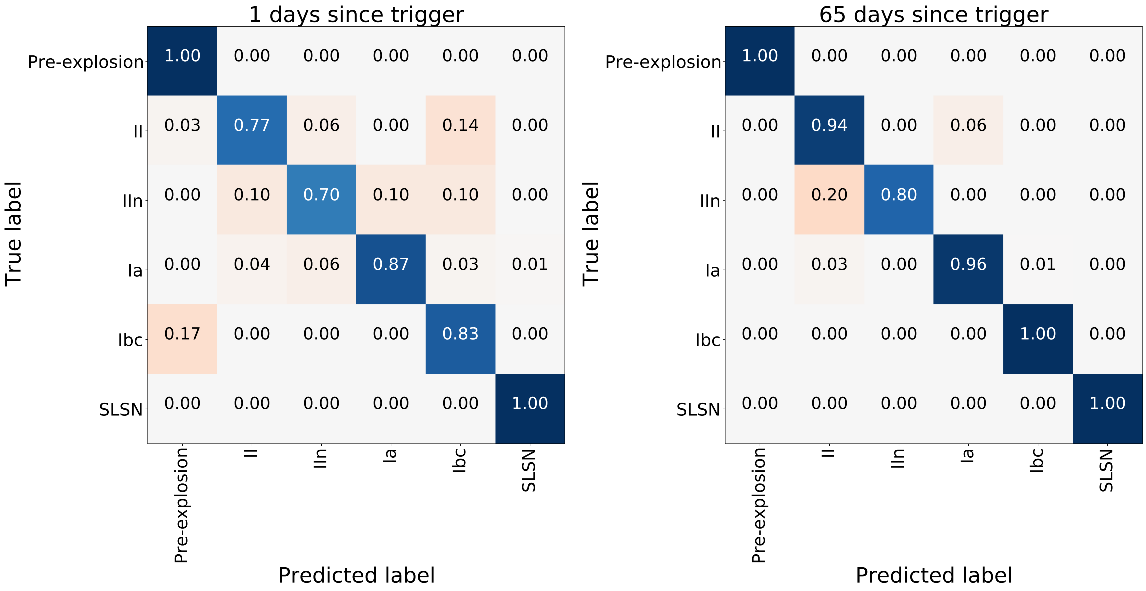


Fig. 4.10 Confusion matrix illustrating the performance of the PS1 classifier applied to the validation set consisting of 20% of the spectroscopically-confirmed dataset (111 transients). The left figure shows the results of the classifier on all transients in the validation set at 1 day since trigger, and the right figure shows the results of the classifier across the full light curve at 65 days after trigger.

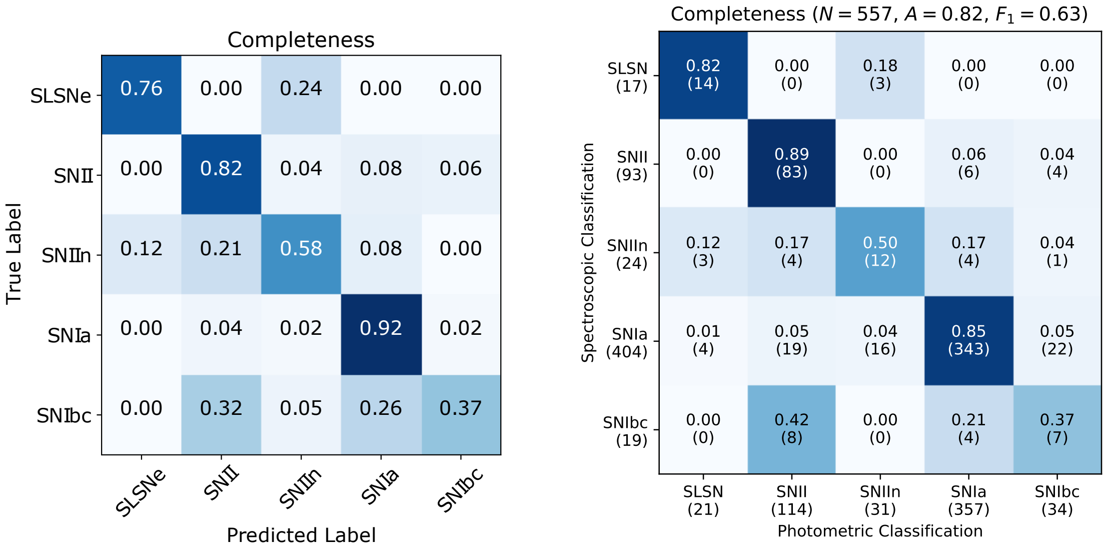


Fig. 4.11 Confusion matrices from Figure 6 of V20 (left) and Figure 6 of H20 (right) illustrating their classifier's performance on the 557 spectroscopically-confirmed PS1 transients. These plots can be compared with our performance on full light curves as shown on the right plot of Figure 4.10.

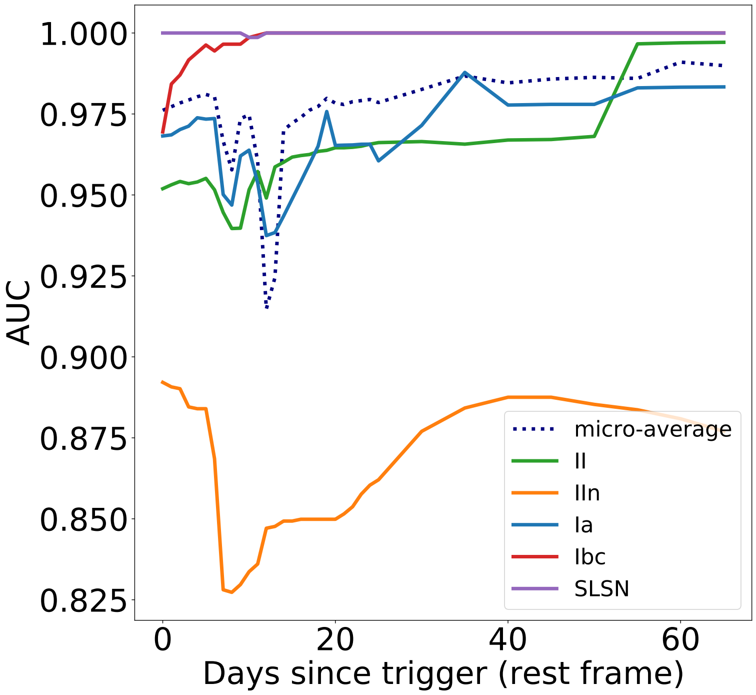


Fig. 4.12 The area under the ROC curve (AUC) vs time since trigger for the PS1 classifier applied to the testing set consisting of 20% of the spectroscopically-confirmed dataset. The ROC curves are not shown but the AUC for each class at every time-step are summarised by this figure. The micro-average curve is the average of the ROC curves in all classes weighted by the number of objects in each class.

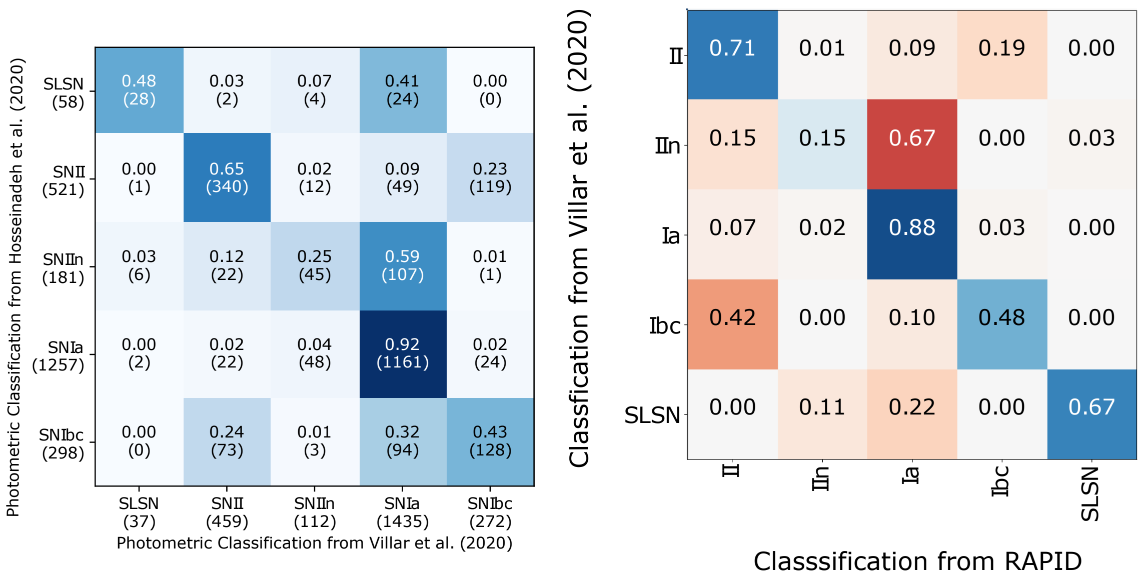


Fig. 4.13 Confusion matrices illustrating the classifications of the 2315 unlabelled photometric sample of transients from the PS1 dataset and the classification correlations between different classifiers. The left plot illustrates the classifications of H20 against V20 (as taken from Figure 10 of H20) and the right plot illustrates the classifications from this work against the classifications provided by V20. Since H20 against V20 use the full light curve for providing a classification, we use the RAPID classifications over the full light curve (at 65 days after trigger) in the plot.

of our new RAPID classifier. In Figure 4.11, we show the confusion matrices on the 557 spectroscopically-confirmed PS1 transients from Figure 6 of V20 and Figure 6 of H20 as a direct comparison to our late-time confusion matrix on the right of Figure 4.10. Unlike our confusion matrix that only tests the performance on 111 transients in our validation set, both authors use leave-one-out cross-validation to test the performance against all 557 transients. This requires that they build 557 classifiers each trained on 556 transients and classify the performance of the remaining transient. While this method can provide an accurate estimate of the model performance, it is far too computationally intensive for us to train this many classifiers. Therefore, while not an exactly fair comparison, our RAPID classifier appears to perform much better than both V20 and H20. Furthermore, unlike RAPID, neither V20 nor H20 are real-time classifiers, and thus they can only provide classifications once the full light curve has been observed.

V20 and H20 go on to apply their classifier trained on spectroscopically-confirmed supernovae onto their photometric sample of unlabelled supernovae. Since the true label is unknown for these transients, we can only compare our classifications with theirs and examine how the classifications between each of the three classifiers are correlated. On the left plot of Figure 4.13 we reproduce the plot from Figure 10 of H20 that compares their classifications to the photometric sample of 2315 unlabelled supernovae with the classifications from V20. Similarly, on the right plot of Figure 4.13, we plot the confusion matrix showing the correlations of classifications between RAPID and V20 on the photometric sample. While all three classifiers agree on the majority of transients, there is a lot of disagreement. For example, RAPID agrees with 88% of the SNe Ia classifications, but tends to classify many more transients as SNe Ia compared to V20. The difference in performance between the classifiers hints that they are using different information in the light curves to make their classifications.

From Figures 4.10 and 4.11, we know that RAPID is a more accurate classifier on all classes. However, in Hložek et al. (2020), they showed that combining the results of the top classifiers submitted to PLAsTiCC resulted in a much better performance than any single classifier. Therefore in future work, it might be beneficial to build an ensemble of the three classifiers that may outperform any individual classifier.

4.5.1.3 Applications to the Foundation cosmological sample using PS1 classifier

The Foundation Supernova Survey (Foley et al., 2018b) is a survey on the Pan-STARRS telescope that aims to provide a large sample of SNe Ia for cosmology. The growing tension between the early-time and late-time universe measurements of H_0 (see Section 1.1.1.1) has led to a deeper analysis of the systematic uncertainties in SNIa cosmology. Currently, the largest systematic uncertainty of SNIa cosmology is the calibration of the low-redshift SNIa sample (e.g. Betoule et al., 2014a; Jones et al., 2019; Rest et al., 2014; Scolnic et al., 2014; Sullivan et al., 2011). To this end, the Foundation Supernova Survey has tried to provide a large sample of precisely calibrated low-redshift SNe Ia. The Foundation Supernova Survey First Data Release (Foley et al., 2018b) has obtained a set of 180 SNe Ia that are useful for cosmology. These have each been observed using the Pan-STARRS telescope and are well sampled over the full duration of the light curves. We note that Foundation is a follow-up survey and that all of the supernovae have been spectroscopically confirmed. In this section, we run our classifier, that uses only photometry, to see what fraction of the cosmological sample we are able to recover without using any spectroscopy.

RAPID classification	Count
SNIa	171 (95%)
SNIbc	3 (1.7%)
SNIIn	1 (0.6%)
SNII	0
SLSN	5 (2.8%)

Table 4.1 The 180 SNe Ia from the Foundation cosmological sample were classified by RAPID and the highest probability classification using the full light curve resulted in the shown class predictions.

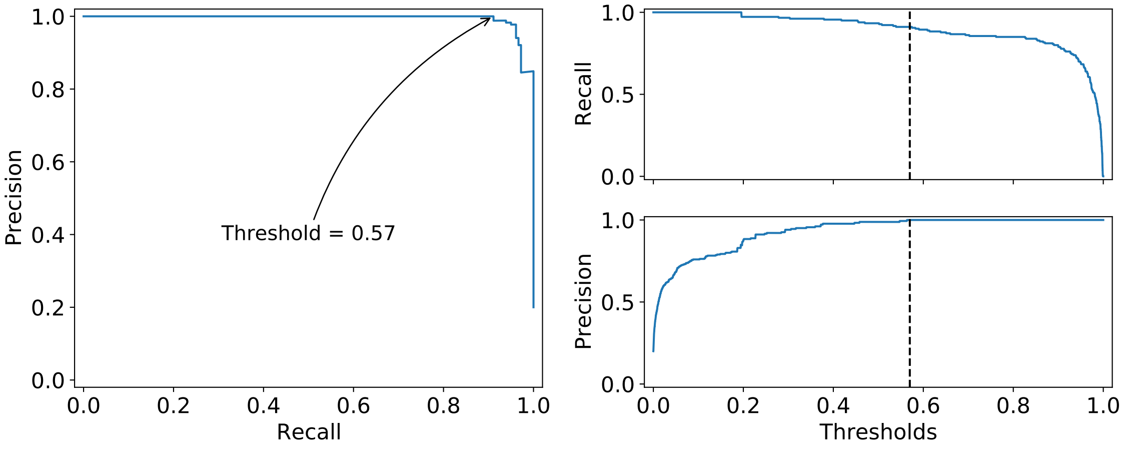


Fig. 4.14 The *precision* and *recall* of the SNe Ia in the Foundation cosmological sample plotted at different RAPID threshold probabilities. The Precision-Recall curve is plotted at different SNIa threshold probabilities in the left subfigure. The recall and precision are plotted against threshold in the right subfigures. An arrow and vertical dashed line is shown to indicate a threshold probability of 0.57 which is the highest recall (91% of SNe Ia) that gives perfect precision of SNe Ia.

