Monthly Notices
of the
ROYAL ASTRONOMICAL SOCIETY

MNRAS **503,** 4838–4854 (2021) doi:10.1093/mnras/stab633
Advance Access publication 2021 March 15

Transient-optimized real-bogus classification with Bayesian convolutional neural networks – sifting the GOTO candidate stream

T. L. Killestein,[1]★ J. Lyman,[1] D. Steeghs,[1] K. Ackley,[2,3] M. J. Dyer,[4] K. Ulaczyk,[1] R. Cutter,[1] Y.-L. Mong,[2,3] D. K. Galloway,[2,3] V. Dhillon,[4,5] P. O'Brien,[6] G. Ramsay,[7] S. Poshyachinda,[8] R. Kotak,[9] R. P. Breton,[10] L. K. Nuttall,[11] E. Pallé,[5] D. Pollacco,[1] E. Thrane,[2,3] S. Aukkaravittayapun,[8] S. Awiphan,[8] U. Burhanudin,[4] P. Chote,[1] A. Chrimes,[1,12] E. Daw,[4] C. Duffy,[7] R. Eyles-Ferris,[6] B. Gompertz,[1] T. Heikkilä,[9] P. Irawati,[8] M. R. Kennedy,[10] A. Levan,[1,12] S. Littlefair,[4] L. Makrygianni,[4] D. Mata Sánchez,[10] S. Mattila,[9] J. Maund,[4] J. McCormac,[1] D. Mkrtichian,[8] J. Mullaney,[4] E. Rol,[2,3] U. Sawangwit,[8] E. Stanway,[1] R. Starling,[6] P. A. Strøm,[1] S. Tooke,[6] K. Wiersema[1] and S. C. Williams[9,13]

[1] *Department of Physics, University of Warwick, Gibbet Hill Road, Coventry CV4 7AL, UK*
[2] *School of Physics & Astronomy, Monash University, Victoria 3800, Australia*
[3] *OzGRav-Monash, School of Physics and Astronomy, Monash University, Victoria 3800, Australia*
[4] *Department of Physics and Astronomy, University of Sheffield, Sheffield S3 7RH, UK*
[5] *Instituto de Astrof'isica de Canarias, E-38205 La Laguna, Tenerife, Spain*
[6] *School of Physics & Astronomy, University of Leicester, University Road, Leicester LE1 7RH, UK*
[7] *Armagh Observatory & Planetarium, College Hill, Armagh BT61 9DG, UK*
[8] *National Astronomical Research Institute of Thailand, 260 Moo 4, T. Donkaew, A. Maerim, Chiangmai 50180 Thailand*
[9] *Department of Physics & Astronomy, University of Turku, Vesilinnantie 5, FI-20014 Turku, Finland*
[10] *Jodrell Bank Centre for Astrophysics, Department of Physics and Astronomy, The University of Manchester, Manchester M13 9PL, UK*
[11] *Institute of Cosmology and Gravitation, University of Portsmouth, Portsmouth PO1 3FX, UK*
[12] *Department of Astrophysics/IMAPP, Radboud University, NL-6500 GL, Nijmegen, the Netherlands*
[13] *Finnish Centre for Astronomy with ESO (FINCA), University of Turku, Quantum, Vesilinnantie 5, FI-20014 Turku, Finland*

Accepted 2021 March 1. Received 2021 February 19; in original form 2020 November 6

ABSTRACT

Large-scale sky surveys have played a transformative role in our understanding of astrophysical transients, only made possible by increasingly powerful machine learning-based filtering to accurately sift through the vast quantities of incoming data generated. In this paper, we present a new real-bogus classifier based on a Bayesian convolutional neural network that provides nuanced, uncertainty-aware classification of transient candidates in difference imaging, and demonstrate its application to the datastream from the GOTO wide-field optical survey. Not only are candidates assigned a well-calibrated probability of being real, but also an associated confidence that can be used to prioritize human vetting efforts and inform future model optimization via active learning. To fully realize the potential of this architecture, we present a fully automated training set generation method which requires no human labelling, incorporating a novel data-driven augmentation method to significantly improve the recovery of faint and nuclear transient sources. We achieve competitive classification accuracy (FPR and FNR both below 1 per cent) compared against classifiers trained with fully human-labelled data sets, while being significantly quicker and less labour-intensive to build. This data-driven approach is uniquely scalable to the upcoming challenges and data needs of next-generation transient surveys. We make our data generation and model training codes available to the community.

Key words: methods: data analysis – techniques: photometric – surveys.

1 INTRODUCTION

Transient astronomy seeks to identify new or variable objects in the night sky, and characterize them to learn about the underlying mechanisms that power them and govern their evolution. This variability can occur on time-scales of milliseconds to years, and at luminosities ranging from stellar flares to luminous supernovae that outshine their host galaxy (Kulkarni 2012; Villar et al. 2017). Through observations of optical transient sources we have obtained evidence of the explosive origins of heavy elements (e.g. Abbott et al. 2017b, Pian et al. 2017), traced the accelerating expansion of our Universe across cosmic time (e.g. Perlmutter et al. 1999), and located the faint counterparts of some of the most distant and energetic astrophysical events known: gamma-ray bursts (e.g. Tanvir

★ E-mail: t.killestein@warwick.ac.uk

et al. 2009). Requiring multiple observations of the same sky area to detect variability, transient surveys naturally generate vast quantities of data that require processing, filtering, and classification – this has driven the development of increasingly powerful techniques bolstered by machine learning to meet the demands of these projects.

Many of the earliest prototypical transient surveys began as galaxy-targeted searches, performed with small field-of-view instruments. In the early stages of these surveys candidate identification was performed manually, with humans 'blinking' images to look for varying sources. This process is time-consuming and error-prone, and represented a bottleneck in the survey dataflow which heavily limited the sky coverage of these surveys. The first 'modern' transient surveys (e.g. LOSS; Filippenko et al. 2001) used early forms of difference imaging to detect candidates in the survey data, automating the candidate detection process and enabling both faster response times and greater sky coverage. LOSS proved extremely successful, discovering over 700 supernovae in the first decade of operation, providing a homogeneous sample that has proven useful in constraining supernova rates for the local Universe (Leaman et al. 2011; Li et al. 2011).

Difference imaging has since emerged as the dominant method for the identification of new sources in optical survey data. With this method, an input image has a historic reference image subtracted to remove static, unvarying sources. Transient sources in this difference image appear as residual flux, which can be detected and measured photometrically using standard techniques. Various algorithms have been proposed for optical image subtraction, either attempting to match the point spread function (PSF) and spatially varying background between an input and reference image (Alard & Lupton 1998; Becker 2015), or accounting for the mismatch statistically (Zackay, Ofek & Gal-Yam 2016) to enable clean subtraction. Difference imaging also provides an effective way to robustly discover and measure variable sources in crowded fields (Wozniak 2000).

Driven by both improvements in technology (large-format CCDs, wide-field telescopes) and difference imaging algorithms, large-scale synoptic sky surveys came to the fore. In this mode, significant areas of sky can be covered each night to a useful depth and candidate transient sources automatically flagged. This has driven an exponential growth in discoveries of transients, with over 18 000 discovered in 2019 alone.[1] Wide-field surveys such as the Zwicky Transient Facility (ZTF; Bellm et al. 2019), PanSTARRS1 (PS1; Chambers et al. 2016), the Asteroid Terrestrial-impact Last Alert System (ATLAS; Tonry et al. 2018), and the All-Sky Automated Survey for SuperNovae (ASAS-SN; Shappee et al. 2014) have proven to be transformative, collectively discovering hundreds of new transients per night.

With the ability to repeatedly and rapidly tile large areas of sky in order to search for new and varying sources, the follow-up of optical counterparts to poorly localized external triggers became possible, in the process ushering in the age of multimessenger astronomy. An early example was detection of optical counterparts to *Fermi* gamma-ray bursts by the Palomar Transient Factory (PTF; Law et al. 2009). Typical localization regions from the *Fermi* GBM instrument (Meegan et al. 2009) were of order 100 square degrees at this time, representing a significant challenge to successfully locate comparatively faint ($r \sim 17$–19) GRB afterglows. Of the 35 high-energy triggers responded to, eight were located in the optical (Singer et al. 2015), demonstrating the emerging effectiveness of synoptic sky surveys for this work.

Another recent highlight has been the detection of an optical counterpart to a TeV-scale astrophysical neutrino detected by the IceCUBE facility (Aartsen et al. 2017). Recent and historical wide-field optical observations of the localization area combined with high-energy constraints from *Fermi* enabled the identification of a flaring blazar, believed to be responsible for the alert (IceCube-170922A; IceCube Collaboration 2018) . This rapidly increasing survey capability has culminated recently in the landmark discovery of a multimessenger counterpart to the gravitational wave (GW) event GW170817 (Abbott et al. 2017a, b).

1.1 Real-bogus classification

For many years, the rate of difference image detections generated per night by sky surveys has significantly exceeded the capacity of teams of humans to manually vet and investigate each one. This has motivated the development of algorithmic filtering on new sources, to reject the most obvious false positives and reduce the incoming datastream to something tractable by human vetting. With the growing scale and depth of modern sky surveys, simple static cuts on source parameters cannot keep pace with the rate of candidates, with high false positive rates leading to substantial contamination by artefacts. This situation has motivated the development of machine learning (ML) and deep learning (DL) classifiers, which can extract subtle relationships/connections between the input data/features and perform more effective filtering of candidates. The dominant paradigm for this task has so far been the real-bogus formalism (e.g. Bloom et al. 2012), which formulates this filtering as a binary classification problem. Genuine astrophysical transients are designated 'real' (score 1), whereas detector artefacts, subtraction residuals, and other distractors are labelled as 'bogus' (score 0). A machine learning classifier can then be trained using these labels with an appropriate set of inputs to make predictions about the nature of a previously unseen (by the classifier) source within an image.

This real-bogus classification is only one step in a transient detection pipeline. Having established the candidates appearing as astrophysically real sources, further filtering is required to determine if they are scientifically interesting, or distractors – the definition of 'interesting' is naturally governed by the science goals of the survey. This process draws in contextual information from existing catalogues, historical evolution, and more fine-grained classification routines. The last step before triggering follow-up and further study (at least currently) is human inspection of the remaining candidates. No single filtering step is 100 per cent efficient in removing false positives/low significance detections, thus human vetting is required to identify promising candidates and screen out any bogus detections that have made it this far. Real-bogus classification is the most crucial step, reducing the volume of candidates that later steps must process and the amount of bogus candidates that humans must eventually sift through to find interesting objects – a balance between sensitivity (to avoid missing detections irretrievably) and specificity (avoiding floods of low-quality candidates) must be reached.

Real-bogus classification is a well-studied problem, beginning with early transient surveys (Romano, Aragon & Ding 2006; Bailey et al. 2007), and evolving both in complexity and performance with the increasing demands placed on it by larger and deeper sky surveys such as PTF (Brink et al. 2013), PanSTARRS1 (Chambers et al. 2016), and the Dark Energy Survey (Goldstein et al. 2015). Early classifiers were generally built on decision tree-based predictors such as random forests (Breiman 2001), using a feature vector as input. Feature vectors comprise extracted information about a given candidate, and often include broad image-level statistics/descriptions

[1] https://wis-tns.org/

designed to maximally separate real and bogus detections in the feature space. Examples include the source full width at half-maximum computed from the 2D profile, noise levels, and negative pixel counts. More elaborate features can be composed via linear combinations of these quantities, which may exploit correlations and symmetries. Another method of deriving features is to compute compressed numerical representations of the source via Zernicke/shapelet decomposition (Ackley et al. 2019).