We have classified the Foundation cosmological sample of SNe Ia as a function of time, but have recorded the RAPID classifications using the full light curve in this analysis. As illustrated in Table 4.1, 171 out of the 180 supernova were correctly classified as a SNIa. Three transients were misclassified as SNe Ibc, five were misclassified as SLSNe, and one was misclassified as a SNIIn. Thus, using only photometry without obtaining spectra, RAPID correctly identified 95% of the cosmological sample.

To instead obtain a high precision (pure) sample, we analysed the Precision-Recall curves at various thresholds. See Section 2.5.5 for a description of the Precision-Recall metric. The precision is a measure of the *purity* of the sample at a given threshold probability, while the recall is a measure of the *completeness* of the sample at a given threshold probability. We plot the precision and recall at various threshold probabilities in Figure 4.14.

For supernova cosmology, we need a pure sample of SNe Ia. Thus, if we examine the highest recall that still gives a precision of 1, we can identify a threshold probability at which to accept a RAPID SNIa classification. Using a threshold probability of 0.57, we obtain a perfect precision sample of SNe Ia, while still recalling 164 out of 180 SNe Ia (91%). We indicate this probability with an arrow and dashed vertical lines in Figure 4.14.

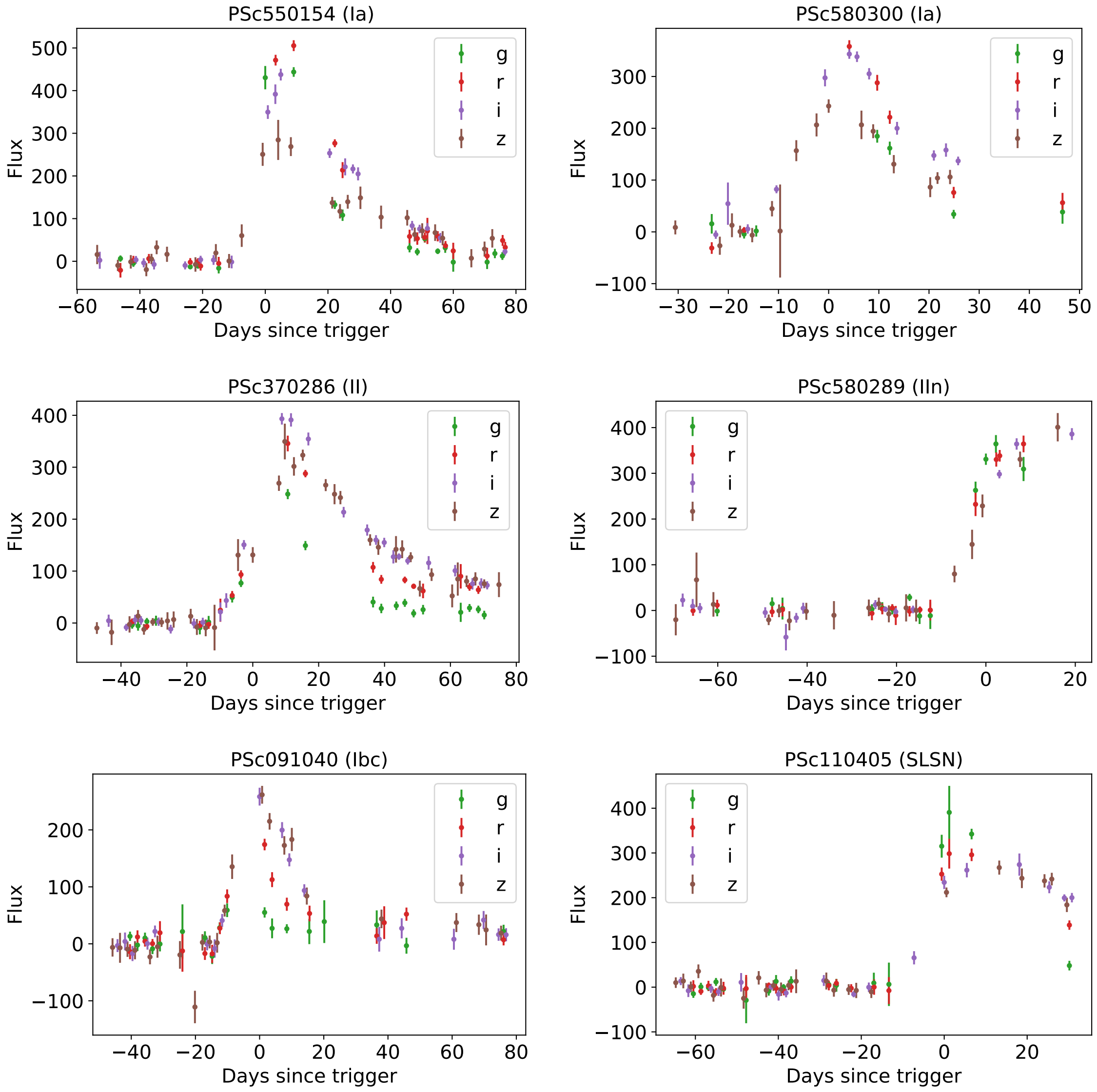
Therefore, we have shown that by using RAPID and setting a higher threshold probability at which to accept SNIa classifications, we can obtain a high-precision sample of SNe Ia useful for cosmology.

4.5.2 ZTF real data classifier

In the original version of RAPID detailed in Chapter 2, we only trained our classifier on ZTF light curve simulations. However, the performance of our classifier has had mixed performance when run on real data from the ZTF MSIP data stream. In this section, we train a classifier using just a collection of real data to compare against our classifier built from ZTF simulations. Due to the limited data available, we can only classify into a smaller number of classes, however.

We train a classifier using the collection of 2839 supernovae that we collected and detailed in Section 4.3.3. We split the dataset into a training set consisting of 80% of the dataset (2271 transients) and 20% for the testing set (568 transients). As illustrated in Figures 4.3 and 4.4, our collected dataset has a wide range of labels with some classes having very few objects. We group the transients into four classes: SNIa (consisting of all SNe Ia and subtypes: Ia, Ia91T, Ia91bg, Iacsm, Iapec, Iax), SNII (consisting of all SNe II and subtypes: II, IIL, IIP, IIn, IIpec), SNIbc (consisting of all SNe Ibc and subtypes: IIb, Ib, Ibn, Ic, IcBL, Ibc), and SLSN (consisting of the SLSNI and SLSNII objects). We augmented the training set using the method outlined in Section 4.4.1 to obtain approximately 5000 transients in each of these four transient classes. We prepared the training set using the same methods detailed in Sections 2.3.4-2.3.5. We trained the classifier on this augmented dataset and evaluated its performance on the 568 transients in the testing set.

In Figure 4.16, we plot the confusion matrices showing the accuracy of our real data ZTF classifier at early times just after trigger (1 day since trigger) and after the full light curve was


PSc550154 (Ia)

PSc580300 (Ia)

PSc370286 (II)

PSc580289 (IIn)

PSc091040 (Ibc)

PSc110405 (SLSN)

Fig. 4.15 Example Pan-STARRS light curves illustrating the multi-passband pre-trigger forced photometry. The examples all illustrate that before trigger, the light curves show the early rise in luminosity despite each data point being below the detection limit. The early light curve is very useful for classification and is only available with forced photometry, and is thus not currently available in the ZTF MSIP light curves. Each plot is titled with the supernova's Pan-STARRS identifier and the type in parentheses.

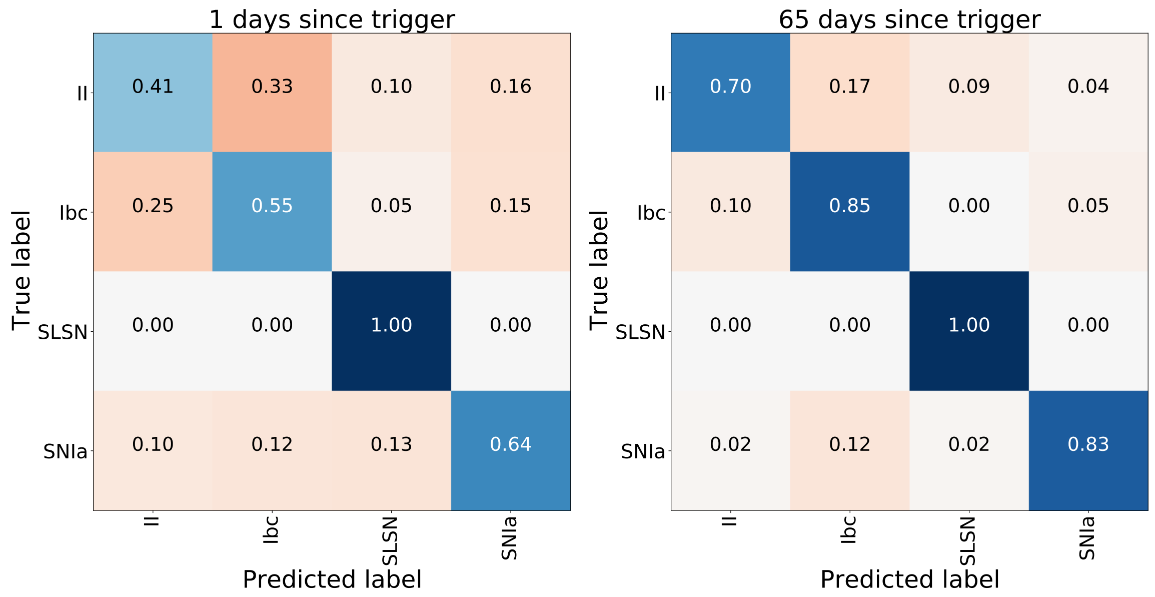


Fig. 4.16 Confusion matrices illustrating the performance of the RAPID classifier trained on the real ZTF dataset and applied to testing set consisting of 568 supernovae. The left figure shows the results of the classifier on all transients in the testing set at 1 day since trigger, and the right figure shows the results of the classifier across the full light curve at 65 days after trigger.

observed (65 days since trigger). The classification performance is very impressive at late times, correctly classifying over 80% of transients. However, at early times the performance is not as good as the PS1 classifier, most likely because there is no pre-trigger photometry available in the light curves. Unlike the PS1 dataset (see figure 4.15), the ZTF MSIP alert packets do not provide forced photometry and only provide upper limit magnitudes for the observations that were not significant enough to trigger an alert. It is for this reason that the classifier does not learn anything about a pre-explosion class.

The SNII and SNIbc classes are mostly misclassified with each other, which is not surprising given that these two classes look similar and are often aggregated into a single class by previous works. In Figure 4.16, it is clear that if we combine the SNII and SNIbc classes, we achieve much better classification accuracies, particularly at early times (left plot) when it is more difficult to distinguish between these classes. We achieve perfect classification of SLSNe because the few SLSNe in our sample occur at higher redshifts than the rest of the sample.

In Figure 4.17, we summarise the ROC curves at all times by plotting the AUC as a function of time for each class. Even at the time of trigger, all classes have an AUC of at

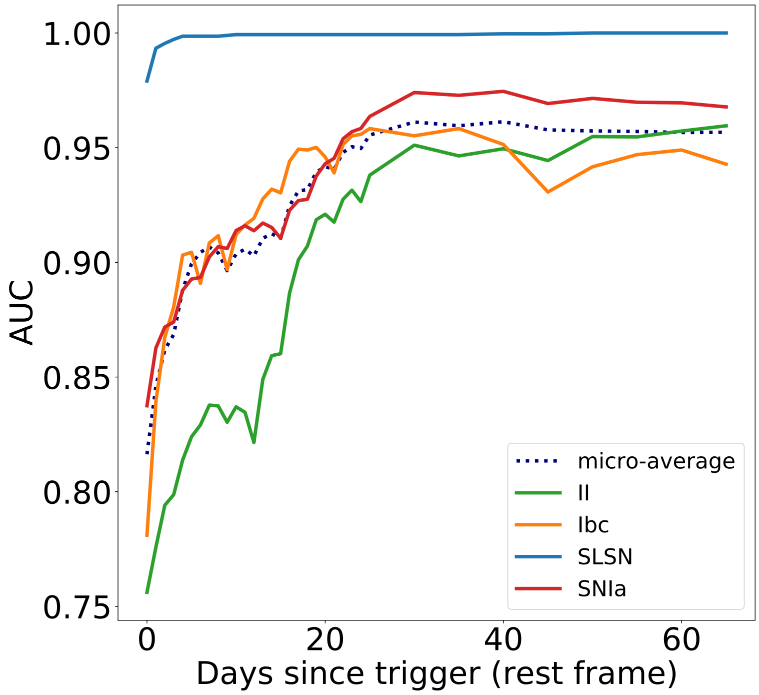


Fig. 4.17 The area under the ROC curve (AUC) vs time since trigger for the ZTF real data classifier applied to the testing set consisting of 568 supernovae (20% of the dataset). The ROC curves are not shown but the AUC for each class at every time-step are summarised by this figure. The micro-average curve is the average of the ROC curves in all classes weighted by the number of objects in each class.

least 0.75 with the micro-averaged[7] AUC across all classes starting at 0.82. At this time, the classifier is able to reasonably distinguish the classes using only the available redshift information and one set of photometric observations. The SLSNe achieve a near perfect AUC across all times mainly because their high redshift easily distinguishes them from the other classes. The AUC of the other supernova classes increases quickly until about 25-30 days. At this time, most light curves are well past peak luminosity and are dimming closer to background brightnesses.

4.5.3 ZTF simulations classifier

In this section, we improve upon the classifier built for the original version of RAPID discussed in Chapter 2. Making several improvements to the simulations (as described in Section 4.3.2), and improving the upon the architecture (as detailed in Section 4.4), we train a new classifier that should ideally perform better on real data. After simulating $\sim 10,000$ ZTF transients in each of the following classes: SNIa, SNII, SNIbc, SNIa-91bg, SNIa-x, Kilonova, and SLSN-I, we trained a new classification model. We split the dataset into a training set, consisting of 80% of the dataset, and a validation set, consisting of 20% of the dataset. We evaluate its performance using the same performance metrics used previously. To evaluate the performance of this classifier on real data, we also test the classifier on the collection of 2839 real ZTF transients collected and described in Section 4.3.3. We can compare the performance of our classifier against the one we trained on real data shown in the previous subsection 4.5.2.

4.5.3.1 Performance on simulated ZTF testing set

We first validated the performance of our classifier on 20% of our simulated dataset, consisting of ~ 2000 transients from each of the seven classes. In Figure 4.18 we plot the confusion matrices showing the accuracy of our classifications at early times just after trigger (1 day since trigger) and after the full light curve was observed (65 days since trigger). The classification performance has significantly improved for most classes when compared to the original RAPID in Muthukrishna et al. (2019a) and shown in Figure 2.7 of chapter 2. Most notable, is the significant improvement on the core-collapse supernovae (SNII and SNIbc). While this can partly be explained by the absence of the CART class that is known

[7] The micro-averaged ROC curve aggregates the results from all objects from all classes. It is effectively the average of the ROC curves from all classes weighted by the number of objects in each class.

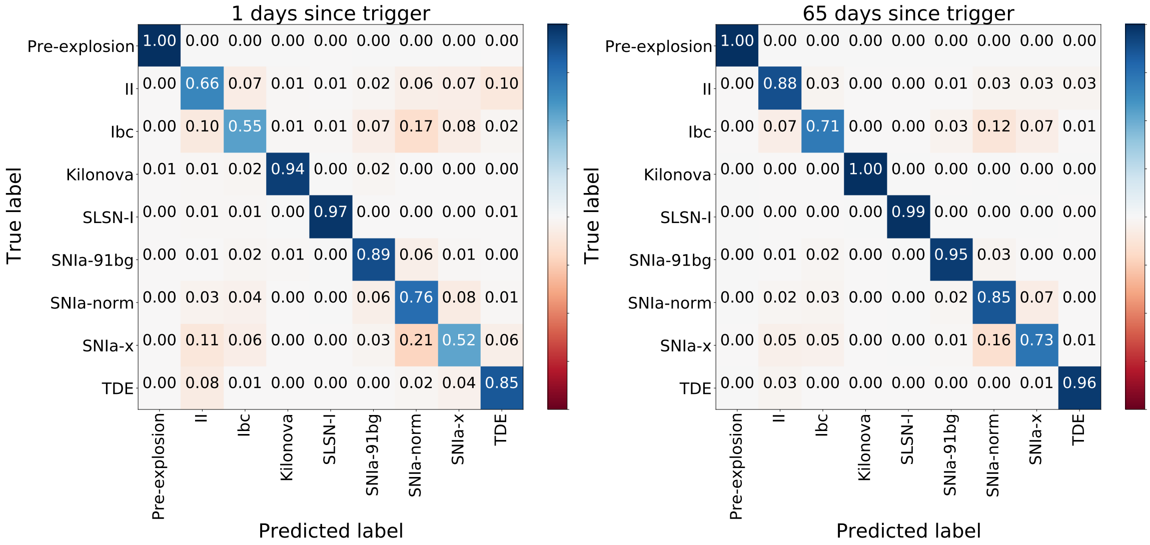


Fig. 4.18 Confusion matrices illustrating the performance of our classifier trained on our new ZTF simulations and tested on the validation set consisting of 20% of the simulated dataset. The left figure show the results of the classifier near detection (1 day after trigger) and the right figure shows the results at late times (65 days after trigger).

to look quite similar to CCSNe (see Figure 2.8), it is most probably due to the improved simulations that use more realistic models from Vincenzi et al. (2019). The fewer classes to classify between may also contribute to the improved performance. At late times, the SNIa and SNIa-x classes are mostly misclassified with each other, as is to be expected as their light curves look very similar. SLSNe have excellent classification performance, even at early times. With redshift information, it is often easy to distinguish SLSNe as they are much brighter than the other classes. The ease of classifying kilonovae might be explained by their very unique light curve shape when compared to other supernovae; and perhaps the Bulla models (Bulla, 2019) are more obviously different from other transients than the Kasen models (Kasen et al., 2017) used in the previous version of RAPID.

Similarly to the other sections, in Figure 4.19, we plot the AUC as a function of time since trigger to assess the performance of classifying each class at different times. After trigger, the AUC is above 0.9 for all classes, indicating that our classifier is very effective. The performance improves rapidly for all classes until around 10 days past trigger. At this point, most transients have just passed peak brightness, and the time and luminosity of peak is key information that allows RAPID to quickly discriminate between different classes.

At around 15 days past trigger, the SNII and SNIbc classes have a sudden drop in performance before they gradually rise again. The fact that it's only these two classes that

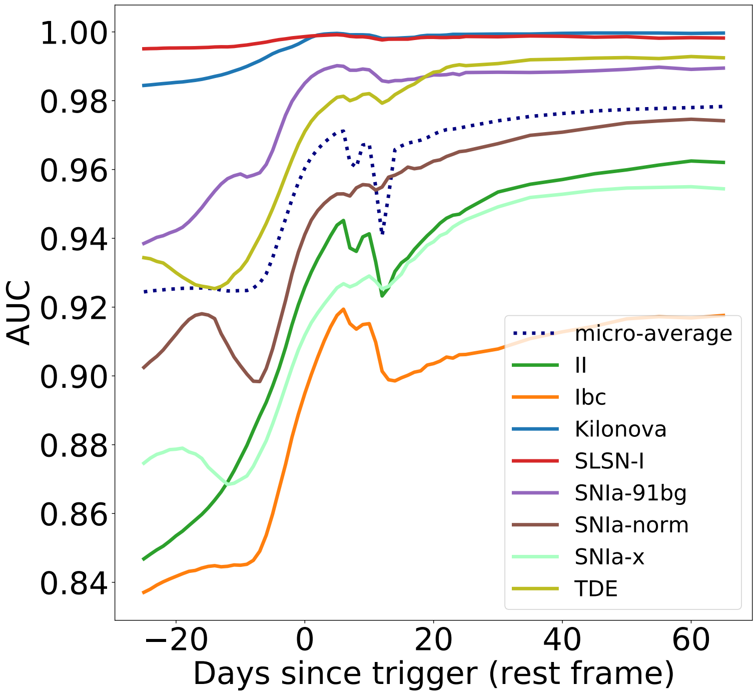


Fig. 4.19 The area under the ROC curve (AUC) vs time since trigger for the classifier trained on ZTF simulations and applied to the testing set consisting of 20% of the dataset. The ROC curves are not shown but the AUC for each class at every time-step are summarised by this figure. The micro-average curve is the average of the ROC curves in all classes weighted by the number of objects in each class.

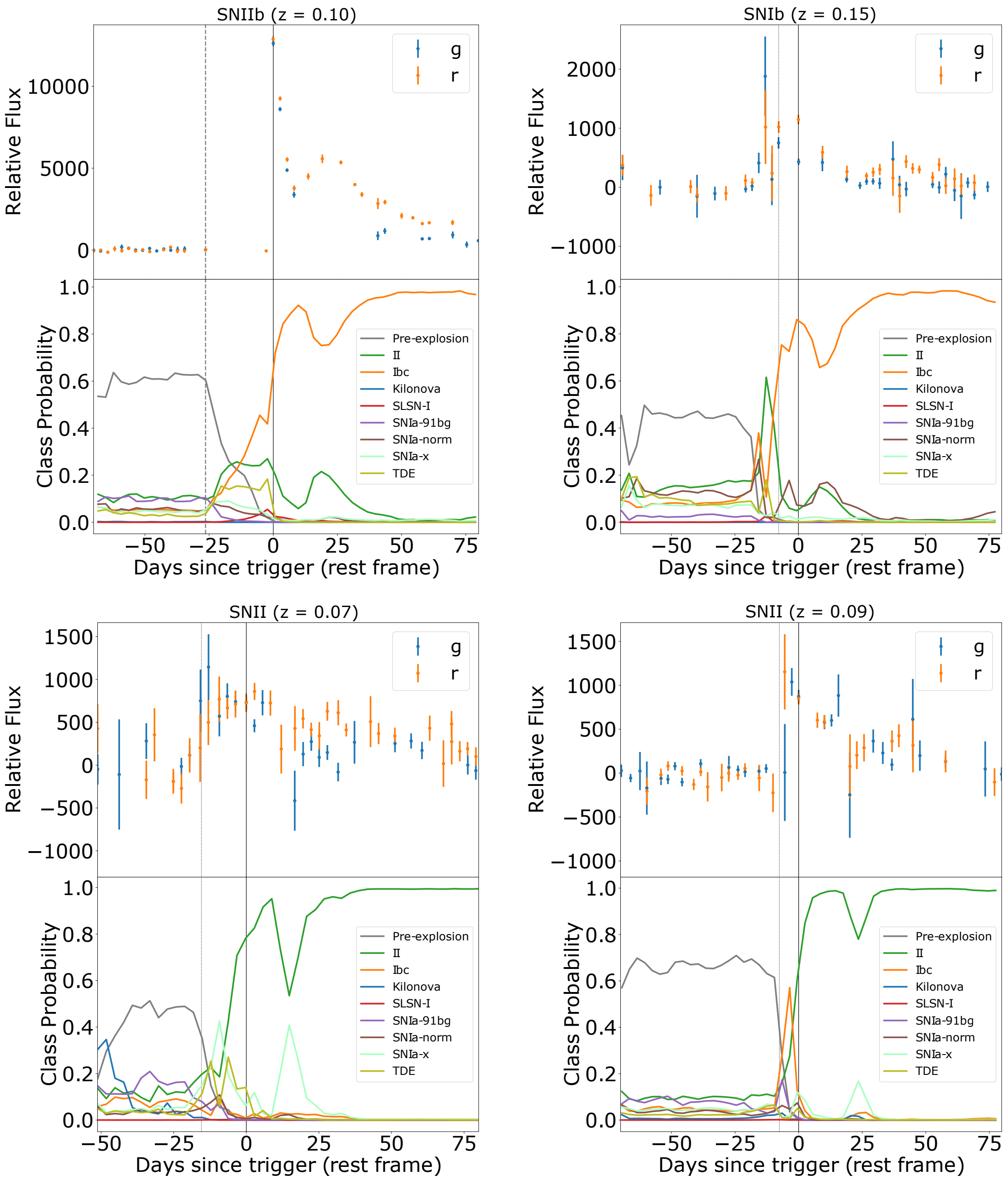


Fig. 4.20 Example classifications of simulated ZTF Type II and Ibc supernovae. These light curves and classifications were selected because they are examples of cases where there is a slight drop in the true class' classification probability at around 15 days after trigger.

drop indicates that most misclassifications are occurring between these two classes. Given that they are both CCSNe, and many authors choose to aggregate them into one class, this is not often a big problem for most astronomers. However, it is interesting, because it points to the light curves of these classes having some key difference at this time. Initially, I suspected that this drop might be explained by Type IIb supernovae. It is well known that SNe IIb initially appear similar to SNe II, but a few weeks after their explosion, they lose their Hydrogen envelope and they begin to appear more like SNe Ib (Yoon et al., 2017). As described in Yoon et al. (2017), the very small amount of hydrogen in SNe IIb causes this shift.

However, after looking at examples of classified transient light curves, we noticed that the classification probabilities of many SNe II and SNe Ibc dropped slightly around 15 days after trigger which is usually shortly after peak luminosity. We sifted through over a thousand SNe II and Ibc to identify causes of the drop in class probability and have plotted a few examples of this drop in four light curves in Figure 4.20. The drop occurred for many different subtypes of type II and Ibc supernovae, and the cause of the drop was not obvious. However, in many examples, the drop in probability coincided with a slight increasing or decreasing bump in flux at around 15 to 25 days past trigger. In the SNIIb example in Figure 4.20, the bump in the *r* band aligns with the drop in classification probability, and in the two SNII examples, a drop in the *g* band causes the decrease in classification probabilities. In the SNIb event, there is no obvious cause. We could not come up with a conclusive reason for what information the classifier is using that causes this drop. However, in most cases, we suspect that a second rise or drop in magnitude in flux causes the classifier to be confused between classes.

In future work, we will analyse the misclassifications more deeply to discover what features in the SNII and SNIbc classes are causing this issue. However, given a high enough threshold classification probability, this bump should not be cause for much concern.

4.5.3.2 Performance on real data from ZTF

After publishing RAPID in Muthukrishna et al. (2019a), we noticed that there were many issues with our classifier when running on real data compared to running on simulations. We attribute these problems to a range of issues that were discussed in Sections 4.3.2 - 4.3.4, but have hopefully been improved by our new data. In this subsection, we run our classifier (that was trained on simulations) on our collection of 2839 real transients observed by ZTF and detailed in Section 4.3.3 with the distribution of transients among the labelled classes shown

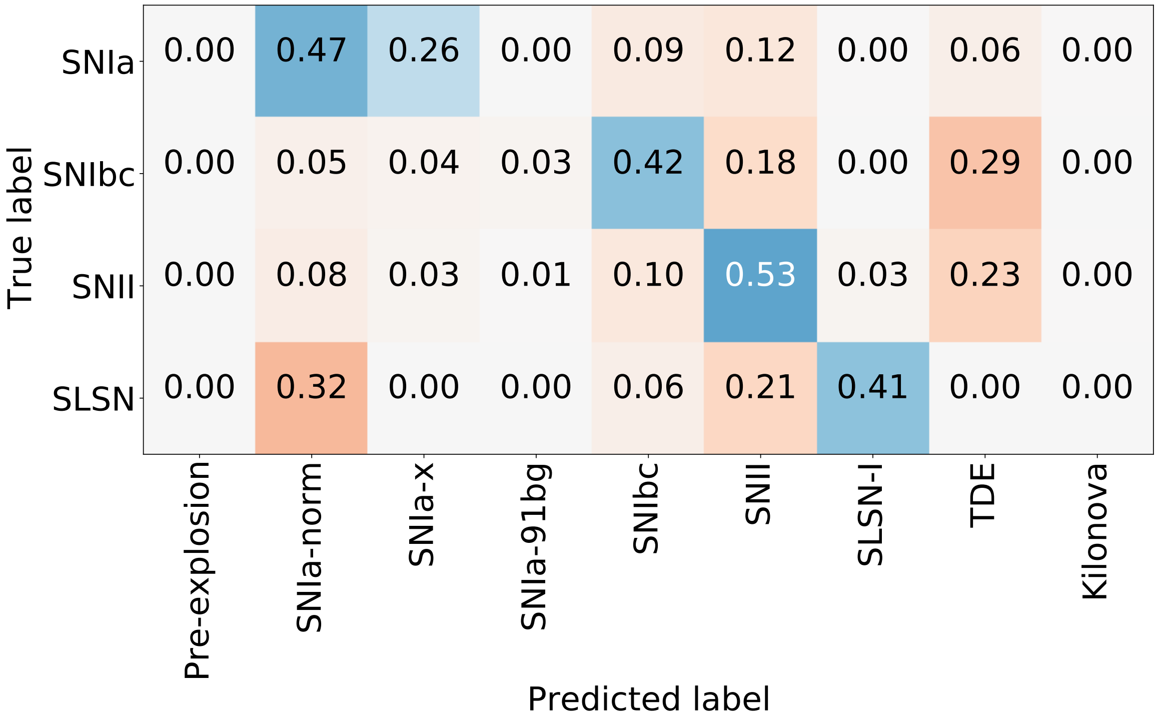


Fig. 4.21 Confusion matrix for the classifier trained on ZTF simulations and applied to the collection of real ZTF transients. The confusion matrix shows classifications at 65 days past trigger. We show the broad supernova class label taken from the Open Supernova Catalog as labelled by a range of different astronomers/collaborations on the vertical axis, and show RAPID's predicted label on the horizontal axis. Misclassifications are shown in red while "good" classifications are shown in blue. We group all predictions that are of the same broad supernova type as a 'good' classification in the plot. The number of transients from each class that are being classified are shown in Figure 4.4.

in Figure 4.4. We only record the classification of the full light curve (at 65 days after trigger). In Figure 4.21, we show the confusion matrix indicating the broad supernova class taken from the Open Supernova Catalog (as labelled by a range of different astronomers/collaborations) on the vertical axis, and RAPID's predicted classification on the horizontal axis. We show misclassifications in shades of red and classifications that are of the same broad type of supernova in shades of blue.