However, feature selection can represent a bottleneck to increasing performance. Features are typically selected by humans to encode the salient details of a given detection, attempting to find a compromise between classification accuracy and speed of evaluation. This introduces the possibility of missing salient features entirely, or choosing a suboptimal combination of them.

Directly using pixel intensities as a feature representation avoids choosing features entirely, instead training on flattened and normalized input images (Wright et al. 2015; Mong et al. 2020), these have demonstrated improved accuracy over fixed-feature classifiers. However, this approach quickly (quadratically) becomes inefficient for large inputs. Using a smaller input size means information on the surrounding area of each detection is unavailable, limiting the visible context and affecting classification accuracy as a result.

Recently, convolutional neural networks (CNNs, LeCun et al. 1995) have led to a paradigm shift in the field of computer vision and machine learning, which has been transformative in the way we process, analyse, and classify image data across all disciplines. CNNs use learnable convolutional filters known as kernels to replace feature selection. These filters are cross-correlated with the input images to generate 'feature maps', effectively compact feature representations. Through the training process, the filter parameters are optimized to extract the most salient details of the inputs, which can then be fed into fully connected layers to perform classification or regression. In this way, the model can select its own feature representations, avoiding the bottleneck of human selection. Multiple layers can be combined to achieve greater representational power, known as deep learning (LeCun, Bengio & Hinton 2015). Recent work using CNNs has demonstrated state-of-the-art performance at real-bogus classification (Cabrera-Vives et al. 2017; Gieseke et al. 2017; Duev et al. 2019; Turpin et al. 2020). CNNs are also efficiently parallelizable making them suitable for high-volume data processing tasks. While providing substantial accuracy improvements over previous techniques, deep learning is particularly reliant upon large and high-quality training sets to minimize overfitting, arising from the high number of model parameters. Although augmentation and regularization techniques can minimize this risk, they are no substitute for a larger data set. The performance of any classifier is ultimately limited by the error rate on the training labels, so it is important to also ensure the data set is accurately labelled. Making a large, pure, and diverse training set can be among the most challenging parts of developing a machine learning algorithm, and significant effort has been focused on this area in recent years.

Traditionally the 'gold-standard' for machine learning data sets across computer science and astronomy has been human-labelled data, as this represents the ground truth for any supervised learning task. Use of citizen science has proven to be particularly effective, leveraging large numbers of participants and ensembling their individual classifications to provide higher accuracy training sets for machine learning through collaborative schemes such as Zooniverse (Lintott et al. 2008; Mahabal et al. 2019). However, even in large teams, human labelling of large-scale data sets is time-consuming and inefficient requiring hundreds–thousands of hours spent collectively to build a data set of a suitable size and purity. Specifically for real-bogus classification, there are also issues with completeness and accuracy for human labelling of very faint transients close to the detection limit. These faint transients are where a classifier has potential to be the most helpful, so if the training set is fundamentally biased in this regime, any classifier predictions will be similarly limited. To go beyond human-level performance, we cannot solely rely on human labelling, additional information is required. One specific aspect of astronomical data sets that can be leveraged to address both issues discussed above is the availability of a diverse range of contextual data about a given source. Sizeable catalogues of known variable stars, galaxies, high-energy sources, asteroids, and many other astronomical objects are freely available and can be queried directly to identify and provide a more complete picture of the nature of a given source.

Significant effort is being invested in data processing techniques for transient astronomy in anticipation of the Vera C. Rubin Observatory (Ivezić et al. 2019), due to begin survey operations in 2022. Via the Legacy Survey of Space and Time (LSST), the entire southern sky will be surveyed down to a depth of $r^{'} \sim 24.5$ in five colours at high cadence, providing an unprecedented discovery engine for transients to depths previously unprobed at this scale. The dataflow from this project is expected to be a factor 10 greater than current transient surveys, and promises to be transformative in the fields of supernova cosmology, detection of potentially hazardous near-Earth asteroids, and mapping the Milky Way in unprecedented detail. The main high-cadence deep sky survey promises to provide a significant increase in the number of genuine transients we detect, but also a significant increase in the number of bogus detections assuming there are not similarly large improvements in the capability of machine learning-based filtering techniques. Development of higher performance classifiers is crucial to fully exploit this stream, but also more granular classification involving contextual data (as recently demonstrated by Carrasco-Davis et al. 2020) to ensure that novel and scientifically important candidates are identified promptly enough to be propagated to teams of humans and followed up.

A related goal of increasing importance in the big data age of the Rubin Observatory and similar projects is that of quantifying uncertainty – being able to identify detections that the classifier is confident are real, and providing a classifier a way to indicate uncertainty on more tenuous examples. This objective goes beyond the simple value of the real-bogus score, and can then be used to find the optimal edge cases to feed to human labellers, allowing new data to be continually integrated to improve performance and keep the classifier's knowledge current and applicable to a continuously evolving set of instrumental parameters. Current generation transient surveys provide a crucial proving ground for development of these new techniques.

1.2 The Gravitational-Wave Optical Transient Observer (GOTO)

The Gravitational-Wave Optical Transient Observer (Steeghs et al., in preparation) is a wide-field optical array, designed specifically to rapidly survey large areas of sky in search of the weak kilonovae and afterglows associated with GW counterparts. The work we present in this paper was conducted during the GOTO prototype stage, using data taken with a single 'node' of telescopes situated at the Roque de los Muchachos observatory on La Palma. Each node comprises eight co-mounted fast astrograph optical tube assemblies (OTAs) combining to give a ∼40 square degree field of view in a single pointing. GOTO performs surveys using a custom wide L band filter (approximately equivalent to $g^{'} + r^{'}$) down to $L \approx 20$, providing an effective combination of fast and deep survey capability uniquely suited to tackling the challenging large error boxes associated with GW detections. As demonstrated in Gompertz et al. (2020), the prototype GOTO installation is capable of conducting sensitive searches for the optical counterparts of nearby binary neutron star mergers, even with weak localizations of ∼1000 square degrees. When not responding to GW events, GOTO performs an all-sky survey utilizing difference imaging to search for other interesting transient sources. Although the GOTO prototype datastream will be the primary data source used to investigate the performance of the machine learning techniques developed in this paper, the methods are inherently scalable and will also be deployed for the future GOTO datastream from four nodes spread over two sites. For now, we concentrate on a calendar year of prototype operations (spanning 2019-01-01–2020-01-01) – which represents a significant data set, comprising 44 789 difference images in total.

Raw images are reduced with the GOTO pipeline (Steeghs et al., in preparation). Here, we provide a very brief overview of the process for context, and delegate more in-depth discussion to the specific upcoming pipeline papers. The typical survey strategy for GOTO is three exposures per pointing, which undergo standard bias, dark and flat correction, and then are median-combined to reject artefacts and improve depth. Throughout this paper, we refer to this median-combined stack of subframes as a ‘science image’. Each combined image is matched to a reference template, which passes basic quality checks, and aligned using the SPALIPY[2] code. Image subtraction is performed on the aligned science and reference images with the HOTPANTS algorithm (Becker 2015) to generate a difference image. To locate residual sources in the difference image, source extraction is performed using SEXTRACTOR (Bertin & Arnouts 1996). Detections in the difference image are referred to as ‘candidates’ through the remainder of this paper. For each candidate, a set of small stamps are cut out from the main science, template and difference images and this forms the input to the GOTO real-bogus classifier. This process and proposed improvements are discussed in more detail in Section 2.1. From here, candidates that pass a cut on real-bogus score (using a preliminary classifier) are ingested into the GOTO Marshall – a central website for GOTO collaborators to vet, search, and follow up candidates (Lyman et al., in preparation).

In line with the principal science goals of the GOTO project, the real-bogus classifier discussed in this work is constructed specifically to maximize the recovery rate of extragalactic transients and other explosive events such as cataclysmic variable outbursts. Small-scale stellar variability can be easily detected via difference imaging, but is better studied through the aggregated source light curves. An operational requirement for the current version of this classifier is the ability to perform consistently across multiple different hardware configurations. During classifier development, the GOTO prototype used two different types of optical tube design, each with varying optical characteristics that led to different PSFs, distortion patterns, and background levels/patterns. Due to limited data availability, training a classifier for each individual OTA (or group of OTAs of the same type) was not viable. This requirement adds an additional operational challenge over survey programs such as the Zwicky Transient Facility (ZTF, Bellm et al. 2019) and PanSTARRS1 (PS1, Chambers et al. 2016), which use a static, single-telescope design. If acceptable results can be achieved with this heterogeneous hardware configuration, then further performance gains can be expected when the design GOTO hardware configuration is deployed. This will use telescopes of consistent design and improved optical quality meaning less model capacity needs to be directed towards making the classification performance stable and across a diverse ensemble of optical distortions.

In this paper, we propose an automated training set generation procedure that enables large, minimally contaminated, and diverse data sets to be produced in less time than human labelling and at larger scales. This procedure also introduces a data-driven augmentation scheme to generate synthetic training data that can be used to significantly improve the performance of any classifier on extragalactic transients of all types, but with particular effectiveness for nuclear transients. Using this improved training data, we apply Bayesian convolutional neural networks (BCNNs) to astronomical real-bogus classification for the first time, providing uncertainty-aware predictions that measure classifier confidence, in addition to the typical real-bogus score. This opens up promising future directions for more complex classification tasks, as well as optimally utilizing the predictions of human labellers. We emphasize that although this classifier is discussed in the context of GOTO and our associated science needs, the techniques discussed are fully general and could be applied to general real-bogus classification at other projects easily. Our code, GOTORB, is made freely available online[3] with this in mind.

2 TRAINING SET GENERATION AND AUGMENTATION

The ‘real’ content of our training set is composed of minor planets, similar to Smith et al. (2020). Assuming the sky motion is large (but not so large that the source is trailed) these objects are typically detected in the science image but not the template image, which provides a clean subtraction residual resembling an explosive transient. Due to the large pixels of the GOTO detectors and short exposure times of each sub-image, very few asteroids move sufficiently quickly to trail. We estimate that sky motions of 1 arcsec per minute or greater will lead to trailing.

There are significant numbers of asteroids detectable down to $L \sim 20.5$ with GOTO, and the sky motion ensures that a diverse range of image configurations are sampled. With the large ∼40 square degree field of view provided by GOTO, a whole-sky average of 4.6 asteroids per pointing are obtained, with this number significantly increasing towards the ecliptic plane. Using ephemerides provided by the `astorb` database (Moskovitz et al. 2019), based on observations reported to the Minor Planet Center,[4] difference image detections can be robustly cross-matched to minor planets in the field. This provides a significant pool of high-confidence, unique, and diverse difference image detections from which to build a clean training set.