Overall, 64% of the transients in the dataset were correctly classified as its broad supernova type. The majority of transients in the real dataset are labelled as 'Ia' on the OSC (consisting of 70% of the dataset). The classifier based on simulations correctly classified 73% of these as a type of SNIa (SNIa-norm or SNIa-x). The other SNIa subtypes are more commonly misclassified as CCSNe. The second most populous class were the transients labelled as a 'SNII' subtype on the OSC consisting of 351 transients (12% of the dataset). Only 53% of these were correctly classified as SNe II. While there were no TDEs in the dataset, a large fraction of the real transients were incorrectly classified as TDEs. The overall classification performance is much poorer than the classifier that was trained on real data (shown in section 4.5.2).

The poorer performance here highlights issues with using classifiers trained on simulations and applying them to real data. There are remaining issues in simulating transients that match the properties of transients observed in real datasets and we discussed some of these within section 4.3.4. The fact that the PLAsTiCC ZTF-like simulations include photometry before maximum whereas the real data only include upper limits may be a large factor for why we obtain good classification performance on simulations but not on real data. We will pass on the analysis in this chapter to the PLAsTiCC version 2 team to help identify methods of improving simulations to better match reality. However, despite the advantages of using simulated training sets, until simulations improve, classifiers trained on real data will provide more reliable performance and may be required for ongoing transient brokers.

4.6 Effect of survey observing strategy on classification performance

4.6.1 Effect of colour on classification performance

So far in this chapter, we have trained RAPID on different datasets to evaluate its performance on real data. The slightly different observing strategies between PS1 and ZTF makes an impact on our classification performance. For example, the extra *iz* passbands in PS1 compared to ZTF with a similar cadence likely leads to better classification performance.

In this section, we examine the effect of additional passbands on classification performance. The PS1 real dataset is very small and thus has slightly inconsistent performance each time we train the classifier because it is highly dependent on which transients are used in the training and testing sets. Instead, we choose to use PS1 simulations to test the effect of colour on RAPID's performance.

Recently, the Young Supernova Experiment (YSE) collaboration has begun training a new version of RAPID for use on their survey using the Pan-STARRS telescopes. To do this, they have used the PLAsTiCC modelling software to simulate transients that match

	micro-average	SNIa-norm	SNII	SNIIn	SNIbc	SLSN-I
gr	(0.84, 0.91)	(0.73, 0.83)	(0.77, 0.86)	(0.75, 0.82)	(0.78, 0.84)	(0.84, 0.92)
gri	(0.86, 0.92)	(0.80, 0.87)	(0.80, 0.88)	(0.78, 0.82)	(0.83, 0.88)	(0.87, 0.93)
griz	(0.86, 0.93)	(0.80, 0.88)	(0.81, 0.90)	(0.78, 0.84)	(0.83, 0.88)	(0.87, 0.93)

Table 4.2 A summary of the AUC scores from Figure 4.22 showing how the AUC scores change for classifiers trained on different passbands. In the brackets of each cell we state the AUC score at trigger and at late times (65 days past trigger) in the first and second element of the brackets, respectively.

the observing properties of Pan-STARRS. Patrick Aleo, a PhD student at the University of Illinois Urbana-Champaign working in Professor Gautham Narayan's group, is working to improve these simulations to match real Pan-STARRS observations. He has simulated transients from five different classes: SNIa-norm, SNII, SNIIn, SNIbc, and SLSN-I.

In this section, I am using the PLAsTiCC-like Pan-STARRS simulations developed by Patrick and the YSE collaboration to train three versions of RAPID: one trained on PS1 simulations using all *griz* passbands, a second using only *gri* passbands, and the third using only *gr* passbands. To only examine the effect of the passbands, I do not provide any redshift information to the classifiers. I have plotted the AUC as a function of time for each of these classifiers in Figure 4.22. In Table 4.2 I summarise the results of these figures as a table showing the AUC scores at trigger and at late times (65 days after trigger).

The classifications improves significantly after the addition of *i* band compared to the classifier trained on only the *gr* passbands. The classifier trained on the *gr* bands has a micro-averaged AUC of 0.84 and 0.91 at trigger and at late times, respectively. Whereas, the classifier trained using *gri* bands has a micro-averaged AUC of 0.86 and 0.92 at trigger and late times, respectively. While most classes improve due to the additional *i*-band information, the SNIa-norm class has the most significant improvement, increasing from 0.73 to 0.80 at trigger and increasing from 0.83 to 0.87 at late times. The increase is also much more rapid shortly after trigger when the *i* and *z* bands are included before increasing at a slower rate after around 6 days past trigger. Around this time is usually where maximum brightness occurs for SNe Ia and indicates that the near-infrared (NIR) bands add significant information in the early phase of SNe Ia. A similar trend can be seen for the SNIbc classes. The addition of *z* band only makes a small improvement on the classification ability of most classes compared to when *i* band was added. At early times, the addition of *z* band makes a negligible impact

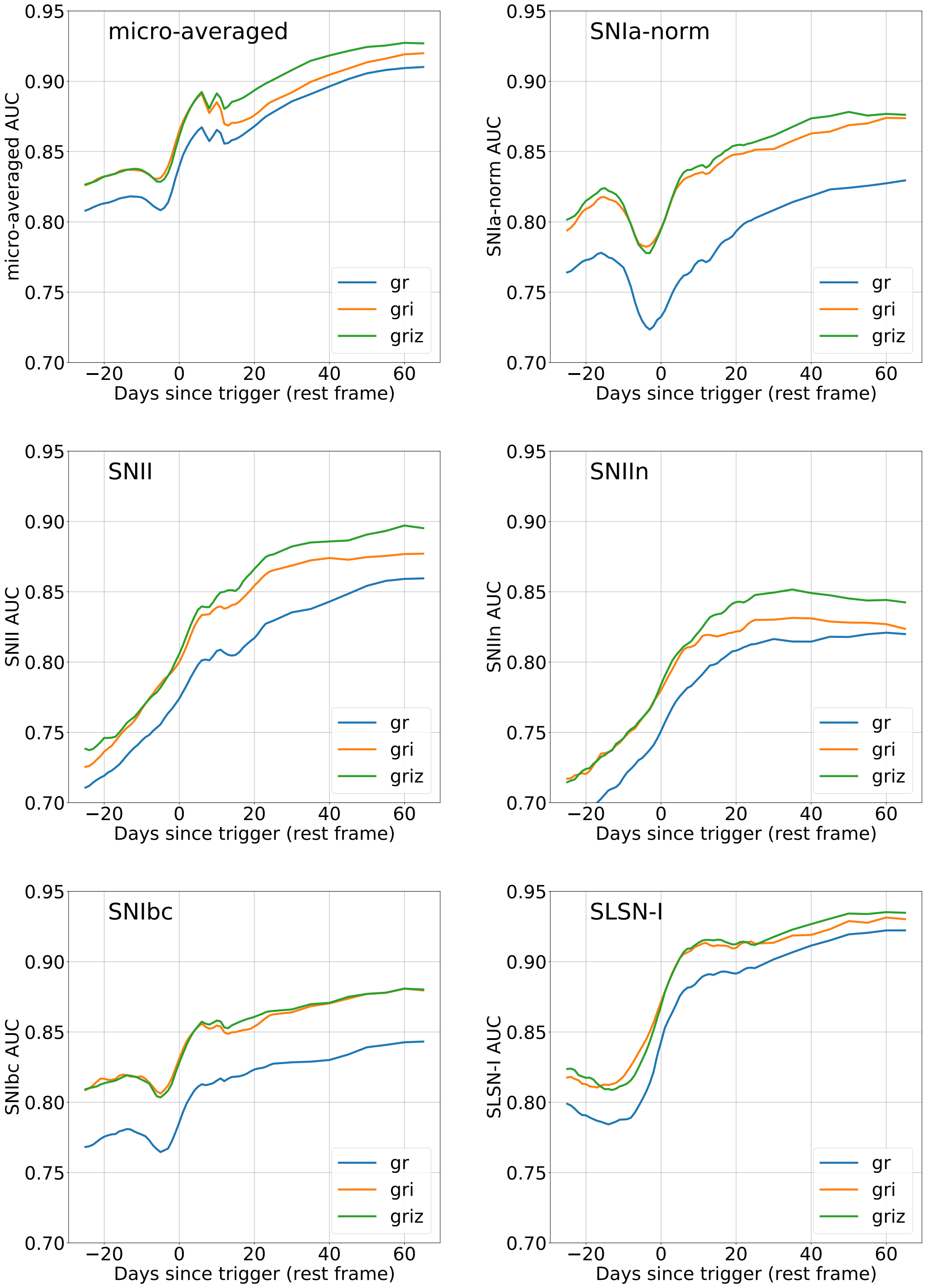


Fig. 4.22 The effect of passbands on classification performance. The area under the ROC curve (AUC) vs time since trigger (in the observer frame, since redshift is not used) for three classifiers trained on Pan-STARRS simulations. In each plot we show the performance of the three classifiers that use only *gr* passbands (blue), only *gri* passbands (orange), and all four passbands *griz* (green) for each class. Unlike the previous classifiers shown in this chapter, none of the classifiers use redshift information. The classification performances improve (obtain higher AUCs) with more passband information. The first plot compares the micro-averaged AUCs (that is the weighted-average AUC across all classes).

on classification performance, and it is only at late times that the SNII and SNIIn classes have a more significant improvement in performance (while the other classes have very little improvement).

To better understand how the presence of i and z bands improves the classification performance, we plot the median colours as a function of time since trigger for each class in Figure 4.23 with the median absolute deviations (MAD) as the shaded regions. We use the median and MAD instead of the mean and standard deviation as these statistics are more robust to outliers in the dataset. We compute the colours by first converting the fluxes to magnitudes and taking the difference between the magnitudes in different passbands.

We note that we have corrected all light curves for Milky Way extinction before making these colour curves (see section 2.3.4). Thus, the plots represent the evolution of colour for each transient class as caused by the intrinsic colour variation of each transient and the simulated host galaxy dust.

The population of transients in each class display a wide distribution of colours as illustrated by the large shaded regions in Figure 4.23 and show significant overlap among the classes on all colours. This highlights that classifying light curves based on colours alone is not possible and thus reinforces the need for complex architectures such as deep learning for classification. Nevertheless, while it is not possible to ascertain which light curve features our neural network is using to classify between transient classes, some patterns in the colour curves can be seen that may hint at what our algorithm might be using to classify between transients.

The distribution of the SNIbc colour curves are the reddest (most positive) of all the classes and indicate that their host galaxies may have the most extinction. Previous work such as van Dyk et al. (1996), Kelly et al. (2008), and Drout et al. (2011) have shown that SNe Ibc typically reside in dusty star forming regions. This may explain why the SNe Ibc light curves are the reddest, particularly because they have the reddest colours even before the transient phase of the light curve (well before trigger). We also notice that the SLSNe have the bluest (least positive) colours. This can be explained by the fact that SLSNe reside in very faint host galaxies and are known to have very blue spectral energy distributions (SEDs) relative to other supernova classes (e.g. Lunnan et al., 2020; Quimby, 2012; Quimby et al., 2018). As illustrated in Figure 4.23, these differences in colour between CCSNe and the other classes, are more obvious in the colours that include i and z bands than the $g - r$ colours. Thus, these NIR bands may help our neural network classifier distinguish these classes better.