We use the online SkyBoT cone search (Berthier et al. 2006, 2016) to retrieve the positions and magnitudes of all minor planets within the field of view of each GOTO image, then cross-match this table with all valid difference image detections using a 1 arcsec threshold value to identify the asteroids present in the image. The ephemerides provided are of sufficient quality that this is adequate to match even faint ($L \sim 20$) asteroids. To avoid spurious cross-matches, only asteroids brighter than the 5σ limiting magnitude of the image are considered. An alternative offline cone search is made accessible via the PYMPC package[5] Python package, which the code

[2] https://github.com/Lyalpha/spalipy

[3] https://github.com/GOTO-OBS/gotorb

[4] https://www.minorplanetcenter.net/

[5] https://pypi.org/project/pympc/

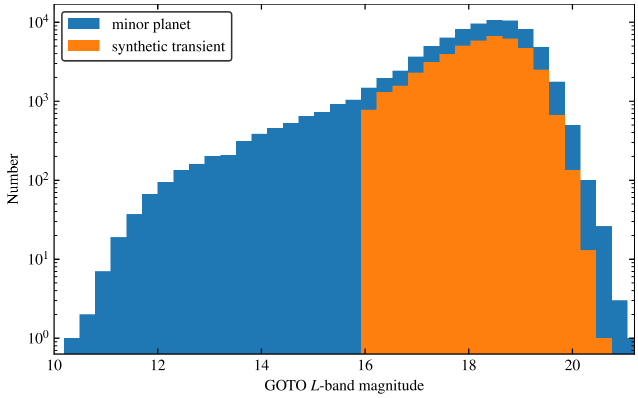

Figure 1. Magnitude distribution of the minor planets (MP) used to build our training set. Bright-end number densities are dominated by the true magnitude distribution of the minor planets, where the faint-end density is constrained by the GOTO limiting magnitude. The magnitude distribution of synthetic transients (SYN) is a subsample of the minor planet magnitude distribution, except with a cut at $L \sim 16$, to avoid unrealistically bright objects.

can fall back on if SkyBoT is unavailable. Using minor planets, the training set can reliably be extended to fainter magnitudes, where the performance of human vetters begins to significantly decrease. Fig. 1 illustrates the magnitude distribution of minor planets used to construct the training set.

To create the bogus content of our training set, we randomly sample detections in the difference image following Brink et al. (2013). Bogus detections overwhelmingly ($\gtrsim$ 99 per cent) outnumber real detections in each difference image, so it is justified to sample in this way. One significant source of contamination taking this approach is variable stars, therefore we remove all known variable stars from the random bogus component by cross-matching against the ATLAS Variable Star Catalogue (Heinze et al. 2018) with a 5 arcsec radius. These variable star detections can constitute 2–4 per cent of the entire bogus data set. Of the detections removed by this step, a small fraction of these will be high-amplitude variable stars which have a strong subtraction residual in a given night's data, and thus represent real sources lost. Automating the correct labelling of these sources using light curve information is feasible, but would add significant complexity and more potential failure modes, so we instead opt to remove the variable stars entirely and simply add more verifiably 'real' detections in their place in the form of more minor planets. Inevitably, some small fraction of uncatalogued variable stars will be missed with this procedure, and we develop tools to identify them retrospectively after model training in Section 3.3.

To improve the classifier's resistance to specific challenging subtypes of data poorly represented in our algorithmically generated training set, we inject human-labelled detections into the data set. More specifically, candidates from the GOTO Marshall (discussed in full in Lyman et al., in preparation) are included, which were misidentified by the classifier in the pipeline at the time as real and later labelled as bogus by human vetters. The previous classifier was a rapidly deployed prototype CNN similar in design to that presented here, trained on a smaller data set of minor planets and random bogus detections. These detections are included to allow the classifier to screen out artefacts missed by the prototype image processing pipeline, including satellite trails and highly wind-shaken PSFs. This artificially increases the diversity of the bogus component of the training set, as these edge-case detections would rarely be selected by naive random sampling and so be poorly represented within the model. Although these detections represent a small fraction of the overall training set ($\sim$ 5 per cent), they provide a marked improvement in performance in the real-world deployment of the classifier, including marginal gains on more typical detections.

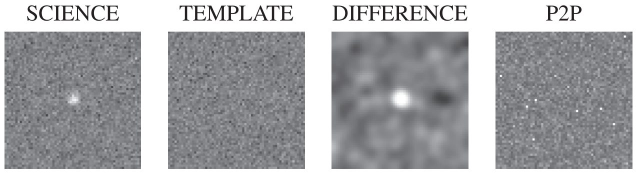

Figure 2. Example data format for a set of idealized synthetic images of a single Gaussian source newly appearing in the science image. We apply a naive convolution of science image with template PSF and vice versa in producing the difference image for visualization purposes. From left to right: science median, template median, difference image, pixel-wise peak-to-peak variation across contributing images to science median. Cut-outs are 55 × 55 pixels square, corresponding to a side length of 1.1 arcmin.

2.1 Data extraction and format

For each detection identified for inclusion in our training/validation/test sets, a series of stamps are cut out from the larger GOTO image centred on the difference image residual. In common with previous CNN-based classifiers, we use small cut-outs of the median-stacked science and template images, as well as the resultant difference image after image subtraction. The size of these stamps is an important model hyperparameter, which we explore in more detail in Section 3.1. A example of the model inputs for a synthetic source are illustrated in Fig. 2.

An important addition to our network's inputs compared to previous work is a peak-to-peak (p2p) layer. This is included to characterize variability across the individual images that make up a median stacked science image, and is calculated as the peak-to-peak (maximum value − minimum value) variation of each pixel computed across all individual images that composed the median stack. To ensure consistent alignment across all individual stamps and remove any jitter, we cut out the region based on the RA/Dec. coordinates of the source detection in the median stack. This additional provides an effective discriminator for spurious transient events such as cosmic ray hits and satellite trails. If sufficiently bright, these are not removed by the simple median stacking in the current pipeline due to the small number of subframes used. This is particularly problematic for cosmic ray hits which are convolved with a Gaussian kernel for image subtraction, and appear PSF-like in the difference image. This can create convincing artefacts which are difficult to identify without access to the individual image level information. In testing, this reduced the false positive rate on the test set by $\sim$ 0.2 per cent. Although this is not a sizeable improvement when evaluated on the full data set, cosmic ray hits constitute a very small percentage of overall detections. Testing instead on a human-labelled set of bogus detections which were initially scored as real by the existing deployed classifier (without a p2p layer), there is a 2–3 per cent decrease in false positive rate.

For all of the above steps, stamps extending beyond the edge of the detector have missing areas filled in with a constant intensity level of 10^{-6}, to distinguish them quantitatively from masked (i.e. saturated) pixels which are assigned a value of zero in the difference image by the pipeline. The specific intensity level chosen for this offsetting is not important, and we choose our value to be well above machine precision (significant enough to influence the gradients) but well below the typical background level. To ensure that the classifier remains numerically stable in later training steps, each stack of stamps undergoes layer-wise L2 normalization to reduce the input's magnitude. Each stamp has the mean subtracted and is then divided through by the L2 ($\sqrt{\vec{x}\cdot\vec{x}}$) norm.

2.2 Synthetic transients

Although asteroids provide a convenient source of PSF-like residuals to train on, it is important to note that they cannot fully replicate genuine transients. Asteroids are markedly simpler to learn and discriminate for a classifier since they lack the complex background of a host galaxy. The main goal of this classifier is to detect extragalactic transients, so adapting the training set to maximize performance on these objects is important. An ideal approach would be to add a large number of genuine transients into the training set. However, GOTO has not been on-sky long enough to collect a suitably large set of these detections, and we only build the training set from the previous year of data. Even assuming every supernova over the past year is robustly detected in our data this will still yield a number of transients that is significantly less than the target size of our training set. This would create a severely imbalanced data set, which could in principle be used but with reduced classification performance. Using spectroscopically confirmed transients may also inject an element of observational bias into our training set, as events that have favourable properties for spectroscopy (in nearby galaxies, offset from their host, bright) are preferentially selected (Bloom et al. 2012) to be followed up. Instead, we reserve a set of real, spectroscopically confirmed transients GOTO has detected (∼900 as of 2020 August) for benchmarking purposes, as they represent a valuable insight into real-world performance and can be used to directly evaluate the effectiveness of any transient augmentation scheme we employ, as in Section 4.2.

PSF injection has been used heavily in prior work to generate synthetic detections for testing recovery rates and simulating the feasibility of observations. This process can be computationally intensive, involving construction of an effective PSF (ePSF) from combining multiple isolated sources or fitting an approximating function (e.g. a Gaussian) to sources in the image. The ePSF model can then be scaled and injected into to the image to simulate a new source. By injecting sources in close proximity to galaxies in individual images then propagating this through the data reduction pipeline, synthetic transients could be generated in a realistic way. However, the fast optical design of GOTO makes this a complex task, as the PSF varies as a function of source position on the detector. Sources in the corners of an image display mild coma, which, combined with wind-shake and other optical distortion, can lead to unusual PSFs that are not accurately reproduced by the mean PSF. In principle, this could be accounted for by computing PSFs for subregions of a given image or assuming some spatially varying kernel to fit for, but this would add sizeable overheads to the injection process and will always be an approximation.

Recent new techniques such as generative adversarial networks (GANs, Goodfellow et al. 2014) have shown promise in generating novel training examples that can be used to address class imbalances/scarcity in training sets (Mariani et al. 2018), and have recently started to be applied to astrophysical problems (Yip et al. 2019). However, these networks are computationally expensive, complex to train and understand the outputs of, and do not fully remove the need for large data sets. A robust human-interpretable method for generating synthetic examples is a better approach for the noisy, diverse data sets used in real-bogus classification.

We propose a novel technique for synthesizing realistic transients that can be used to significantly improve transient-specific performance when compared to a pure minor planet training set, without requiring PSF injection or other CPU-intensive approaches. For each minor planet detected in an image, the GLADE galaxy catalogue (Dálya et al. 2018) is queried for nearby galaxies within a set angular distance of 10 arcmin, chosen such that the PSF of sources within this region are consistent. Pre-built indices are used via CATSHTM (Soumagnac & Ofek 2018) to accelerate querying GLADE. The algorithm chooses the galaxy with the brightest galaxy (minimum B-band magnitude) within range, then generates a cut-out stamp with a randomly chosen x, y offset relative to the galaxy centre. For the implementation within this work, the x, y pixel offsets are drawn from a uniform distribution $U(-7.7)$ chosen to fully cover the range of offsets for nearby galaxies. Sources that are completely detached from any host galaxy are better represented by the minor planet component of the training set. This ensures that a diverse range of transient configurations (nuclear, offset, orphaned) are sampled. The minor planet and galaxy stamp are then directly summed to produce the synthetic transient. For the purposes of real-bogus classification, accurately matching the measured transient host-offset distribution is not crucial. The host offset distribution contains implicit and difficult to quantify biases resulting from the specific selection functions of the transient surveys that populate it – it does not reflect accurately the underlying distribution of astrophysical transients. By choosing from a uniform distribution, we instead aim to attain consistent performance across a wide range of host offsets that overlap with the range inferred from the transient host offset distribution.

The original individual images for each component are retrieved to correctly compute the peak-to-peak variation of the combined stamp. Model inputs are pre-processed and undergo L2 normalization (as discussed in Section 2.1) prior to training and inference, so additional background flux introduced by this method does not affect the model inputs. The noise characteristic of this combined stamp is not straightforward to compute due to the highly correlated noise present in the difference image and varying intensity levels, and could be higher or lower depending on the specific stamps – with the straightforward Gaussian case providing a $\sqrt{2}$ reduction in noise. This is likely not problematic for the classifier, providing a form of regularization that could improve generalization accuracy. We also assume that the spatial gradients in background across both stamps are approximately constant, as the stamp scale is far smaller than the overall frame scale – naturally this breaks down in the presence of nebulosity/galaxy light but this represents a overwhelmingly small fraction of the sky. We also reject all minor planets with $L < 16$, as these are significantly brighter than the selected host galaxy so are better represented by the pure minor planet candidates. This also cuts down significantly on saturated detections of dubious quality. This choice has no detrimental effect on bright-end performance, as discussed in Section 4. A random sample of synthetic transients generated with this approach is shown in Fig. 3. Our method bears some similarity in retrospect to the approach of Cabrera-Vives et al. (2017), who added stamps from the science image into difference images to simulate detections in 'random' locations. Our approach uses confirmed difference image detections of MPs and puts them in more purposeful locations, while preserving the noise characteristics of the difference stamp.