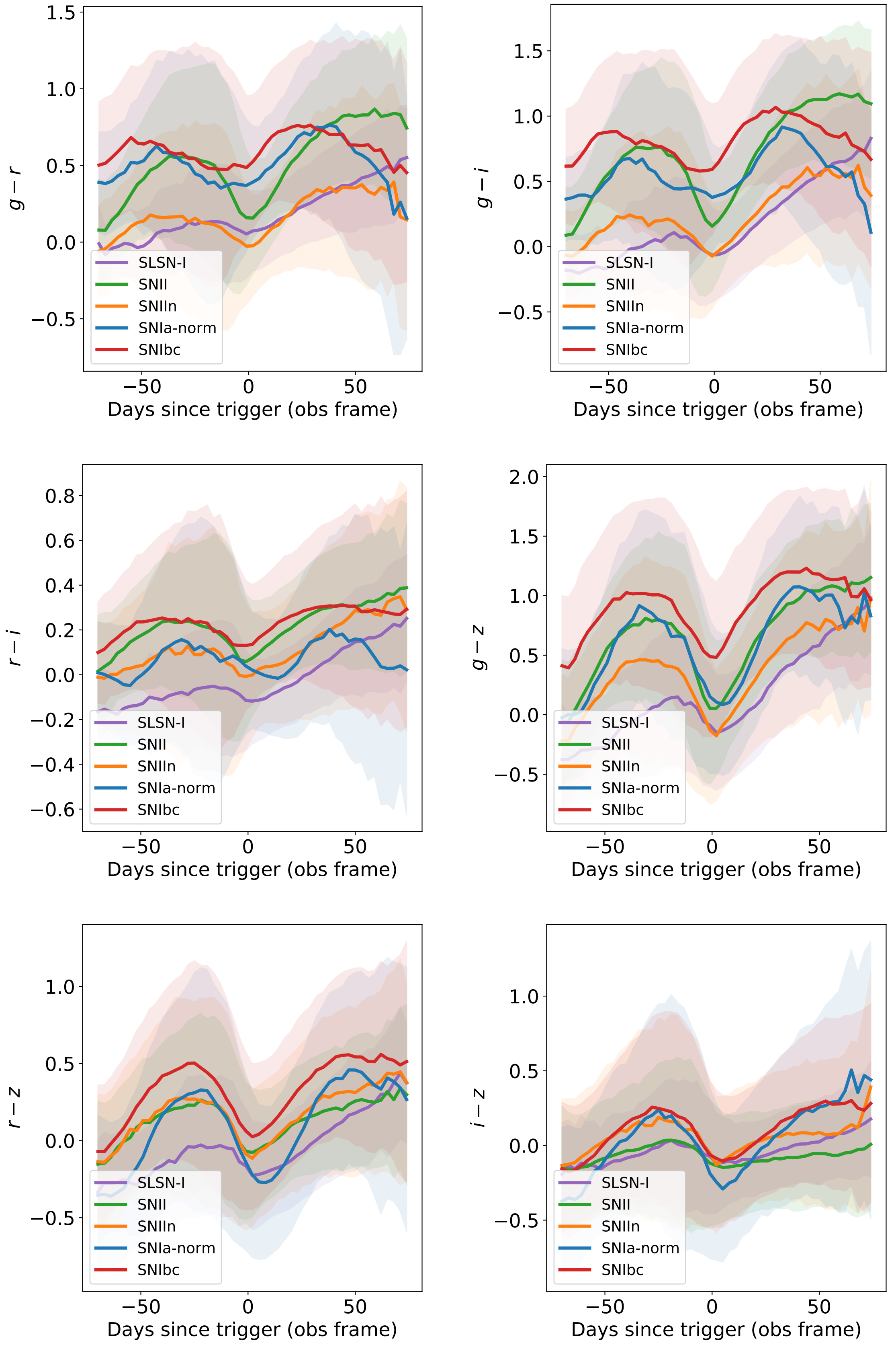


Fig. 4.23 Colour vs time since trigger (in the observing frame) for the Pan-STARRS simulations. We plot the median of colours for the populations of transients in the Pan-STARRS simulated dataset as the solid lines, and the median absolute deviations as the shaded regions.

Furthermore, the medians of the $g-r$, $g-i$, and $r-i$ colours all shows significant differences between the classes after trigger. The SLSNe and SNIIn look very similar in the $g-r$ and $g-i$ bands, but show marked differences in the $r-i$ bands, indicating that the i band is important to better distinguish these classes. The median of the SNIa-norm curves in the $g-i$ and $r-i$ passbands are very different to the other curves just after trigger and explain how important the i band is to separating SNe Ia from the other classes.

However, the $i-z$ colour curves look nearly the same for all classes and highlight that the z band offers very little new information compared to the i band. Looking back at Table 4.2, it is clear that the classifier that includes *griz* bands only offers marginal improvement compared to the classifier trained on *gri* bands. In fact, at trigger, the difference is barely noticeable. At late times, the performance is only slightly improved. Thus, we can conclude that the inclusion of the z-band does not offer much additional information useful for classification.

This analysis is useful for surveys that are focused on early classification, such as YSE, as it may highlight that observation time on z-band might be better spent elsewhere (for example, on improving the cadence, exposure time, or sky-coverage) if early classification is the main objective.

4.6.2 LSST observing strategy

In the above section, we analysed the impact of colour on classification performance. In future work, we will analyse the effect of a range of survey observing strategies on classification performance. The upcoming LSST is currently at a critical stage of deciding the optimal observing strategy for the survey. Considerations such as the cadence, the number of visits to a field per night, the exposure time, dithering, and the footprint and area coverage of the survey are all under debate within the LSST community (LSST Science Collaborations et al., 2017). The LSST Dark Energy Science Collaboration have recently analysed the impact of the observing strategy on various cosmological constraints (Lochner et al., 2018, 2021), and many other groups have published several metrics on the observing strategy for several different science interests as white papers (e.g. Almoubayyed et al., 2020; Foley et al., 2018a; Gezari et al., 2018; Verma et al., 2019). However, an important metric that hasn't been evaluated is the impact of the LSST observing strategy on transient classification performance. The methods shown in this chapter provide a unique opportunity to create a metric of survey strategy against the performance of real-time transient classification.

Several LSST observing strategies have been simulated to meet some of the needs presented in the aforementioned white papers and are discussed in Jones et al. (2021a). The PLAsTiCC modelling software suite enables us to input these different observing strategy simulations to simulate LSST-like light curves. In future work, we should train RAPID on several different LSST survey simulations and evaluate how classification performance changes with different observing strategies. We can analyse several metrics such as how early we can classify common and rare transient classes based on different cadences, number of passband visits per night, and exposure time, for example.

The sheer scale and size of LSST means that we have a unique opportunity to revolutionise our understanding of the transient universe. Photometric transient classifiers play a pivotal role in helping to identify which transients are most useful for follow-up observations. Evaluating survey strategies with respect to classification performance is important to ensuring that the LSST observing strategy is optimised for effective photometric classifications. Future research should use works such as RAPID as a new metric for survey strategies.

4.7 Conclusions

The new generation of astronomical surveys is observing huge volumes of data that are being processed by transient brokers. These brokers require effective real-time classifications of transients to enable the astronomical community to prioritise their follow-up of incoming transient alerts based on the best prediction of the transient type. In Chapter 2, we detailed our novel real-time classifier, RAPID, that was one of the first attempts at developing a real-time classifier capable of dealing with the anticipated data volumes. However, RAPID and most other transient classifiers have only demonstrated their performance on simulations. Very few classifiers have shown much success on real surveys. In this chapter, we have improved RAPID and built three different RAPID classifiers for real transients observed by ZTF and Pan-STARRS. Our work is the first to show success at classifying *real* transients as a function of time.

We have made a few notable changes from the original RAPID methodology. The first being that we updated the neural network to use Temporal Convolutional Networks instead of Gated Recurrent Units because they were faster to train and provided more stability while training. The second major change was that we applied an adapted version of the data augmentation method used in Boone (2019) to help train our classifier on the smaller real datasets.

We used this new method to first build a classifier for Pan-STARRS using a collection of spectroscopically-confirmed PS1 supernovae. We compared our performance with recent classifiers built on the same dataset by V20 and H20, and have shown that we achieve better performance on all classes. Our PS1 classifier had excellent performance, achieving micro-averaged AUC scores of 0.98 on the validation set. We then applied our classifier to the cosmological SNIa sample from the Foundation Supernova Survey to assess whether we could recover transients useful for cosmology. We found that by setting an SNIa acceptance classification probability of 0.57, we could obtain 100% purity (precision) while still having a completeness (recall) of 91% (164 out of 180) of the SNIa sample. Currently, the Young Supernova Experiment based on the Pan-STARRS telescope are training our RAPID classifier to run on the survey. In the near future, we will share the results of our classifier trained on PS1 data to assist YSE in finding young supernovae.

Next, we built classifiers for the ZTF. We collected 2839 ZTF transients labelled by the astronomical community on the Open Supernova Catalog, and used our new RAPID methodology to build a classifier for ZTF. We achieved micro-averaged AUC scores of 0.86 and 0.95 at one day and at 65 days after trigger, respectively. These corresponded to accuracies over 80% for the SNIa, SLSN and combined core-collapse classes over the full light curve. The performance was not as good as the PS1 classifier's performance. We think that part of this can be explained by the better data quality from the spectroscopically-confirmed PS1 sample that was well-cleaned by V20 and the fact that PS1 has two extra passbands (*griz* instead of just *gr*). However, we think that the main reason for the poorer performance, particularly at early times, is because we do not have pre-trigger photometry. One of the critical components of the RAPID architecture is that it learns a "Pre-explosion" class that teaches the classifier about what the background looks like. Moreover, the rise in amplitude of the light curve from explosion time until it is bright enough to trigger an alert is very important information that RAPID uses in its classification. This early rise is exemplified in Figure 4.15, where the access to forced photometry provides important information about the early light curve. The importance of the early rise for RAPID is evident by the fact that the AUC scores increase quickly before trigger. The lack of pre-trigger photometry also limits the capability for early classification. Unlike PS1, the ZTF alert packets only contain upper limit magnitudes before trigger. LSST is expected to have real-time forced photometry on the light curves. We conclude that that lack of photometry before trigger severely limits our classification performance.

This is again exemplified when we built a new classifier trained on ZTF simulations and applied it to real data. Even after building a classifier that fixed many of the issues from Chapter 2, we still achieved poor performance on the collection of real ZTF light curves when using our classifier trained on ZTF simulations, achieving an overall full light curve accuracy of 64%. There remains a slight mismatch in the distributions of simulations and real data for the flux and redshift of SNe Ia that might also be contributing to the poor performance. We could not easily evaluate whether the simulations of the other classes matched real observations because of the very few transients in our dataset from non-SNIa classes.

Despite the poor performance of the classifier trained on ZTF simulations on real data, it had excellent performance on the simulated validation set, achieving a micro-averaged AUC score around 0.97 very shortly after trigger. The performance on real data highlighted that until simulations improve, classifiers should be trained on real data if they are to have success on real observations.

We ended the chapter by analysing the impact of observing strategy and, in particular, passband information on classification performance. We showed that using *gri* bands from Pan-STARRS provides significantly better classification performance than only using *gr* bands, particularly for SNe Ia. However, the improvement of using *griz* instead of only *gri* bands was only marginal. Future work should further analyse these trends and examine how classification performance changes with observing strategy. In particular, the PLAsTiCC-like simulations of different observing strategies provides a unique opportunity to analyse how transient light curves change with observing strategy, and in particular, how classification performance changes. LSST is at a critical point of deciding its observing strategy, and evaluating the impact of survey properties on classification performance is very important for optimising our ability to classify transients observed by LSST.

Chapter 5

Conclusions

5.1 Challenges in the new era of time-domain astronomy

Time-domain astronomy has reached an incredible new era where unprecedented amounts of data are becoming available. Upcoming surveys such as the Legacy Survey of Space and Time (LSST) are going to revolutionise transient astronomy. LSST is expected to observe over 10 million transient alerts every night, at least two orders of magnitude more than any preceding survey. Astronomers will be able to probe new regimes in the time-domain, as they survey the sky deeper, wider and faster than ever before. New surveys will provide opportunities to discover entirely new classes of transients while also enabling a deeper understanding of known classes. Correlating these sources with alerts from gravitational wave, high-energy particle, and neutrino observatories will enable new breakthroughs in multi-messenger astrophysics. However, with these new opportunities come new challenges. The massive alert rate from these surveys far outstrips the follow-up capacity of the entire astronomical community combined. Thus, it has never been more important that astronomers develop fast and automated methods of identifying transient candidates for follow-up observations.

This thesis has tackled two major challenges facing the new era of transient astronomy. These are as follows.

1. Early-time classification of the transient data stream.
2. Early-time detection of rare and previously unknown transients.

It is necessary that we meet these new challenges using methods that are scalable to the enormous volumes of data expected from next generation surveys. By doing so, astronomers will have much better tools to prioritise the follow-up of transients to better understand the physical mechanisms behind known objects and discover entirely new classes of transients.

The first challenge is being able to automate the classification of the incoming data stream where only sparse early-time data is available. It is important so that astronomers can identify the most interesting objects for more detailed analysis. Moreover, classifying objects as early as possible, will allow for early follow-up observations that provide a deeper insight into the progenitors and physical mechanism of transient classes.

However, classification eludes the problem of discovering new unknown classes of transients. This is the second challenge. Discovery in astronomy has been driven by serendipity and by identifying anomalies in data sets. For example, Jocelyn Bell first noticed an unusual "scruff" in her data that turned out to be the first ever discovery of a pulsar (Hewish et al., 1968). Similarly, Duncan Lorimer noticed an anomalous and incredibly bright pulse that turned out to be the first ever discovery of a Fast Radio Burst (Lorimer et al., 2007). What these discoveries have in common, is that they were serendipitous, and took the right person looking at the right time. However, if either Bell or Lorimer had been inundated with the data volumes expected from upcoming surveys, it is uncertain whether they would have stumbled upon these serendipitous discoveries. With the huge volumes of data in the next generation of surveys, astronomers are in need of a way to automate serendipity. That is, we need automatic anomaly detection tools that flag unusual and interesting transient phenomena that might warrant further follow-up observations. Furthermore, being able to identify anomalies as they happen in real-time is necessary to allow for early follow-up.

Spectroscopic surveys such as 4MOST and DESI are planning to use their multi-fibre spectroscopes to automatically follow many LSST targets. In addition, astronomers will use their own telescope time and resources to follow up the most interesting candidates. Transient brokers such as LASAIR, ANTARES, and ALerCE are aiming to ingest the observational data from LSST and trigger follow-up telescopes. These transient brokers require tools that perform classification and anomaly detection in real-time. In this thesis, I presented novel methods that meet these challenges.

5.2 Summary

This thesis has detailed the development of new photometric classification and anomaly detection frameworks, with the goal of meeting the needs of the new era of large-scale time-domain astronomy.

5.2.1 Classification

In Chapter 2, I presented a new real-time photometric classifier, RAPID. It is the first method specifically designed to provide early classification of astronomical time-series data and can automatically identify transients from within a day of the initial alert, to the full lifetime of a light curve. Using state-of-the-art recurrent neural networks with uni-directional gated recurrent units, we use only photometry to classify transients as a function of time. Using the modelling software developed for the Photometric LSST Astronomical Time-series Classification Challenge (PLAsTiCC, Kessler et al., 2019), we developed light curve simulations of 12 different transient classes that matched the observing properties of the Zwicky Transient Facility (ZTF). We applied our neural-network-based classifier to this simulated dataset and achieved excellent results competitive with other contemporary approaches. However, unlike most previous work, RAPID does not require the full phase coverage of the light curve, but is flexible enough to provide photometric classifications given any phase coverage of the light curves. A critical component of RAPID that enables its early classification is that it uses measurements of the source before an alert is triggered. The photometry with insufficient significance to trigger an alert still encodes information about the transient that RAPID is able to use to obtain good classification probabilities as soon as the alert is triggered. Moreover, our architecture defines a pre-explosion phase of the transient that enables RAPID to learn what the background and the pre-transient phase looks like so that it learns to distinguish transients from the background.

We used a range of metrics to evaluate our classifier's performance and presented detailed confusion matrices, Receiver Operating Characteristic (ROC) curves, and measures of precision and recall for each transient class. Our success on the simulated dataset is best summarised by the micro-averaged area under the ROC curve (AUC). We obtained AUC scores averaged across the 12 transient classes of 0.95 and 0.98 2 days and 40 days after an alert trigger, respectively. Thus, even at early times, our classification capabilities are impressive and exceed the success of other similar attempts at classification.

Many previous attempts at classification use feature-based learning algorithms that often are too computationally expensive to be feasible to run on the real-time data streams coming from next generation surveys such as LSST. To compare our approach against feature-based learning methods, we also developed a Random Forest classifier trained on features extracted from the light curve. We found that the classification performance was comparable, but RAPID had the advantage of obtaining time-varying classifications and being incredibly fast, being able to easily classify the millions of alerts expected from new surveys.

However, since publishing our work in Muthukrishna et al. (2019a), we have tried to apply our classifier to real data from the ZTF MSIP public survey. We have found that we achieve unreliable results on real data when using our classifier that was trained on simulations. In Chapter 4, we detail many reasons for the mediocre performance on real data. Firstly, unlike our PLAsTiCC-based ZTF simulations, the real alert packets from the public ZTF survey do not provide forced photometry on pre-trigger data. Instead, only upper limit magnitudes are provided, and we cannot easily use this information to train RAPID. We noticed that pre-alert photometry is essential to providing good early-time classifications. Other issues that limited our classification performance were that the CCSNe models in our ZTF simulations were flawed and were not a good representation of real data. Similarly, there were small bugs in the SNIa-91bg and SNIa-x PLAsTiCC models.

In Chapter 4, we created new ZTF simulations that used better core-collapse supernova models from Vincenzi et al. (2019), we used updated SNIa-91bg and SNIa-x PLAsTiCC models, made use of the latest kilonova models from Bulla (2019), and we updated the observing logs used in the simulations to use 3-years of data to better represent the observing conditions in ZTF. The results of these improvements were presented, and we showed that we significantly improved upon the classification performance shown in Chapter 2, especially when concerned with the SNII and SNIbc classes. However, when applying our classifier to a collection of real ZTF data, we still had a poor performance when compared to our performance on the simulations. We discussed reasons for this performance including the issue of no pre-trigger photometry in the real data, and mismatches between our simulated and real light curve distributions.

To build a classifier that had good performance on real observations, we instead trained classifiers on real data instead of simulations. We collected a range of ZTF transients that had been spectroscopically classified by the astronomical community, and also obtained a dataset of Pan-STARRS1 (PS1) light curves from Villar et al. (2020c). These datasets were obviously much smaller than our simulations, and so we used data augmentation techniques to build a bigger training set. We used the data augmentation methods from Boone (2019), which were used to win the PLAsTiCC challenge on full light curve classification. Using this method, and using Temporal Convolutional Neural Networks instead of Recurrent Neural Networks (because of their slightly improved training performance), we built classifiers trained on ZTF data and PS1 data.

Our classifier trained on spectroscopically-confirmed PS1 light curves had excellent performance, achieving micro-averaged AUC scores around 0.98. These results were better than full light curve classifiers from Hosseinzadeh et al. (2020) and Villar et al. (2020c) who built classifiers on the same dataset. We then used this newly built classifier to classify the Foundation cosmological sample of SNe Ia. We presented the precision and recall of our classifier at a range of threshold probabilities, and demonstrated that we could obtain 100% precision (purity) in our SNIa classification while still having a completeness (recall) of 91% of the SNe Ia if we set an SNIa acceptance classification probability from RAPID of 0.57. This result is impressive because the cosmological sample was spectroscopically confirmed, but we can achieve this precision and recall using only photometry.

Our classifier trained on our collection of ZTF real data had reasonable performance, achieving AUCs of 0.85 and 0.94 within a day and 65 days after trigger, respectively. The poorer performance on ZTF light curves compared to PS1 light curves is caused by the extra 2 passbands of colour information in PS1, and the fact that we have pre-trigger photometry in the PS1 light curves. We analysed the impact of passband information on classification performance for Pan-STARRS, illustrating that only either i or z band is needed in addition to gr bands to provide significantly better classifications. We conclude by highlighting that we can create classifiers trained on real data, but that having pre-trigger photometry and extra NIR colour information is important to having good early-time classifiers. Pre-trigger photometry and six passbands are expected from LSST.

5.2.2 Anomaly detection

The classifiers we built in Chapters 2 and 4 will be important for categorising the millions of transient alerts in upcoming surveys. However, if a new class of transient is presented to these classifiers, they will likely provide unexpected results. Standard supervised classifiers are not able to classify objects they have not been specifically trained on. To flag new previously unseen transient types, anomaly detection algorithms are required. In Chapter 3, I present two novel anomaly detection frameworks that can be applied in real-time and are fast enough to run on the upcoming LSST.

The key idea of our approaches is to build transient models that can predict future fluxes, and then define anomaly scores based on the deviation of a prediction and an observation. The first approach is a probabilistic deep neural network (DNN) built using Temporal Convolutional Networks, where we train separate models for each transient class. It regresses future fluxes against past data to predict a future set of fluxes. The second approach is based on a parametric fit to a partial light curve using the Bazin function (Bazin et al., 2009), where we extrapolate a prediction 3-days after a given set of past data. We build a Bayesian model, defining a prior distribution of Bazin parameters based on fits to the population of transients from a transient class. The priors for a specific class enable us to make predictions of a new light curve assuming a particular class. In both the DNN and Bazin approaches, we compare our prediction (assuming a particular transient class) with an observation to obtain anomaly scores relative to a specific transient class.

Our two methods allow us to identify anomalies as a function of time, and we have demonstrated their performances on both ZTF simulations and real data from the ZTF MSIP public survey. We show that the DNN model is too flexible, in that in can predict fluxes very effectively for any class despite being trained on one class. This means, that it is not able to distinguish between different classes, and its flexibility makes it a poor anomaly detector. However, the Bazin model is very effective at identifying anomalies. In particular we combine the models from three common supernova classes, SNIa, SNII, and SNIbc, to identify anomalies relative to these classes. We plot the False Anomaly Rates (FAR) against the True Anomaly Rates (TAR) for a range of different threshold anomaly scores as a ROC curve, and show that we are able to obtain low FARs with high TARs. We show that the AUC rises over the course of the light curve, and after about 20 days after trigger, we are able to effectively find most anomalous classes with an average AUC around 0.8.

We also built an anomaly detector using the Bazin approach on real data from the ZTF MSIP public survey. We obtained anomaly score relative to the SNIa class and obtained AUC scores over the full light curve of 0.84, 0.65, and 0.87 for the SNII, SNIbc, and SLSN classes, respectively relative to the SNIa class. Thus, our Bayesian parametric approach is effective at identifying transients outside the modelled class, making it an effective anomaly detector.

5.3 Future directions

Since RAPID's publication in Muthukrishna et al. (2019a), it has been used widely throughout the transient astronomical community. At the time of writing, the paper has 48 citations. I have released the work as a publicly available software with documentation available at https://astrorapid.readthedocs.io/. The trained classifiers shown throughout this thesis and our novel classification architecture with easy-to-use code that enables astronomers to train RAPID on their own datasets is available on GitHub[1] and the Python Packaging Index[2]. Recently, groups such as the AMPEL (Alert Management, Photometry and Evaluation of Light curves) collaboration (Nordin et al., 2019) have trained RAPID on their private ZTF alerts to identify SNe Ia and CCSNe. Similarly, the recent Young Supernova Experiment (YSE, Jones et al., 2021b) are currently training RAPID on Pan-STARRS simulations to allow them to identify young transients. I am working closely with these teams to help train a RAPID classifier that is effective for real-time classification for YSE and AMPEL.

Furthermore, the LASAIR and ANTARES transient brokers are working to implement RAPID in their pipelines to classify transients from the ZTF MSIP data stream. Working with these groups has highlighted issues with applying RAPID to real data. I have tried to solve many of these issues in Chapter 4, but with current PLAsTiCC simulations, we have concluded that we need to train RAPID on real observations before it can be successful on real data. In future work, I will share the improved real data classifiers developed in this thesis with LASAIR and ANTARES.

We have shown that classifiers trained on real data are more reliable than classifiers trained on simulations. However, the problem with real datasets is that they are dominated by SNe Ia, and there are too few non-SNIa to represent the diversity expected from new surveys. The PLAsTiCC simulations were designed to match the expected diversity and distribution of transients in the LSST. However, upon creating ZTF simulations from the PLAsTiCC modelling software, it is clear that the simulated transients (at least in the SNIa class) look different from the types of light curves being observed by ZTF. This was evidenced by a slight mismatch in the SNIa distributions of fluxes and redshifts, and by the poor performance of our classifier trained on simulations applied to real data. PLAsTiCC version 2 is currently in preparation to fix many of the problems highlighted in the first version of PLAsTiCC, and to focus on an early-time photometric challenge. In future work, we will use the analysis presented in this thesis to help improve the simulations in PLAsTiCC v2.

[1] https://github.com/daniel-muthukrishna/astrorapid

[2] https://pypi.org/project/astrorapid/

Recent work by Gagliano et al. (2021) has shown that using only host galaxy information without any photometry of spectroscopy of the transient event can achieve 70% accuracy at distinguishing supernova classes. The current version of RAPID only uses photometry, sky position, and redshift information. Throughout Chapter 4, we saw that using pre-trigger photometry and redshift provided a lot of information that gave good classifications as early as the time of trigger. Using the set of host galaxy properties and features selected by Gagliano et al. (2021) as additional contextual information in RAPID should enable even better classification particularly at early-times.

At the time of writing, the LSST is at a critical point of deciding on the optimal observing strategy. Many different strategies have been proposed because different science objectives benefit from different cadences and survey properties. The cadence, colour, and other survey properties has a significant impact on our ability to classify different transient classes. We have begun simulating PLAsTiCC light curves using a few different proposed observing strategies for the LSST. In future work, we will apply RAPID to the LSST simulations to evaluate how different observing strategies affect the classification performance of different classes of transients at early and late phases.

Furthermore, we have shown the success of our anomaly detection framework at identifying rare transient classes with respect to common supernovae. In future, we should evaluate the performance of our method on other anomalous transients, such as the recently discovered Fast Blue Optical Transients (FBOTs, see Section 1.2.9) and other one-off events. However, what is an important limitation of our work is that we have not made the distinction between anomalies and *interesting* anomalies. While we think that a lot of the transient events that are uninteresting will be removed by *real-bogus* cuts by transient brokers, there may still be many uninteresting transient phenomena that get flagged by our algorithm. Recent work by Lochner & Bassett (2020) and Ishida et al. (2019a) have developed Active Learning frameworks that include *Human-in-the-loop learning*. These frameworks involve getting astronomers to select which of the flagged anomalies were interesting and then getting the algorithm to find more anomalies that are similar to that event. For static data sets (not in real-time) an algorithm called isolation forests have been used significantly and have been shown to have good success at incorporating Human-in-the-loop learning. However, for time-series anomaly detection, this is a more difficult challenge. But, future work should use the methods learned in works such as Lochner & Bassett (2020) and Ishida et al. (2019a) to ensure that only the subset of interesting anomalies are flagged by automated anomaly detectors.

5.4 Final remarks

Overall, this thesis has provided unique methods of identifying follow-up candidates in real-time transient surveys. Using anomaly detection alongside transient classifiers will enable astronomers to prioritise which transients are most suitable for followup observations based on the best classification of the transient type and a transient's likelihood of being an anomalous new transient.

I envisage running RAPID and our anomaly detector simultaneously on upcoming surveys such as LSST. Using both the class probability scores from RAPID and the anomaly scores will provide astronomers with two useful metrics to prioritise follow up candidates. For example, an astronomer hoping to follow-up common supernovae should follow up candidates that have both a high classification probability score for a common supernova type and a low anomaly score.

Real-time anomaly detection and classification are going to be hugely important for discovery in the new era of large-scale transient surveys. This thesis has provided important research into solving these problems for the future of time-domain astronomy.

References

Abadi M., et al., 2016, in Proceedings of the 12th USENIX Conference on Operating Systems Design and Implementation. OSDI'16. USENIX Association, Berkeley, CA, USA, pp 265–283

Abbott B. P., et al., 2017a, Physical Review Letters, 119, 161101

Abbott B. P., et al., 2017b, Nature, 551, 85

Abbott B. P., et al., 2017c, ApJ, 848, L12

Aguirre C., Pichara K., Becker I., 2018, preprint, (arXiv:1810.09440)

Aldering G., et al., 2002, in Tyson J. A., Wolff S., eds, Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series Vol. 4836, Survey and Other Telescope Technologies and Discoveries. pp 61–72, doi:10.1117/12.458107

Almoubayyed H., et al., 2020, MNRAS, 499, 1140

Ambikasaran S., Foreman-Mackey D., Greengard L., Hogg D. W., O'Neil M., 2015, IEEE Transactions on Pattern Analysis and Machine Intelligence, 38, 252

Arcavi I., et al., 2016, ApJ, 819, 35

Arcavi I., et al., 2017, Nature, 551, 64

Arnett W. D., 1982, ApJ, 253, 785

Astier P., et al., 2006, A&A, 447, 31

Bahdanau D., Cho K., Bengio Y., 2014, preprint, (arXiv:1409.0473)

Bai S., Zico Kolter J., Koltun V., 2018, arXiv e-prints, p. arXiv:1803.01271

Bazin G., et al., 2009, A&A, 499, 653

Becker A., 2015, HOTPANTS: High Order Transform of PSF ANd Template Subtraction (ascl:1504.004)

Bellm E., 2014, in Wozniak P. R., Graham M. J., Mahabal A. A., Seaman R., eds, The Third Hot-wiring the Transient Universe Workshop. pp 27–33 (arXiv:1410.8185)