This approach has strong advantages over simply injecting transients into galaxies. By selecting only galaxies close to each minor planet, the PSF is preserved and is consistent, regardless of how distorted it may be. Injection-based methods require estimation/assumption of the image PSF, which is typically a parametrized

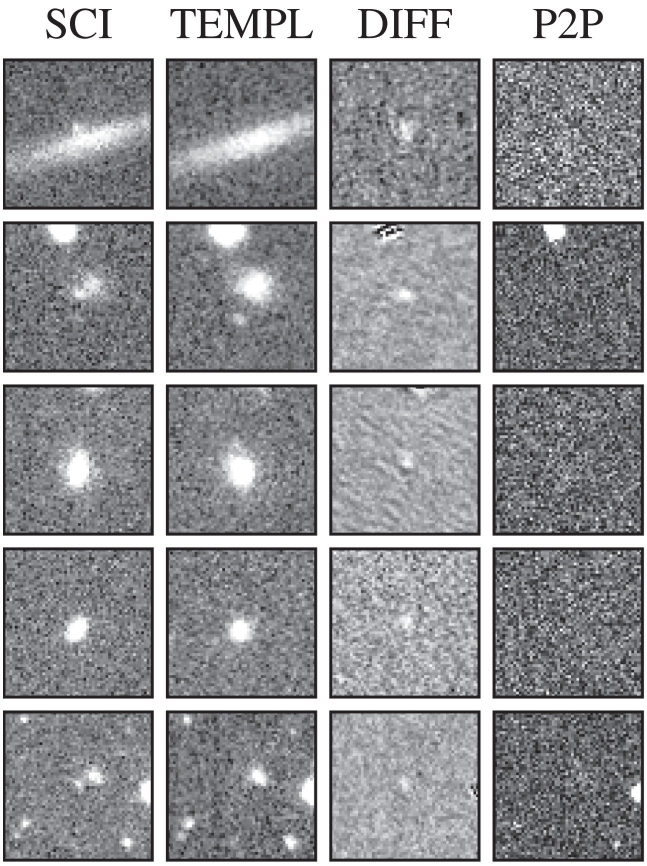

Figure 3. Randomly selected sample of synthetic transients generated with our algorithm, displayed in the same format as in Fig. 2. Significant variations in the PSF are visible due to sampling directly from the image, improving classifier resilience.

function determined by fitting isolated sources. Given the variation in PSF across images and across individual unit telescopes, this would be a computationally intensive task, and would likely lead to poorer results compared to using minor planets. However, using only these synthetic transients introduces unintended behaviour in the trained model that significantly degrades classification performance if not remedied. Since every synthetic transient in the training set is associated with a host galaxy by design, the model will over time learn to associate all detections with galaxies as being real as there is no loss penalty for doing so. To resolve this, we also inject galaxy residuals as bogus detections, randomly sampling from the remaining GLADE catalogue matches at a 1:1 transient:galaxy residual ratio. This way, the model learns that the salient features of these detections are not the galaxy, but the PSF-like detection embedded in them.

2.3 Training set construction

Using the techniques developed in the sections above, we build our training set with GOTO prototype data from 2019-01-01 to 2020-01-01. This ensures that our performance generalizes well across a range of possible conditions – with PSF shape and limiting magnitude being the most important properties that benefit from this randomization. A breakdown of training set proportions and properties is given in Table 1.

Our code is fully parallelized at image level, meaning that a full training set of ∼400 000 items can be constructed in under 24 h on a 32-core machine. Training sets can also be easily accumulated on multiple machines and then combined thanks to the use of the HDF5 file format. The main bottlenecks of training set generation are IO-related – loading in image data to prepare the stamps, and querying the GLADE catalogue and SkyBoT cone search.

Table 1. Breakdown of the composition of our data set, partitioned according to training and test sets. The validation data set is not shown, but is composed of 10% of the training data set, chosen randomly at training time.

Metalabel	Train	Test	
Minor planet	72 992	8133	
Synthetic transient	40 192	4521	
Random bogus	177 556	19 645	
Galaxy residual	28 040	3190	
Marshall bogus	24 577	2662	
Total	343 357	38 151	381 508

3 CLASSIFIER ARCHITECTURE

As a starting point, we follow the BRAAI classifier of Duev et al. (2019) in using a downsized version of the VGG16 CNN architecture of Simonyan & Zisserman (2014). This network architecture has proven to be very capable across a variety of machine learning tasks, and is a relatively simple architecture to implement and tweak. This architecture uses conv-conv-pool blocks as the primary component – two convolutions are applied in sequence to extract both simple and compound features, then the resultant feature map is reduced in size by a factor 2 by 'pooling', taking the maximum value of each 2 × 2 group of pixels. This architecture also uses small kernels (3 × 3) for performance. These structures are illustrated in Fig. 4. We use the configuration as presented in Duev et al. (2019) for development, but later conduct a large-scale hyperparameter search to fine-tune the performance to our specific data set (Section 3.1). The primary inputs to the classifier are small cut-outs of the science, template, difference, and `p2p` images as discussed in Section 2.1, which we refer to as stamps.

The sample weights for real and bogus examples are adjusted to account for the class imbalance in our data set, set to the reciprocal of the number of examples with each label. Class weights are not adjusted on a per-batch basis, as our training set is only mildly imbalanced. For regularization, we apply a penalty to the loss based on the L2 norm of each weight matrix. This penalizes exploding gradients and promotes stability in the training phase. L1 regularization was trialled but did not produce significantly better results. We also use spatial dropout (Tompson et al. 2015) between all convolutions which provides some regularization, but primarily is used for the purposes of uncertainty estimation (see Section 3.3) – a small dropout probability of ∼0.01 is found to be optimal from work in Section 3.1. Due to the significant training set size and our use of augmentation, very little regularization is needed for a model of this (comparatively) low complexity.

To further increase the effective size of our training set, we randomly augment training examples with horizontal and vertical flips, which provide a factor 4 increase in effective training set size over unaugmented stamps. We also trialled the usage of 90 deg rotations following (Dieleman, Willett & Dambre 2015), which do not require interpolations and thus do not introduce spurious artefacts that could add additional learning complexity. In contrast to other works (Cabrera-Vives et al. 2017; Reyes et al. 2018), we find consistent performance (over multiple training runs) with simple

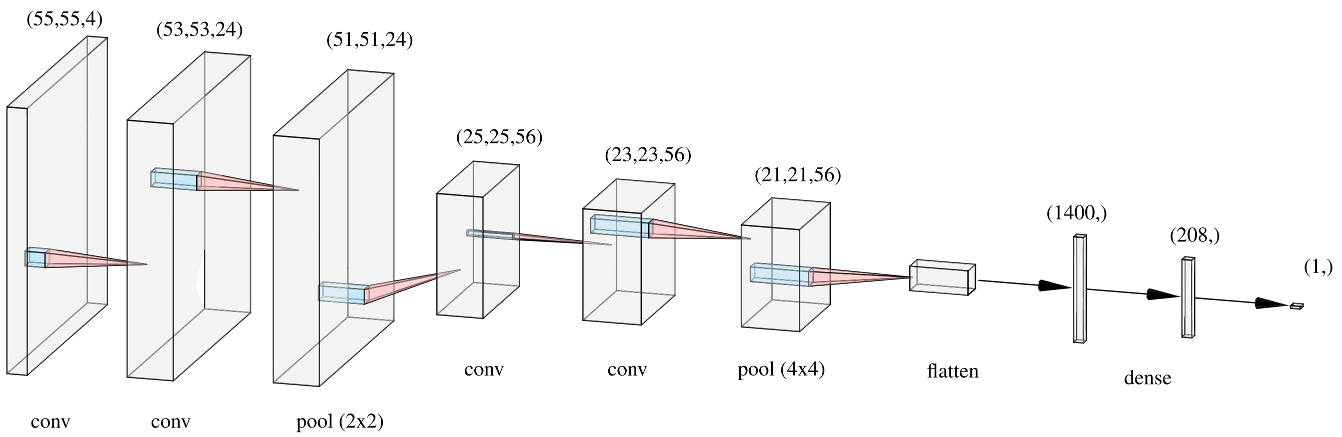

Figure 4. Block schematic of the optimal neural network architecture found by hyperparameter optimization in Section 3.1. Each block here represents a 3D image tensor, either as input to the network, or the product of a convolution operation generating an 'activation map'. Classification is performed using the scalar output of the neural network. Directly above each 3D tensor block the dimensions in pixels are shown, along with the operation that generates the next block below it represented by the coloured arrow. Not illustrated for clarity here are the dropout masks applied between each layer and the activation layers. Base figure produced with NNSVG (LeNail 2019).

reflections – potentially having already reached the saturation region of the learning curve.

Our model is implemented with the KERAS framework (Chollet et al. 2015), running with an optimized build of the TENSORFLOW backend (Abadi et al. 2015). For parameter optimization we use the ADAM optimizer of Kingma & Ba (2014), which provides reliable convergence, and use the binary cross-entropy as the loss function. To avoid overfitting, we utilize an early stopping criterion conditioned on the validation data set loss – if there has been no decrease in validation loss within 10 epochs, the model training is terminated. We perform model training and inferencing on CPU only, to mirror the deployment architecture used in the main GOTO pipeline. Using a single 32-core compute node, training the finalized model to early-stopping at ∼170 epochs takes around 10 h. Inferencing is significantly quicker, with an average throughput of 7500 candidates per second with no model ensembling performed. Our model training code is freely available via the `gotorb` Python package,[6] which includes the full range of tunable parameters and model optimizations we implement.

3.1 Tuning of hyperparameters/training set composition

To achieve the maximum performance possible with a given neural network, we conduct a search over the model hyperparameters to assess which combinations lead to the best classification accuracy and model throughput. Initially the ROC-AUC score (Fawcett 2006) was used as the metric to optimize as in many cases this is a more indicative performance metric than others, however this did not translate directly to improvements in classification performance. We conjecture this may be due to the score-invariant nature of the ROC-AUC statistic – it only captures the probability that a randomly selected real example will rank higher than a randomly selected bogus example, which is independent of the specific real-bogus threshold chosen. We instead opt to use the accuracy score, as this directly maps to the quantity we want to maximize in our model.