Bellm E. C., et al., 2019, PASP, 131, 018002

Berger E., et al., 2009, ApJ, 699, 1850

Betoule M., et al., 2014a, A&A, 568, A22

Betoule M., et al., 2014b, Astronomy & Astrophysics, submitted, 30

Bildsten L., Shen K. J., Weinberg N. N., Nelemans G., 2007, ApJ, 662, L95

Blondin S., Tonry J. L., 2007, ApJ, 666, 1024

Bloom J. S., et al., 2012, PASP, 124, 1175

Blundell C., Cornebise J., Kavukcuoglu K., Wierstra D., 2015, in Bach F., Blei D., eds, Proceedings of Machine Learning Research Vol. 37, Proceedings of the 32nd International Conference on Machine Learning. PMLR, Lille, France, pp 1613–1622, http://proceedings.mlr.press/v37/blundell15.html

Bond I. A., et al., 2001a, MNRAS, 327, 868

Bond I. A., et al., 2001b, MNRAS, 327, 868

Boone K., 2019, arXiv e-prints, p. arXiv:1907.04690

Branch D., Wheeler J. C., 2017a, Type IIL Supernovae. Springer Berlin Heidelberg, Berlin, Heidelberg, pp 267–280, doi:10.1007/978-3-662-55054-0_13, https://doi.org/10.1007/978-3-662-55054-0_13

Branch D., Wheeler J. C., 2017b, Type IIn Supernovae. Springer Berlin Heidelberg, Berlin, Heidelberg, pp 281–318, doi:10.1007/978-3-662-55054-0_14, https://doi.org/10.1007/978-3-662-55054-0_14

Branch D., Wheeler J. C., 2017c, Type IIb Supernovae. Springer Berlin Heidelberg, Berlin, Heidelberg, pp 319–344, doi:10.1007/978-3-662-55054-0_15, https://doi.org/10.1007/978-3-662-55054-0_15

Branch D., Wheeler J. C., 2017d, Type Ib Supernovae. Springer Berlin Heidelberg, Berlin, Heidelberg, pp 345–377, doi:10.1007/978-3-662-55054-0_16, https://doi.org/10.1007/978-3-662-55054-0_16

Branch D., Wheeler J. C., 2017e, Type Ic Supernovae. Springer Berlin Heidelberg, Berlin, Heidelberg, pp 379–412, doi:10.1007/978-3-662-55054-0_17, https://doi.org/10.1007/978-3-662-55054-0_17

Breiman L., 2001, Mach. Learn., 45, 5

Bulla M., 2019, MNRAS, 489, 5037

Campbell H., et al., 2013, ApJ, 763, 88

Cardelli J. A., Clayton G. C., Mathis J. S., 1989, ApJ, 345, 245

Carrasco-Davis R., et al., 2018, arXiv e-prints, p. arXiv:1807.03869

Chambers K. C., et al., 2016, arXiv e-prints, p. arXiv:1612.05560

Charnock T., Moss A., 2017, ApJ, 837, L28

Che Z., Purushotham S., Cho K., Sontag D., Liu Y., 2018, Scientific Reports, 8, 6085

Chen H.-Y., Fishbach M., Holz D. E., 2018, Nature, 562, 545

Cho K., van Merrienboer B., Gulcehre C., Bahdanau D., Bougares F., Schwenk H., Bengio Y., 2014, in Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). Association for Computational Linguistics, pp 1724–1734, doi:10.3115/v1/D14-1179

Chollet F., et al., 2015, Keras, https://keras.io

Chung J., Gulcehre C., Cho K., Bengio Y., 2014, in NIPS 2014 Workshop on Deep Learning, December 2014.

Cooke J., et al., 2012, Nature, 491, 228

Coppejans D. L., et al., 2020, ApJ, 895, L23

Coulter D. A., et al., 2017, Science, 358, 1556

Cowperthwaite P. S., et al., 2017, ApJ, 848, L17

D'Andrea C. B., et al., 2018, arXiv e-prints, p. arXiv:1811.09565

Dark Energy Survey Collaboration et al., 2016, MNRAS, 460, 1270

Davis T. M., et al., 2007, ApJ, 666, 716

Djorgovski S. G., et al., 2011, preprint, (arXiv:1102.5004)

Drout M. R., et al., 2011, ApJ, 741, 97

Drout M. R., et al., 2014, ApJ, 794, 23

Drout M. R., et al., 2017, Science, 358, 1570

Dunham E. W., Mandushev G. I., Taylor B. W., Oetiker B., 2004, PASP, 116, 1072

Efstathiou G., Gratton S., 2019, arXiv e-prints, p. arXiv:1910.00483

Evans P. A., et al., 2017, Science, 358, 1565

Eyer L., Mowlavi N., 2007, Journal of Physics: Conference Series, Volume 118, Issue 1, id. 012010 (2008)., 118

Ezquiaga J. M., Zumalacárregui M., 2018, Frontiers in Astronomy and Space Sciences, 5, 44

Fang K., Metzger B. D., Murase K., Bartos I., Kotera K., 2019, ApJ, 878, 34

Fernández S., Graves A., Schmidhuber J., 2007, in Proceedings of the 17th International Conference on Artificial Neural Networks. ICANN'07. Springer-Verlag, Berlin, Heidelberg, pp 220–229

Filippenko A. V., 1997, Annual Review of Astronomy and Astrophysics, 35, 309

Filippenko A. V., et al., 1992a, AJ, 104, 1543

Filippenko A. V., et al., 1992b, ApJ, 384, L15

Fitzpatrick E. L., 1999, PASP, 111, 63

Flewelling H. A., et al., 2020, ApJ, 251, 7

Folatelli G., et al., 2010, AJ, 139, 120

Foley R. J., Mandel K., 2013, ApJ, 778, 167

Foley R. J., et al., 2013, ApJ, 767, 57

Foley R. J., et al., 2018a, arXiv e-prints, p. arXiv:1812.00514

Foley R. J., et al., 2018b, MNRAS, 475, 193

Foreman-Mackey D., Hogg D. W., Lang D., Goodman J., 2013, PASP, 125, 306

Förster F., et al., 2020, arXiv e-prints, p. arXiv:2008.03303

Fransson C., Lundqvist P., 1989, ApJ, 341, L59

Fremling C., Dugas A., Sharma Y., 2018, Transient Name Server Classification Report, 1810

Fremling C., Dugas A., Sharma Y., 2019, Transient Name Server Classification Report, 329

Gagliano A., Narayan G., Engel A., Carrasco Kind M., LSST Dark Energy Science Collaboration 2021, ApJ, 908, 170

Gaia Collaboration et al., 2019, A&A, 623, A110

Gal Y., Ghahramani Z., 2015a, arXiv e-prints, p. arXiv:1506.02142

Gal Y., Ghahramani Z., 2015b, arXiv e-prints, p. arXiv:1506.02157

Gal-Yam A., 2012a, in Roming P., Kawai N., Pian E., eds, Vol. 279, Death of Massive Stars: Supernovae and Gamma-Ray Bursts. pp 253–260 (arXiv:1206.2157), doi:10.1017/S1743921312013014

Gal-Yam A., 2012b, Science, Volume 337, Issue 6097, pp. 927- (2012)., 337, 927

Gal-Yam A., 2019, ARA&A, 57, 305

Gal-Yam A., et al., 2009, Nature, 462, 624

Garnavich P. M., Tucker B. E., Rest A., Shaya E. J., Olling R. P., Kasen D., Villar A., 2016, ApJ, 820, 23

Gezari S., et al., 2015, ApJ, 804, 28

Gezari S., van Velzen S., Hung T., Cenko B., Arcavi I., 2018, arXiv e-prints, p. arXiv:1812.07036

Giles D., Walkowicz L., 2019, MNRAS, 484, 834

Goobar A., Leibundgut B., 2011, Annual Review of Nuclear and Particle Science, 61, 251

Goodfellow I. J., Shlens J., Szegedy C., 2015, in Bengio Y., LeCun Y., eds, 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings. http://arxiv.org/abs/1412.6572

Graham M. J., et al., 2019, PASP, 131, 078001

Guillochon J., Parrent J., Kelley L. Z., Margutti R., 2017, ApJ, 835, 64

Guillochon J., Nicholl M., Villar V. A., Mockler B., Narayan G., Mandel K. S., Berger E., Williams P. K. G., 2018a,
apjs, 236, 6

Guillochon J., Nicholl M., Villar V. A., Mockler B., Narayan G., Mandel K. S., Berger E., Williams P. K. G., 2018b, ApJ, 236, 6

Guo C., Pleiss G., Sun Y., Weinberger K. Q., 2017, in Proceedings of the 34th International Conference on Machine Learning - Volume 70. ICML'17. JMLR.org, p. 1321–1330

Guy J., et al., 2007, A&A, 466, 11

Guy J., et al., 2010,
aap, 523, A7

Hannun A., et al., 2014, preprint, (arXiv:1412.5567)

Hewish A., Bell S. J., Pilkington J. D. H., Scott P. F., Collins R. A., 1968, Nature, 217, 709

Hicken M., et al., 2009, ApJ, 700, 331

Hillebrandt W., Niemeyer J. C., 2000, Annual Review of Astronomy and Astrophysics, 38, 191

Hillier D. J., Dessart L., 2019, A&A, 631, A8

Hilton E. J., Hawley S. L., Kowalski A. F., Holtzman J., 2010, ProQuest Dissertations And Theses; Thesis (Ph.D.)–University of Washington, 2011.; Publication Number: AAT 3485401; ISBN: 9781267030030; Source: Dissertation Abstracts International, Volume: 73-02, Section: B, page: 0990.; 157 p.

Hinners T. A., Tat K., Thorp R., 2018, AJ, 156, 7

Hložek R., et al., 2020, arXiv e-prints, p. arXiv:2012.12392

Ho A. Y. Q., et al., 2019, ApJ, 871, 73

Hochreiter S., Schmidhuber J., 1997, Neural Comput., 9, 1735

Hogg D. W., Baldry I. K., Blanton M. R., Eisenstein D. J., 2002, arXiv Astrophysics e-prints,

Hosseinzadeh G., et al., 2017, The Astrophysical Journal, 845, L11

Hosseinzadeh G., et al., 2020, ApJ, 905, 93

Hubble E., 1929, Proceedings of the National Academy of Sciences, 15, 168

Iben I. J., Tutukov A. V., 1984, The Astrophysical Journal Supplement Series, 54, 335

Ioffe S., Szegedy C., 2015, in Proceedings of the 32nd International Conference on Machine Learning - Volume 37. ICML'15. JMLR.org, pp 448–456

Ishida E. E. O., de Souza R. S., 2013, MNRAS, 430, 509

Ishida E. E. O., et al., 2019a, arXiv e-prints, p. arXiv:1909.13260

Ishida E. E. O., et al., 2019b, MNRAS, 483, 2

Ivezić Ž., et al., 2019, ApJ, 873, 111

Jamal S., Bloom J. S., 2020, ApJ, 250, 30

Jha S. W., 2017, Type Iax Supernovae. p. 375, doi:10.1007/978-3-319-21846-5_42

Jones D. O., et al., 2017, ApJ, 843, 6

Jones D. O., et al., 2019, ApJ, 881, 19

Jones L., Yoachim P., Ivezic Z., Eric H. Neilsen J., Ribeiro T., 2021a, lsst-pst/pstn-051: v1.2.1, doi:10.5281/zenodo.4598873, https://doi.org/10.5281/zenodo.4598873

Jones D. O., et al., 2021b, ApJ, 908, 143

Jozefowicz R., Zaremba W., Sutskever I., 2015, in Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37. ICML'15. JMLR.org, p. 2342–2350

Kalchbrenner N., Espeholt L., Simonyan K., van den Oord A., Graves A., Kavukcuoglu K., 2016, arXiv e-prints, p. arXiv:1610.10099

Karpenka N. V., Feroz F., Hobson M. P., 2013, MNRAS, 429, 1278

Kasen D., 2010, The Astrophysical Journal, 708, 1025

Kasen D., Bildsten L., 2010,
apj, 717, 245

Kasen D., Woosley S. E., Heger A., 2011,
apj, 734, 102

Kasen D., Metzger B., Barnes J., Quataert E., Ramirez-Ruiz E., 2017, Nature, 551, 80

Kashi A., Soker N., 2017, MNRAS, 468, 4938

Kasliwal M. M., 2011, PhD thesis, California Institute of Technology

Kasliwal M. M., et al., 2012, ApJ, 755, 161

Kaspi V. M., Beloborodov A., 2017, Annual Review of Astronomy and Astrophysics, vol. 55, issue 1, pp. 261-301, 55, 261

Kelly P. L., Kirshner R. P., Pahre M., 2008, ApJ, 687, 1201

Kessler R., et al., 2009, PASP, 121, 1028

Kessler R., Conley A., Jha S., Kuhlmann S., 2010a, preprint, (arXiv:1001.5210)

Kessler R., et al., 2010b, PASP, 122, 1415

Kessler R., et al., 2013,
apj, 764, 48

Kessler R., et al., 2019, arXiv e-prints, p. arXiv:1903.11756

Kinemuchi K., Smith H. A., Woźniak P. R., McKay T. A., ROTSE Collaboration 2006, AJ, 132, 1202

Kingma D. P., Ba J., 2015, in Proceedings of the 3rd International Conference on Learning Representations. ICLR 2015

Kingma D. P., Welling M., 2013, arXiv e-prints, p. arXiv:1312.6114

Kotsiantis S., Kanellopoulos D., Pintelas P., et al., 2006, GESTS International Transactions on Computer Science and Engineering, 30, 25

Kozyreva A., Kromer M., Noebauer U. M., Hirschi R., 2018, MNRAS, 479, 3106

Krizhevsky A., Sutskever I., Hinton G. E., 2012, in Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1. NIPS'12. Curran Associates Inc., USA, pp 1097–1105

Kuin N. P. M., et al., 2019, MNRAS, 487, 2505

LSST Science Collaboration et al., 2009, arXiv e-prints, p. arXiv:0912.0201

LSST Science Collaborations et al., 2017, preprint, (arXiv:1708.04058)

Lea C., Vidal R., Reiter A., Hager G. D., 2016, arXiv e-prints, p. arXiv:1608.08242

Leavitt H. S., 1908, Annals of Harvard College Observatory, 60, 87

Leavitt H. S., Pickering E. C., 1912, Harvard College Observatory Circular, 173, 1

Leloudas G., et al., 2016, Nature Astronomy, 1, 0002

Lemaître G., 1927, Annales de la Soci&eacute;t&eacute; Scientifique de Bruxelles, 47, 49

Li X., Wu X., 2015, in Proceedings of the International Conference on Acousitics, Speech, and Signal Processing. ICASSP'15. IEEE

Li W., Chornock R., Leaman J., Filippenko A. V., Poznanski D., Wang X., Ganeshalingam M., Mannucci F., 2011, Monthly Notices of the Royal Astronomical Society, 412, 1473