Data-based hyperparameters (training set composition, stamp size, data augmentation) are optimized iteratively by hand due to computational constraints. An approximate real-bogus ratio between 1:2 and 1:3 was found to be optimal, with greater values giving better bogus performance at the cost of recovery of real detections – we opt for 1:2 in the final data set. The overall data set size was found to be the biggest determinant of classification accuracy, with larger data sets showing improved performance – although this increase was subject to diminishing returns with larger and larger data sets. We chose a training set of O(4×10^5) examples, as this was roughly the largest data set we could fit into RAM on training nodes – naturally this could be increased further by reading data from disc on demand, but given CPUs were used for training there was a need to minimize input pipeline latencies as much as possible to compensate. Model performance was found to be relatively insensitive to the ratio of synthetic transients to minor planets, as long as there were at least 10 000 of both in the training set. Using a data set where 100 per cent of the real content came from minor planets led to a ∼ 5 per cent drop in the recovery rate of transients on the test set (see Fig. 11), whereas a 100 per cent synthetic transient data set led to a detrimental 15 per cent decrease in the recovery rate of minor planets, and a 5 per cent drop on the transient test set. This surprising result implies that combining both minor planets and synthetic transients has a synergistic effect, with the combination providing better performance overall. The specific composition of the final data set is listed in Table 1; we found a roughly 2:1 minor planet:synthetic transient ratio to provide the correct balance between overall test set performance and sensitivity to astrophysical transients.

A key parameter explored as part of this study is the input stamp size. Larger stamps take longer to generate and more time to perform inference on, so identifying the minimum stamp size possible without affecting performance is crucial. In Fig. 5, we show the results of training identical models on an identical 330k-example data set, with varying stamp size between 21 and 63 pixels. We find that there is no significant increase in performance for our training data set beyond a stamp size of 55 pixels. The upper limit of this search was set by available RAM, and took 118 h of compute time to complete. When scaled through by the ratio of the GOTO/ZTF plate scales (1.4×), our best value of 55 pixels appears remarkably consistent with the 63 pixel stamps that Duev et al. (2019) found optimal for their network. This is an interesting result, and could imply that the angular scale is actually the more relevant parameter – this might

[6] https://github.com/GOTO-OBS/gotorb

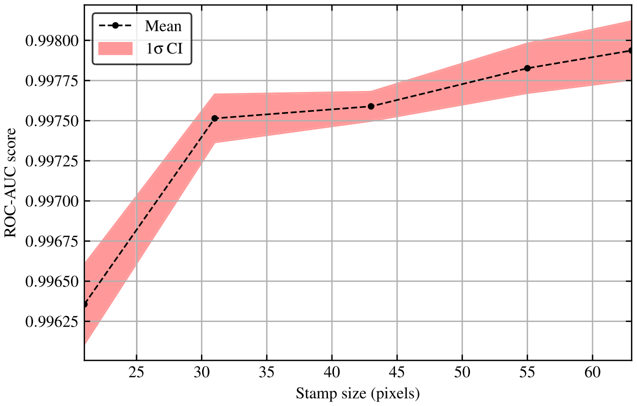

Figure 5. Classifier performance on the test set of a 330 000 example training set as a function of input stamp size. Each point is the average of three independent training runs on the same input training set, with the shaded region representing the 1σ confidence interval.

Table 2. Hyperparameter space over which the optimization search was conducted, split by numerical and categorical variables. The final adopted values are given in the rightmost column.

continuous				
Hyperparameter	Min	Max	Prior	Selected
Block 1 filters (N_1)	8	32	linear	24
Block 2 filters (N_2)	N_1	64	linear	56
$N_{\rm fc}$	64	512	linear	208
Dropout rate	10^{-2}	0.5	log	5.2×10^{-2}
Learning rate	10^{-5}	10^{-2}	log	6×10^{-5}
Regularizer penalty	10^{-8}	10^{-2}	log	2.0×10^{-8}
discrete				
Hyperparameter	Choice			Selected
Kernel initializer	He, Glorot			Glorot
Kernel regularizer	L1, L2			L2
Activation function	ReLU, LeakyReLU, ELU			LeakyReLU

represent some characteristic length-scale that encodes the optimal amount of information about the candidate and surrounding context without including too much irrelevant data.

Network hyperparameters are optimized using the `Hyperband` algorithm (Li et al. 2017) as implemented in the KERAS-TUNER package (O'Malley et al. 2019). This algorithm implements a random search, with intelligent allocation of computational resources by partially training brackets of candidate models and only selecting the best fraction of each bracket to continue training. In testing, this consistently outperformed both naive random search and Bayesian optimization in terms of final performance. Table 2 illustrates the region of (hyper)parameter space we choose to conduct our search over. The upper limits for the neuron/filter parameters are set by purely computational constraints – networks above this threshold take too long to evaluate and train, and so are excluded. We also set an upper limit of 500 000 on the number of model parameters to avoid overly complex models and promote small but efficient architectures. Based on initial experimentation, we require the number of convolutional filters in the second block must be greater than or equal to the number in the first block. This ensures that the largest (and most computationally expensive) convolution operations are performed on tensors that have been max-pooled and thus are smaller, reducing execution time. To maximize performance across all possible deployment architectures, the number of convolutional filters and fully connected layer neurons are constrained to be a multiple of 8. This is one of the requirements for fully leveraging optimized GPU libraries (such as cuDNN, Chetlur et al. 2014), and also enables use of specialized hardware accelerators such as tensor cores in the future. Conveniently, this discretization also makes the hyperparameter space more tractable to explore.

This search took around 1 month to complete on a single 32-core compute node, and sampled 828 unique parameter configurations. The three top-scoring models were then retrained from random initialization through to early stopping to validate their performance, and confirm that the hyperparameter combination led to stable and consistent results. The top three scoring models achieved accuracies on the hyperparameter validation set of 98.88, 98.64, and 98.54 per cent, respectively. Some of the candidate models had to be pruned from the list due to excessive overfitting. The best model was then selected based on the minimum test set error. Our final model achieved a test set class-balanced accuracy of 98.72 ± 0.02 per cent (F1 score 0.9826 ± 0.0003), with the selected hyperparameters listed in Table 2. This outperforms the version human-tuned by the authors through iterative improvement by 0.6 per cent, and trains to convergence in around half the number of epochs. We adopt this model architecture going forward, and characterize the overall performance in greater detail in Section 4. For this final model, the theoretical maximum ROC-AUC is obtained when the real-bogus threshold is set to 0.4, although in live deployment we opt for a conservative value of 0.8 to minimize contamination.

3.2 Quantifying classification uncertainty

Uncertainty estimation in neural networks is an open problem, but is of critical importance for a range of applications. Traditional deterministic neural networks output a single score per class between 0 and 1. This single value would be sufficient to provide a measure of confidence, if properly calibrated. However, neural networks are often regarded as providing overconfident predictions in general, and, worse, providing misidentifications at high confidence. Giving neural networks the ability to make nuanced predictions and account for their own uncertainty in decision making is a potentially powerful improvement, which we discuss in more detail over the next sections.

It is important to be specific and distinguish between epistemic (systematic) and aleatoric (random) uncertainty for the purposes of our classification problem (Kendall & Gal 2017). Aleatoric uncertainty is captured by the classifier's score value, and originates from noise in the input data. More crucial for our application is quantifying the epistemic uncertainty – that is the uncertainty in our choice of neural network's model weights. This epistemic source of error is directly quantifiable through Bayesian neural networks, and in later sections this is the error, confidence, or uncertainty we refer to and attempt to quantify. In the Bayesian framework, this can be achieved by casting model parameters as probability distributions, and using the mechanics of Bayesian statistics to marginalize the neural network output over these distributions, in the process finding the score posterior. In this way, the uncertainty inherent in model selection can be quantified. There are various approximate and exact approaches to achieve this which we outline below.

Dropout (Srivastava et al. 2014) provides a useful form of regularization in neural networks. At each training step, a fraction p (a tunable hyperparameter) of neuron weights are randomly set to zero, decreasing the effective number of parameters of the model. In this way, overfitting can be prevented and generalization accuracy can be increased. In traditional neural networks, dropout is not active at inference time so that all neurons are used for making predictions.

However, Gal & Ghahramani (2015a) demonstrate the profound result that training and evaluating neural networks with dropout is equivalent to performing the approximate Bayesian inference discussed above, with multiple evaluations being equivalent to Monte Carlo integration of the posterior distribution. This is directly applicable to CNNs, via the Monte Carlo dropout technique (Gal & Ghahramani 2015b; referred to as MCDropout for brevity from now on).

Alternative approaches to uncertainty estimation exist (Bayes by Backprop, Blundell et al. 2015), which instead directly performs the approximate Bayesian inference by instead casting neuron weights as distributions with associated hyperparameters, then updating these according to the backpropagated gradients (like deterministic NNs). In this work, we opt to use MCDropout for computational efficiency and for maximal compatibility with existing network architectures and software. No changes to the training loop are required, and only a simple wrapper is required at inference to perform multiple predictions with dropout enabled. The only significant additional computational cost for a Bayesian neural network using the MCDropout technique over a deterministic CNN is at inference time, as multiple samples need to be drawn to approximate the posterior. This performance overhead can be mitigated with suitable batching of the data set. The ability of neural networks to learn complex, non-linear representations in high-dimensional vector spaces is well known and utilized throughout machine learning. However, estimation of the uncertainty of products of neural networks is often a barrier to their implementation in scientific applications, where well-grounded determination of errors is important. MCDropout provides a principled way to introduce this.

Although a comparatively new technique, Bayesian neural networks show emerging promise across a variety of astronomical classification and regression tasks – including supernova light curve classification (Möller & de Boissière 2020), efficient learning of galaxy morphology (Walmsley et al. 2020), and age estimation of stars for galactic archaeology (Ciucă et al. 2021).

There is disagreement in the literature on the precise nature of a Bayesian neural network and how to implement it 'properly', from approximate variational inference as used here, to applying some variant of the Markov Chain Monte Carlo sampler over the weight and bias parameters of the neural network. However, what is relevant for the implementation in this work is that examples the classifier is unconfident about are assigned lower confidence scores than obviously real/bogus detections. More complex tests, such as confirming that the classifier's confidence matches the actual confidence of the data set/some human-derived uncertainty score are beyond the scope of the introductory work presented here.

While these posterior predictions are informative to human vetters, converting them to a single informative summary parameter that captures the overall uncertainty is more useful for integration into pipelines and enabling coarse filtering of candidates. To convert the posterior distributions to meaningful information about the confidence of a given prediction, we utilize the information entropy $\mathbb{H}$. For a binary classification problem, the generic entropy formula can be reduced to

$$\mathbb{H}(p) = -p \log_2 p - (1-p) \log_2 1 - p,$$

where p is the probability of a given detection being real (the real-bogus score). The entropy is maximized for $p = 0.5$, where the probability of being real versus bogus is equal, or the classifier prediction carries no useful information. We define the classifier confidence $\mathbb{C}$ in terms of the average entropy of the posterior distribution samples, scaling to confidences in the range [0, 1] with the relation

$$\mathbb{C} = 1 - \frac{1}{N}\sum_{i=1}^{N} \mathbb{H}_i$$

where N is the number of posterior samples and $\mathbb{H}_i$ is the binary entropy of the ith posterior sample. This metric is equivalent the second term of the BALD acquisition function of Houlsby et al. (2011), and is chosen as it is pre-normalized to [0, 1] unlike standard deviation or similar metrics. Naturally the uncertainties we derive here are correlated with the actual output score, but the multiple samples provide sufficient dispersion that this metric is useful to assess model confidence. In future implementations, these raw posterior samples (or some approximating distribution parameters to reduce data needs) could be fed directly into downstream, more specialized classification tools to enable them to make use of the real-bogus classifier's probabilistic predictions in their own score/posterior.