Livio M., Mazzali P., 2018, Phys. Rep., 736, 1

Lochner M., Bassett B. A., 2020, arXiv e-prints, p. arXiv:2010.11202

Lochner M., McEwen J. D., Peiris H. V., Lahav O., Winter M. K., 2016, ApJ, 225, 31

Lochner M., et al., 2018, arXiv e-prints, p. arXiv:1812.00515

Lochner M., et al., 2021, arXiv e-prints, p. arXiv:2104.05676

Lomb N. R., 1976, Ap&SS, 39, 447

Lorimer D. R., Bailes M., McLaughlin M. A., Narkevic D. J., Crawford F., 2007, Science, 318, 777

Lunnan R., et al., 2017, ApJ, 836, 60

Lunnan R., et al., 2020, ApJ, 901, 61

Lyutikov M., Toonen S., 2019, MNRAS, 487, 5618

Malanchev K. L., et al., 2020, arXiv e-prints, p. arXiv:2012.01419

Malz A., et al., 2018, preprint, (arXiv:1809.11145)

Margutti R., et al., 2019, ApJ, 872, 18

Marion G. H., et al., 2016, The Astrophysical Journal, 820, 92

Martínez-Galarza J. R., Bianco F., Crake D., Tirumala K., Mahabal A. A., Graham M. J., Giles D., 2020, arXiv e-prints, p. arXiv:2009.06760

Martínez-Palomera J., Bloom J. S., Abrahams E. S., 2020, arXiv e-prints, p. arXiv:2005.07773

Maxted P. F. L., Hutcheon R. J., 2018, eprint arXiv:1803.10522

McCulloch W. S., Pitts W., 1943, The bulletin of mathematical biophysics, 5, 115

Metzger B. D., 2017, Living Reviews in Relativity, 20, 3

Mislis D., Bachelet E., Alsubai K. A., Bramich D. M., Parley N., 2016, MNRAS, 455, 626

Mockler B., Guillochon J., Ramirez-Ruiz E., 2019, apj, 872, 151

Modjaz M., et al., 2014, AJ, 147, 99

Modjaz M., Gutiérrez C. P., Arcavi I., 2019, Nature Astronomy, 3, 717

Möller A., de Boissière T., 2020, MNRAS, 491, 4277

Möller A., et al., 2016, J. Cosmology Astropart. Phys., 12, 008

Morgan J. N., Sonquist J. A., 1963, Journal of the American Statistical Association, 58, 415

Moss A., 2018, preprint, (arXiv:1810.06441)

Muthukrishna D., Narayan G., Mandel K. S., Biswas R., Hložek R., 2019a, PASP, 131, 118002

Muthukrishna D., Parkinson D., Tucker B. E., 2019b, ApJ, 885, 85

Nair V., Hinton G. E., 2010, in Proceedings of the 27th International Conference on Machine Learning. ICML'10. Omnipress, USA, pp 807–814

Narayan G., et al., 2018, ApJ, 236, 9

Naul B., Bloom J. S., Pérez F., van der Walt S., 2018, Nature Astronomy, 2, 151

Neil D., Pfeiffer M., Liu S.-C., 2016, in Proceedings of the 30th International Conference on Neural Information Processing Systems. NIPS'16. Curran Associates Inc., USA, pp 3889–3897

Neilson H. R., Percy J. R., Smith H. A., 2016, The Journal of the American Association of Variable Star Observers, vol. 44, no. 2, p. 179, 44, 179

Newling J., et al., 2011, MNRAS, 414, 1987

Nicholl M., et al., 2013, Nature, 502, 346

Nicholl M., Berger E., Margutti R., Blanchard P. K., Milisavljevic D., Challis P., Metzger B. D., Chornock R., 2017a, ApJ, 835, L8

Nicholl M., Guillochon J., Berger E., 2017b, apj, 850, 55

Noebauer U. M., Kromer M., Taubenberger S., Baklanov P., Blinnikov S., Sorokina E., Hillebrandt W., 2017, Monthly Notices of the Royal Astronomical Society, Volume 472, Issue 3, p.2787-2799, 472, 2787

Nordin J., et al., 2019, A&A, 631, A147

Nun I., Pichara K., Protopapas P., Kim D.-W., 2014, ApJ, 793, 23

Ofek E. O., et al., 2007, ApJ, 659, L13

PLASTICC Team PLASTICC Modelers 2019, Unblinded Data for PLAsTiCC Classification Challenge, doi:10.5281/zenodo.2539456, https://doi.org/10.5281/zenodo.2539456

PLASTICC Team PLASTICC Modelers 2021, Libraries & Recommended Citations for using PLAsTiCC Models, doi:10.5281/zenodo.4419884, https://doi.org/10.5281/zenodo.4419884

Padovani P., 2017, Frontiers in Astronomy and Space Sciences, 4, 35

Pascanu R., Mikolov T., Bengio Y., 2013, in Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28. ICML'13. JMLR.org, p. III–1310–III–1318

Pasquet J., Pasquet J., Chaumont M., Fouchez D., 2019, arXiv e-prints, p. arXiv:1901.01298

Pastorello A., Fraser M., 2019, Nature Astronomy, 3, 676

Perets H. B., et al., 2010, Nature, 465, 322

Perley D. A., et al., 2019, MNRAS, 484, 1031

Perlmutter S., et al., 1999, ApJ, 517, 565

Phillips M. M., 1993, ApJ, 413, L105

Pian E., et al., 2017, Nature, 551, 67

Pierel J. D. R., et al., 2018, pasp, 130, 114504

Planck Collaboration et al., 2014, A&A, 571, A16

Planck Collaboration et al., 2020, A&A, 641, A6

Poznanski D., Maoz D., Gal-Yam A., 2007, AJ, 134, 1285

Prentice S. J., et al., 2018, ApJ, 865, L3

Pruzhinskaya M. V., Malanchev K. L., Kornilov M. V., Ishida E. E. O., Mondon F., Volnova A. A., Korolev V. S., 2019, MNRAS, 489, 3591

Quimby R. M., 2012, in Roming P., Kawai N., Pian E., eds, Vol. 279, Death of Massive Stars: Supernovae and Gamma-Ray Bursts. pp 22–28, doi:10.1017/S174392131201263X

Quimby R. M., Aldering G., Wheeler J. C., Höflich P., Akerlof C. W., Rykoff E. S., 2007, ApJ, 668, L99

Quimby R. M., et al., 2018, ApJ, 855, 2

Rasmussen C., Williams C., 2006, Gaussian Processes for Machine Learning. Adaptive Computation and Machine Learning, MIT Press, Cambridge, MA, USA

Rau A., et al., 2009, PASP, 121, 1334

Razavian A. S., Azizpour H., Sullivan J., Carlsson S., 2014, in Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops. CVPRW '14. IEEE Computer Society, Washington, DC, USA, pp 512–519, doi:10.1109/CVPRW.2014.131

Rebbapragada U., Protopapas P., Brodley C. E., Alcock C., 2009, in Bohlender D. A., Durand D., Dowler P., eds, Astronomical Society of the Pacific Conference Series Vol. 411, Astronomical Data Analysis Software and Systems XVIII. p. 264

Rees M. J., 1988, Nature, 333, 523

Ren J., Christlieb N., Zhao G., 2012, Research in Astronomy and Astrophysics, 12, 1637

Rest A., et al., 2005, ApJ, 634, 1103

Rest A., et al., 2014, ApJ, 795, 44

Rest A., et al., 2018, Nature Astronomy, 2, 307

Revsbech E. A., Trotta R., van Dyk D. A., 2018, MNRAS, 473, 3969

Richards J. W., Homrighausen D., Freeman P. E., Schafer C. M., Poznanski D., 2012, MNRAS, 419, 1121

Ricker G. R., et al., 2015, Journal of Astronomical Telescopes, Instruments, and Systems, 1, 014003

Riess A. G., et al., 1998, AJ, 116, 1009

Riess A. G., Casertano S., Yuan W., Bowers J. B., Macri L., Zinn J. C., Scolnic D., 2021, ApJ, 908, L6

Rothchild D. i. p., 2018, Opsim Run Select, http://altsched.rothchild.me:8080/

Ruiter A. J., 2020, IAU Symposium, 357, 1

Ruiz-Lapuente P., Cappellaro E., Turatto M., Gouiffes C., Danziger I. J., della Valle M., Lucy L. B., 1992, ApJ, 387, L33

Sadeh I., 2019, in 36th International Cosmic Ray Conference (ICRC2019). p. 775 (arXiv:1902.03620)

Saha A., et al., 2016, in Peck A. B., Seaman R. L., Benn C. R., eds, Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series Vol. 9910, Observatory Operations: Strategies, Processes, and Systems VI. p. 99100F (arXiv:1611.05914), doi:10.1117/12.2232095

Saito T., Rehmsmeier M., 2015, PLoS One, 10(3)

Sako M., et al., 2008, AJ, 135, 348

Sako M., et al., 2011, ApJ, 738, 162

Savchenko V., et al., 2017, ApJ, 848, L15

Scargle J., 1982, ApJ, 263, 835

Schawinski K., et al., 2008, Science, 321, 223

Scolnic D., et al., 2014, ApJ, 795, 45

Scolnic D. M., et al., 2018, ApJ, 859, 101

Sell P. H., Maccarone T. J., Kotak R., Knigge C., Sand D. J., 2015, MNRAS, 450, 4198

Shappee B. J., et al., 2017, Science, 358, 1574

Shen K. J., Kasen D., Weinberg N. N., Bildsten L., Scannapieco E., 2010, ApJ, 715, 767

Silverman J. M., et al., 2012, MNRAS, 425, 1789

Smartt S. J., 2009, ARA&A, 47, 63

Smartt S. J., et al., 2017, Nature, 551, 75

Smartt S. J., et al., 2018, The Astronomer's Telegram, 11727, 1

Smith N., McCray R., 2007, ApJ, 671, L17

Smith K. W., et al., 2019, Research Notes of the American Astronomical Society, 3, 26

Soares-Santos M., et al., 2017a, ApJ, 848, L16

Soares-Santos M., et al., 2017b, ApJ, 848, L16

Solarz A., Bilicki M., Gromadzki M., Pollo A., Durkalec A., Wypych M., 2017, A&A, 606, A39

Sooknunan K., et al., 2018, preprint, (arXiv:1811.08446)

Soraisam M. D., et al., 2020, ApJ, 892, 112

Srivastava N., Hinton G., Krizhevsky A., Sutskever I., Salakhutdinov R., 2014, J. Mach. Learn. Res., 15, 1929

Stetson P. B., 1996, PASP, 108, 851

Sullivan M., et al., 2006, AJ, 131, 960

Sullivan M., et al., 2011, ApJ, 737, 102

Sutskever I., Vinyals O., Le Q. V., 2014, in Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2. NIPS'14. MIT Press, Cambridge, MA, USA, pp 3104–3112

Szegedy C., Zaremba W., Sutskever I., Bruna J., Erhan D., Goodfellow I. J., Fergus R., 2014, in Bengio Y., LeCun Y., eds, 2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings. http://arxiv.org/abs/1312.6199

Szegedy C., et al., 2015, in 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp 1–9, doi:10.1109/CVPR.2015.7298594

Tanvir N. R., et al., 2017, ApJ, 848, L27

Taubenberger S., 2017, The Extremes of Thermonuclear Supernovae. p. 317, doi:10.1007/978-3-319-21846-5_37

The PLAsTiCC team et al., 2018, preprint, (arXiv:1810.00001)

Tomaney A. B., Crotts A. P. S., 1996, AJ, 112, 2872

Tonry J. L., et al., 2018, PASP, 130, 064505

Troja E., et al., 2017, Nature, 551, 71

Valenti S., et al., 2017, ApJ, 848, L24

Verde L., Treu T., Riess A. G., 2019, Nature Astronomy, 3, 891

Verma A., Collett T., Smith G. P., Strong Lensing Science Collaboration the DESC Strong Lensing Science Working Group 2019, arXiv e-prints, p. arXiv:1902.05141

Villar V. A., Berger E., Metzger B. D., Guillochon J., 2017, ApJ, 849, 70

Villar V. A., et al., 2019, ApJ, 884, 83

Villar V. A., et al., 2020a, Light Curves of Pan-STARRS1 SN-like Transients, doi:10.5281/zenodo.3974950, https://doi.org/10.5281/zenodo.3974950

Villar V. A., Cranmer M., Contardo G., Ho S., Yao-Yu Lin J., 2020b, arXiv e-prints, p. arXiv:2010.11194

Villar V. A., et al., 2020c, ApJ, 905, 94

Villar V. A., Cranmer M., Berger E., Contardo G., Ho S., Hosseinzadeh G., Yao-Yu Lin J., 2021, arXiv e-prints, p. arXiv:2103.12102

Vincenzi M., Sullivan M., Firth R. E., Gutiérrez C. P., Frohmaier C., Smith M., Angus C., Nichol R. C., 2019, MNRAS, 489, 5802

Vincenzi M., et al., 2020, arXiv e-prints, p. arXiv:2012.07180

Webb S., et al., 2020, MNRAS, 498, 3077

Weinberg D. H., Mortonson M. J., Eisenstein D. J., Hirata C., Riess A. G., Rozo E., 2012, Physics Reports, Volume 530, Issue 2, p. 87-255., 530, 87

Welch D. L., Stetson P. B., 1993, AJ, 105, 1813

Wen Y., Vicol P., Ba J., Tran D., Grosse R., 2018, arXiv e-prints, p. arXiv:1803.04386

Whelan J., Iben Icko J., 1973, The Astrophysical Journal, 186, 1007

Yoon S.-C., Dessart L., Clocchiatti A., 2017, ApJ, 840, 10

Yu Y. W., Liu L. D., Dai Z. G., 2018, ApJ, 861, 114

Zampieri L., 2017, Light Curves of Type II Supernovae. p. 737, doi:10.1007/978-3-319-21846-5_26

Zhang X., Zhao J., LeCun Y., 2015, in Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1. NIPS'15. MIT Press, Cambridge, MA, USA, p. 649–657

de Jong R. S., et al., 2012, in McLean I. S., Ramsay S. K., Takami H., eds, Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series Vol. 8446, Ground-based and Airborne Instrumentation for Astronomy IV. p. 84460T (arXiv:1206.6885), doi:10.1117/12.926239

van Dyk S. D., Hamuy M., Filippenko A. V., 1996, AJ, 111, 2017

van Velzen S., et al., 2018, The Astronomer's Telegram, 12263

van der Maaten L., Hinton G. E., 2008, Journal of Machine Learning Research