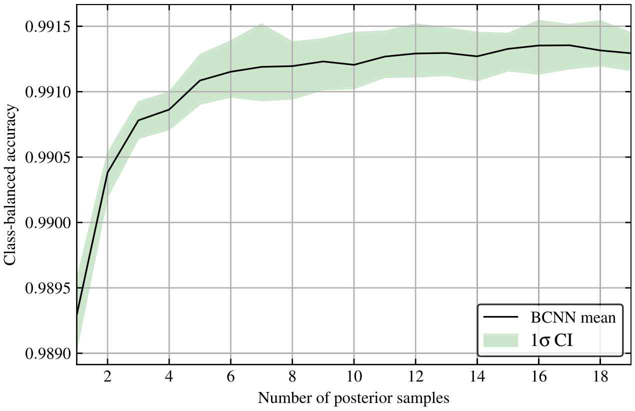

Figure 6. Classification accuracy on the test set from Section 2.3 as a function of the number of posterior samples averaged. Each point is the average of 10 model runs, with the shaded area corresponding to the 1σ confidence interval. The BCNN quickly recovers the performance of a deterministic CNN within statistical uncertainty (99.18 ± 0.03 per cent accuracy, F1: 0.9877) and provides additional information in the form of confidence. No significant improvement in classification accuracy is obtained beyond 10 samples, remaining consistent out to 50 samples.

3.3 Using the uncertainty in classifier predictions

One immediate advantage of Bayesian neural networks over deterministic neural networks is the ability to improve classification performance through model ensembling. Fig. 6 illustrates the gain in accuracy observed by averaging the predictions of our BNN, as a function of the number of posterior samples. Although small, this is a definite improvement over single-evaluation predictions, and is likely constrained by our data set. For the majority of positive and negative examples the model is highly confident about the assigned RB score, so averaging over the posteriors does not improve them significantly. This increase in performance is likely to be greater on more complex (multiclass) classification problems, or scenarios where significantly less training data are available.

Posteriors and/or associated confidence scores can be added to any downstream candidate evaluation tools, providing an additional metric to inform decisions. Objects with both high score and high confidence are highly likely to be genuine, so can be prioritized in human vetting of candidates. This means more time can be spent looking at more marginal candidates, and obvious detections can quickly be identified. Confidence provides a complementary metric

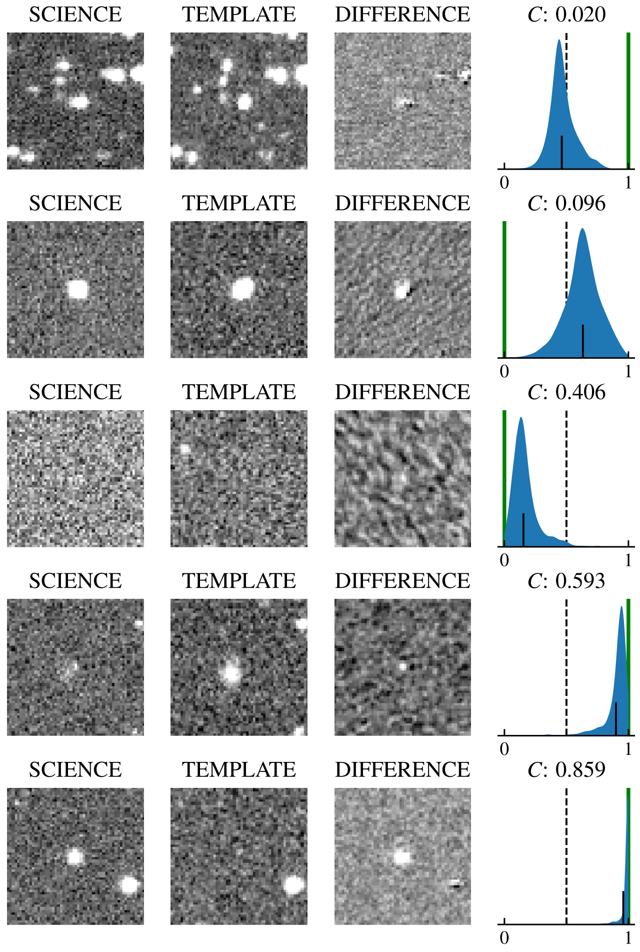

Figure 7. A selection of example posteriors, taken from real GOTO data. The majority of predictions are highly confident, so we select examples of increasing confidence score ($\mathbb{C}$) to display here. Plotted here is a Gaussian kernel-density estimate constructed from 500 posterior samples. The green line indicates the correct label for each candidate, with the black line indicating the mean of the distribution. The dashed line indicates $P_{\text{real}} = 0.5$.

to the pure real-bogus score that can help alleviate some of the issues with the poor dynamic range observed in the classifier outputs at low/high scores. Classification is still performed on the consensus real-bogus score derived from the posterior, with the confidence score intended to aid human decision making. In Fig. 7, we illustrate some example candidates, their associated real-bogus score, and the score posterior.

Classifier confidence is also a useful tool for the training and development process, providing deeper insight into the functioning of the classifier and the associated training set. Predictive uncertainty provides a useful heuristic to clean data sets of mislabelled data. Misclassified detections that the classifier returns a high confidence for are very likely to be mislabelled, as the confidence score is partially based on seeing large numbers of similar detections in the training set. These frames can be actively prioritized in any human relabelling efforts, or fixed cuts on the confidence can be utilized to perform this in a semi-automated way. This 'optimal relabelling' scheme provides a method for human vetters and machine learning models to collaboratively and iteratively refine noisy labels. Our label noise is introduced as humans are imperfect judges of real/bogus, and interpret the vetting rubric in different ways leading to inconsistencies which can harm model performance.

We demonstrate the effectiveness of this procedure on the training set built in this work by training the model first on the uncleaned data set, then attempt to relabel the misclassified detections in the training and test set ordered by decreasing confidence. This amounts to a substantial task of 3580 stamps, which would take a prohibitively long time to relabel by hand, notwithstanding the possibility of human bias in the relabelling. We instead here propose a heuristic relabelling scheme based on the BALD score of Houlsby et al. (2011) that leverages the simplistic nature of binary classification.

The model is first trained on the 'unclean' data set generated with the approaches in Section 2.3, then the BALD score is evaluated over the misidentifications in the test and training sets. From here, a new set of labels is derived by flipping the labels of those examples that have a BALD score less than (thus confidence higher than) the median – effectively accepting the prediction of the classifier over the human vetter. This approach is naturally capable of flipping the labels of accurately labelled stamps incorrectly, but by imposing this cut in classifier confidence it ensures that the majority of relabelled stamps each round correspond to regions of classifier parameter space that are well covered by the training set and so are classified at high confidence. This method effectively trades active human labelling time for passive background computational time, and can be applied iteratively as suggested above to progressively improve the quality of the data set labelling. We manually checked a subset of the sources selected to be relabelled to verify these were sensible and indeed found they were mislabelled detections that had leaked through the quality cuts we applied. After one round of the heuristic relabelling routine outlined above, the class-balanced accuracy achieved on the classifier test set improved markedly from 98.72 ± 0.02 to 99.12 ± 0.01 per cent (F1 score: 0.9826 ± 0.0003 to 0.9877 ± 0.0002), demonstrating the efficacy of this approach. We adopt this cleaned data set for the following sections.

When visualized in an intuitive way, this confidence score can provide insights into the specific families of detection that the classifier is uncertain about. A natural approach to combine this with is to examine the latent space of the neural network. The first convolutional stage of our network can be thought of as a feature extractor, with the resultant feature vector encoding high-level information about the morphological characteristics of our data set, providing insight about potential groupings of detection types through clusterings in this space. To explore the latent space within our model, we apply t-stochastic neighbour embedding (t-SNE, Maaten & Hinton 2008) to the output vector of the layer prior to the fully connected classification layer to reduce the dimensionality and identify clusterings of common data points. The combined process projects an 800-dimensional vector space down to (in our case) a 2D plane. In this space, points with similar latent vectors appear close to each other, thus providing a clustering of the latent space which can be used to visualize the internals of the neural network. This is a purely diagnostic clustering for visualization purposes, as t-SNE does not preserve global distances, nor does it provide a bidirectional mapping from the compressed latent space to the full latent vector space. Fig. 8 illustrates this technique applied to the test set, coloured by both detection subclass (left) and classifier confidence (right).

A useful insight this compressed space provides is the ability to identify clusters of low-confidence points. This immediately reveals types of detection where the classifier may be uncertain, due to intrinsic difficulty of classification (sources close to the detection limit, nuclear transients, unusual PSFs), or scarcity of training data in general. The fact that there are clear divisions between the

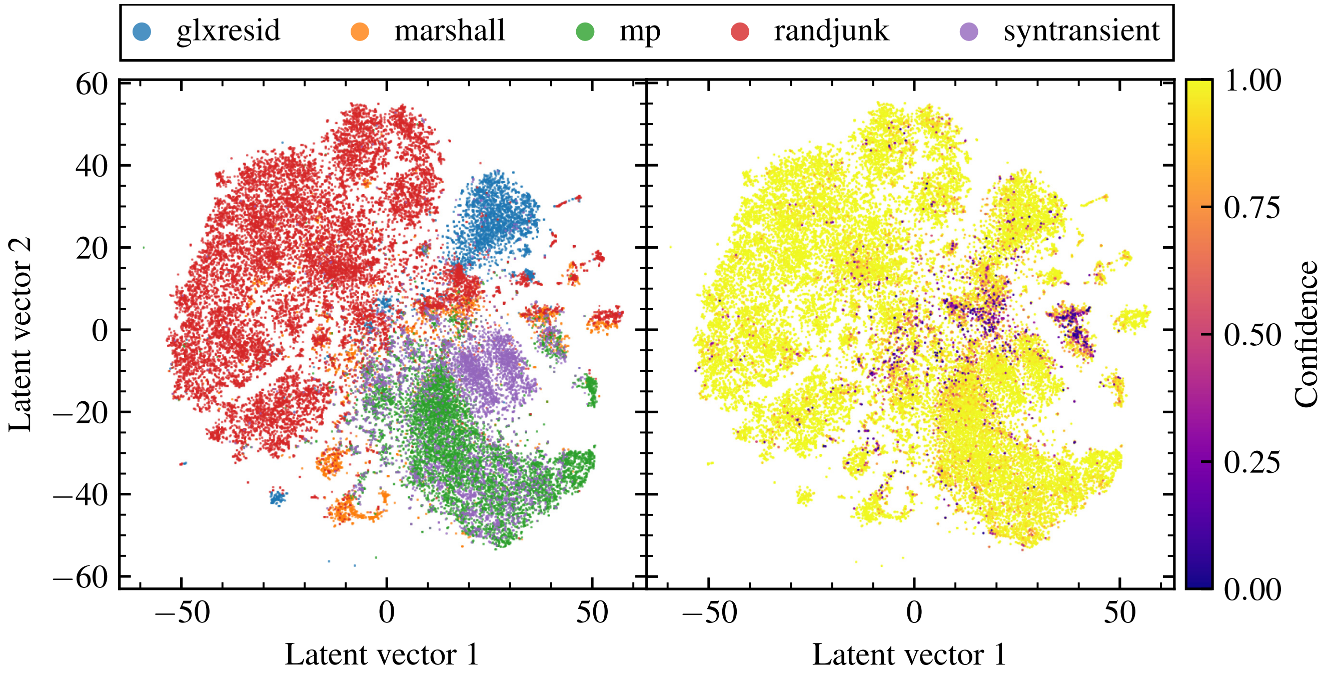

Figure 8. Example class-clustering (left) and confidence (right) maps generated from the classifier's test set. Each colour in the left-hand panel represents a specific subclass of detections, where colour on the right represents classifier confidence. The top legend gives the classes corresponding to each colour in the left-hand panel. Regions of low confidence in the right-hand panel tend to correspond to cluster boundaries in the left, where there is more uncertainty about which class each example belongs to.

coloured subclasses in the left-hand panel of Fig. 8 implies that the classifier has learned something about the intrinsic morphology of the detections beyond simple real-bogus division. Neither the classifier nor the clusterer receive these higher level metalabels, so the clear partitions between the subclasses is purely a result of the internal representations learned.

For more complex data sets where the labelling budget for training examples is limited, Bayesian neural networks enable active learning – a process where the model identifies input data from a large unlabelled pool that would provide the greatest gain in information to it, using the uncertainty. This has been applied to CNNs with great success (Gal, Islam & Ghahramani 2017), and is likely a useful tool for fine-tuning existing training sets in light of new data. We trialled Bayesian active learning as a tool to build the training set presented in this work using the BALD score (Houlsby et al. 2011) as our acquisition function, although it showed no significant improvement over a random selection from the unlabelled pool. This is likely due to the formulation of our classification problem – using only binary labels, and our data being dominated by large numbers of high-confidence real and bogus examples – only rare examples which add little to the overall classification accuracy are acquired. The additional complexity introduced by a multiclass labelling scheme along with the greater entropy provided by having multiple output neurons would likely yield better results.

4 EVALUATION OF CLASSIFIER PERFORMANCE

Machine learning algorithms acquire inherent and often subtle biases based on the training set used in their construction. Given the automated nature of our data set generation, it is particularly important to verify that performance is consistent across a range of parameters of interest, such as transient magnitude. Some care is required in choosing the test set for evaluating classifier performance in a real-world setting, as the training set has been augmented with both human-labelled data and fully synthetic data. Although a low FPR/FNR on the validation and test data is encouraging as it is artificially made more difficult for the classifier to learn, it is not directly representative of the performance we should expect in deployment as a non-negligible component of it is synthetic. Performance characterization should be reinforced with extensive testing on representative samples of GOTO data. A particular focus is to confirm that the synthetic augmentation scheme we implement leads to genuine improvements in the classifier's recovery rate of real transient detections. We also emphasize that in following sections, we effectively test the performance of the real-bogus classifier in isolation – the 'real-world' detection efficiency is the product of the efficiency of multiple pipeline stages, most crucially image subtraction and source extraction. Exploring the impact of these steps is beyond the scope of this paper, and thus are left to future work.

In the following sections, we use 'accuracy' to refer to the class-balanced accuracy, as it is more appropriate for our mildly imbalanced classification task. We also quote results based on the mean scores of 10 posterior samples (motivated by the saturation observed in Fig. 6) since individual evaluations of a Bayesian neural network using MCDropout are based on weaker classifiers due to the presence of dropout. Typical uncertainties (estimated as the standard deviation) on the metrics below are <0.05 per cent, largely arising from the small number of examples around the decision boundary – where uncertainties exceed this they are given explicitly.

4.1 Performance on the test set

To provide a more granular view of the classifier performance, we further split the test set into two groups for the purposes of evaluation.

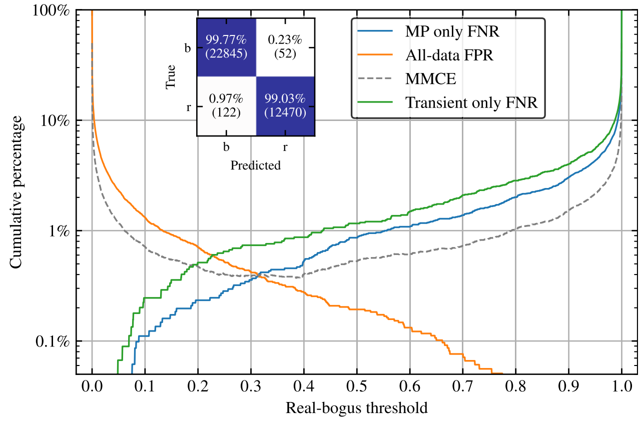

Figure 9. False positive/negative rate evaluated on the test set, excluding Marshall examples. Performance metrics are split based on minor planet and synthetic transients. The grey dashed line (MMCE) represents the full-data set mean misclassification error, which is below 1 per cent between real-bogus scores of 0.1–0.6. Inset: confusion matrix, evaluated on the full test set. There is a slight difference in the false negative rates achieved between the minor planets and synthetic transients, reflecting the increased difficulty posed by complex host morphology and subtraction residuals.

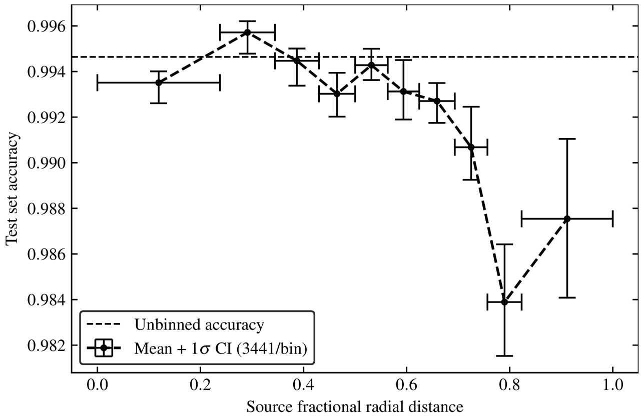

Figure 10. Class-balanced accuracy evaluated on the test set as a function of detector position. We use a series of concentric radial bins, chosen to contain equal numbers of sources for uniform statistics. We scale the radius through by the detector size to give a relative picture of performance. The drop in performance at large radial distances is primarily caused by the extreme optical distortion present in the early GOTO prototype, and only a minor drop of 1 per cent in accuracy in these challenging conditions demonstrates the very robust performance of our classifier. With the design-specification GOTO optics, we anticipate this curve will be level within error.

The first comprises of only the minor planet and random bogus detections. We also test a synthetic transient/galaxy residual test set, to verify that the classifier can genuinely discriminate between galaxies and galaxies with transients. This also reveals any strong performance differences between the two main positive classes, which could skew metrics evaluated on the whole data set. For both test sets, the human-inspected Marshall data are deliberately excluded, since it is significantly more challenging for the classifier than normal detections and does not accurately reflect the true data distribution.

The best-scoring classifier after hyperparameter optimization shows excellent performance, attaining balanced accuracies of 99.49 per cent (F1: 0.9935) and 99.19 per cent (F1: 0.9925) on the minor planet and synthetic transient test data sets, respectively. Fig. 9 illustrates the false positive and negative rates for the classifier on both the minor planet and transient data sets, as a function of the real-bogus threshold chosen. There is a clear difference in false negative rate between the minor planet and transient data sets, reflecting the increased difficulty associated with the complex host morphology associated with the transient examples. The classifier displays a notable skew in the FPR/FNR equality point towards lower values. This is a result of the Marshall injections in the training set, which are made more difficult to learn than the random bogus detections due to being misclassified by the previous classifier. This does not affect classification accuracy, and could be fixed by applying a power transform to the classifier output if required, conditioned on the validation set.

Given the spatially variable optical characteristics present in the GOTO prototype, it is important to confirm that our classifier provides good performance across the full detector – and not simply in the centre where distortion is minimal. In Fig. 10, we plot the class-balanced accuracy score as a function of radial position on the detector, using a series of radial bins chosen to equalize source density. These radial bins are scaled through by the maximum value (corresponding to the image corner) to provide a scale-free measurement of detector position. Class-balanced accuracy is used here as the real-bogus fraction varies as a function of detector position, and care must be taken to account for this. We find a consistent performance of ∼99 per cent out to a fractional radial distance of 0.7, with a slight drop of 1 per cent out at the far edge of the image. This is primarily due to the severe distortion found in the image corners of the GOTO prototype optical tubes, which produces very challenging detections (abnormal PSFs, strong vignetting) both for source extraction and real-bogus classification. Some contribution to this performance decrease is likely from good quality sources close to the edge of the image or close to the edge of the science-template overlap. Estimating reliably these sources and their contribution to the numbers in each bin is a complex task. Suffering only a 1 per cent decrease in performance in these extremely challenging conditions demonstrates the overall robustness of the classifier. With the significantly improved optical quality of the GOTO design specification OTAs, we anticipate that future versions of our classifier trained on data from the upgraded system will display a constant (within statistical error) classification accuracy as a function of detector position.

4.2 Performance on spectroscopically confirmed transients

To provide the most accurate assessment of transient-specific classifier performance and further confirm that our algorithmically generated training set generalizes well, we assemble a test set of genuine astrophysical transients. This set was found by cross-matching a list of all spectroscopically confirmed supernovae reported to the Transient Name Server (TNS) since 2019 January with the GOTO master candidate table. Those with an associated GOTO candidate within 3 arcsec, with TNS discovery magnitude greater than the GOTO source magnitude, and only found in GOTO data after the formal TNS discovery date are accepted. With these cuts, purity is favoured over completeness, a deliberate choice to ensure that the test set is as clean of false positives as possible. This yields 877 known transients recovered in the GOTO prototype data. The whole-sample recovery rate is 97.2 ± 0.3 per cent, consistent with the performance achieved on the synthetic transients. This is a strong indicator that our generation algorithm for synthetic transients produces convincing detections which are useful for learning to detect genuine transients.

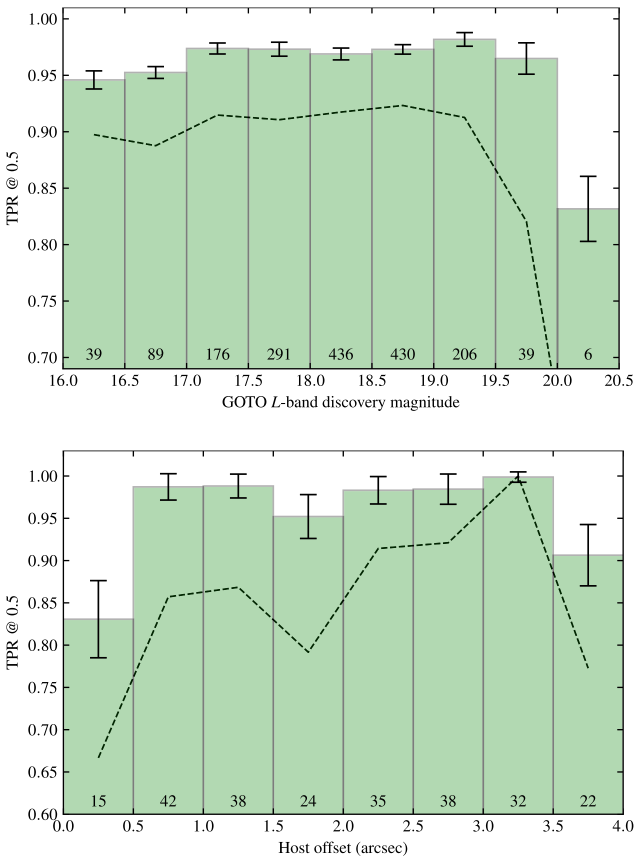

Figure 11. Top panel: Recovery rate (TPR) as a function of GOTO discovery magnitude, at a fixed real-bogus threshold of 0.5. The dashed line indicates the performance of a classifier with a similarly sized training set, but with only minor planet detections. Error bars are derived directly from the classifier score posteriors. The number of detections per bin is written below each bar. The sharp drop-off in the number of detections beyond $L \sim 19.5$ is associated with the median 5σ limiting magnitude of the GOTO prototype, thus expected. Bottom panel: Recovery rate of transients that can be reliably associated with a host galaxy (as cross-matched with WISExSuperCosmos, Bilicki et al. 2016) as a function of host offset. As above, error bars are derived from the classifier score posteriors, and a similarly sized minor planet-based classifier is plotted for comparison. There is a marked improvement in the recovery rate for very small host offsets, particularly for nuclear transients.

Uncertainties on the TNS-derived set are larger than for our synthetic data sets due to both the smaller sample size and the increased complexity of the real data set.

To confirm that consistent performance across a wide range of magnitudes is attained, the recovery rate is evaluated across a series of magnitude bins. Fig. 11 illustrates the transient recovery rate as a function of GOTO L-band magnitude. We find that the classifier maintains excellent performance across the full magnitude range of detections accessible to GOTO, even towards fainter magnitudes. Our galaxy augmentation scheme provides up to a 30 per cent improvement in recovery rate at magnitudes fainter than $L \sim 19.5$ over a pure minor planet training set. This marked improvement at the faint end of our detection range is powerful, as the expected number of other transients increases as a function magnitude, meaning this improvement in recovery rate will yield a corresponding increase in the total number of transients recovered by GOTO. Of particular relevance for GOTO, we expect the majority of kilonovae within the current GW detection volume to also occupy this magnitude range, increasing significantly our recovery rate of these faint transients in particular.

Our augmentation scheme also provides a significant improvement for sensitivity to nuclear transients, considered to be a more difficult transient morphology to detect. Motivated by the typical RMS astrometric noise level of GOTO images, we adopt a fixed threshold of 0.5 arcsec to distinguish between nuclear and offset transients. We find a 13 ± 5 per cent increase in the recovery rate of nuclear transients using the transient-optimized classifier compared to a pure minor planet classifier, on a sample of 15 confirmed detections. This is a direct result of the host offset distribution chosen for the augmentation scheme, which permits full freedom to generate close-in nuclear configurations. The main obstacle to improving this further is the inherent quality of the galaxy subtraction residuals, which limits our bright-end performance.

4.3 Further characterization

Although the main transient sources of interest for GOTO will overwhelmingly be fainter than the saturation level ($L \sim 15$), there are still important secondary science Galactic targets as well as rare transients occurring in nearby Local Group galaxies (e.g. SN2014J; Fossey et al. 2014) which have the potential to brighten beyond the well-sampled regions of our training set. To simulate these bright transients, GOTO detections of the first 100 minor planets are used. These have magnitudes $L \sim 10$–14, and have well-constrained orbits. Using the SKYFIELD code (Rhodes 2019), we generate nightly ephemerides for each minor planet, and locate all difference image detections associated with each object. This yields a benchmark set of around 200 bright asteroid detections. Of the 207 detections, 99.5 per cent are recovered, showing good consistency with the recovery rate on the fainter minor planets in the classifier test set. Of those minor planets with $L \lesssim 10$ 100 per cent are recovered, although small-number statistics limits the usefulness of this metric. This bright-end testing demonstrates the excellent dynamic range of the classifier, showing high (>90 per cent) recovery rates from 10th to 20th magnitude.

Through the host offset distribution choice we make, we expect to generate a reasonable number of transients at zero offset, so this region of parameter space should not be empty in the training set. To test the performance in this regime we repeated the procedure outlined in Section 2.2, except with the host offsetting routine disabled to generate synthetic detections overlapping the galaxy nucleus only. This generated 5100 synthetic nuclear transients, with a magnitude distribution consistent with that in Fig. 1. Testing our model against this data set (with the negative examples being galaxy residuals as in Section 2.3, we obtain a 97.5 per cent accuracy, with a recovery rate (TPR) of ~96 per cent. These scores are lower than the full-data set scores, reflecting the increased difficulty of classification in this regime. The average prediction confidence on the real component of this set is 0.9390, which is less than the average prediction confidence on the real members of the test set is 0.9626, reinforcing that these detections are more difficult than the ‘average’ real detection.

Another important factor to consider with any classifier is how closely the output correlates with probability – known as calibration. Although this does not necessarily impact on the classification performance, having scores that accurately reflect the probabilities of being real/bogus is important for human use of classification outputs and is important for performing inference using classifier scores. In Fig. 12, we illustrate the calibration of this classifier by plotting as a function of classifier score the fraction of real detections at a

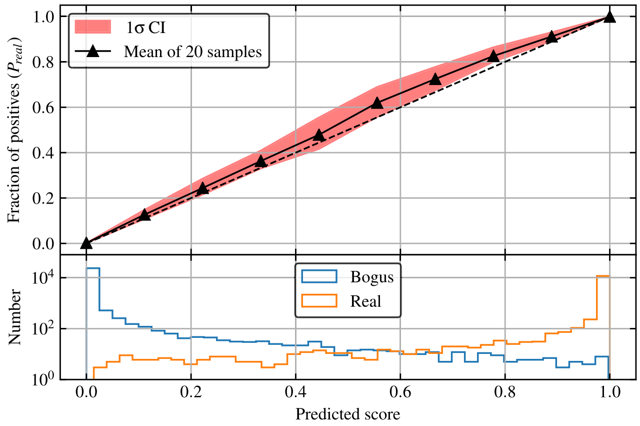

Figure 12. Top panel: classifier calibration curve, illustrating how well the classifier's output score corresponds to probability. The mean of 20 samples and the 1σ confidence interval are plotted to show that individual draws from the posterior remain well-calibrated. Bottom panel: Score distribution for both real and bogus examples – with the relative scarcity of examples with $0.2 < RB < 0.8$ accounting for the greater uncertainty in calibration.

given score. Our uncalibrated classifier shows excellent calibration, and does not show the characteristic sigmoidal calibration curve of other uncalibrated classifiers such as random forests (Niculescu-Mizil & Caruana 2005). Calibration becomes increasingly important if different machine learning models are chained together, with downstream classifiers using the posterior probabilities of the main real-bogus classifier. With our high degree of calibration, we are justified to use our *RB* score as a proxy for P_{real} (the probability a given source is real) in such implementations.

One significant benefit of using a Bayesian neural network is a built-in indicator of out-of-distribution data – that is data poorly represented by or unseen in the training set. For input data that are completely different to the training set, the classifier will return a low confidence score which can then be used to remove/deprioritize the candidate in downstream applications. This confidence can also be used to optimize candidate vetting efforts, with the highest confidence candidates being a natural choice to prioritize over lower confidence, lower quality detections.

In principle, the task-specific knowledge encoded in our trained network weights can be used to accelerate the training of similar real-bogus classifiers through transfer learning, and in principle increase generalization (Yosinski et al. 2014). This requires that the same data input structure is used and there are no changes to model hyperparameters. However, we caution that training in this way is susceptible to local minima and does not offer the opportunity to change the model hyperparameters that training from scratch does – in Section 3.1, we have demonstrated the sizeable performance improvements doing a full hyperparameter search can yield, and so encourage this.

The techniques and framework we implement in this paper are naturally extensible to more challenging astronomical classification tasks such as those outlined at the end of Section 1.1. A key focus is more fine-grained classification – being able to distinguish variable stars, supernovae, nuclear transients, and other astrophysical objects of interest in an automated (and crucially, accurate) way. Fig. 8 already hints at this being a fruitful approach, as we see evidence of morphological differentiation in both the positive and negative classes through the emergence of smaller subclusters. Similarly, leveraging the wealth of contextual information available from astrophysical surveys in a principled, informative, and efficient way within the framework of deep learning poses an open challenge, with potentially significant gains possible. We aim to address these challenges, among others, with development of future generations of the classifier we implement here.

5 CONCLUSIONS

We demonstrate a data-driven approach to generating large, low-contamination training sets, which along with our novel augmentation scheme can be used to train high-performance, transient-optimized real-bogus classifiers. By combining real PSFs from minor planets with galaxies, we generate realistic synthetic transients that provide a measurable improvement in the recovery of genuine astrophysical transients. This technique is computationally lightweight, easily implemented, and directly applicable to a variety of both current and future transient survey streams/data sets.

We also demonstrate the efficacy of Bayesian neural networks for the first time in real-bogus classification, and demonstrate the unique insights that confidence estimation can bring to the real-bogus problem. Being able to assign epistemic confidences to classifier predictions in addition to the more typical real-bogus score provides another parameter for human vetters further downstream to use in identifying promising candidate detections – this can potentially be used in future to further automate decision making in the context of follow-up and reporting. Techniques such as this that minimize human involvement in data-gathering and labelling will become increasingly important in the new 'big-data' era of astronomy that large-scale projects such as the Rubin Observatory and SKA will bring about.

Our classifier demonstrates excellent performance across a wide magnitude range, with a missed detection rate of 0.5 per cent at a fixed 1 per cent false positive rate, and up to 30 per cent improvement in recovery rate of astrophysical transients in the challenging faint end. This has the potential to markedly increase the number of faint transients GOTO can discover, and significantly improves the prospects for detecting the kilonova afterglows of gravitational-wave-driven mergers GOTO was designed to find. We anticipate that improvements to the quality and stability of GOTO's hardware and dataflow will bring significant performance gains for the real-bogus classifier presented here.

GOTO is due to undergo significant expansion over the coming years, with a final configuration of four installations spread across a northern (La Palma) and southern (Siding Spring) site providing a high-cadence datastream covering almost the whole sky down to 20th magnitude every 2–3 d. The tools developed in this work have generated a classifier that is capable of handling and sifting the accompanying volume of candidate transient detections with robust accuracy and high sensitivity.

ACKNOWLEDGEMENTS

We thank the anonymous referee for their insightful comments which helped improve the quality of this manuscript. The Gravitational-wave Optical Transient Observer (GOTO) project acknowledges the support of the Monash-Warwick Alliance; Warwick University; Monash University; Sheffield University; the University of Leicester; Armagh Observatory & Planetarium; the National Astronomical Research Institute of Thailand (NARIT); the University of Turku; the University of Manchester; the University of Portsmouth; the Instituto de Astrofísica de Canarias (IAC) and the Science and Technology Facilities Council (STFC). DS, KU, BG, and JDL acknowledge support from the STFC via grants ST/T007184/1, ST/T003103/1, and ST/P000495/1. JDL acknowledges support from a UK Research and Innovation Fellowship (MR/T020784/1). RPB, MRK, and DMS acknowledge support from the ERC under the European Union's Horizon 2020 research and innovation programme (grant agreement No. 715051; Spiders). POB and RS acknowledge support from the STFC.

This research made use of ASTROPY,[7] a community-developed core Python package for Astronomy (Astropy Collaboration 2013; Price-Whelan et al. 2018), and SCIKIT-LEARN (Pedregosa et al. 2011). The resources to support `astorb.dat` were originally provided by NASA grant NAG5-4741 (PI E. Bowell) and the Lowell Observatory endowment, and more recently by NASA PDART grant NNX16AG52G (PI N. Moskovitz). This research has made use of IMCCE's SkyBoT VO tool. This research has made use of data and/or services provided by the International Astronomical Union's Minor Planet Center.

DATA AVAILABILITY

The GOTORB code is made freely available at https://github.com/GOTO-OBS/gotorb, along with validation examples for testing. Accompanying observational data used in this work will be made available via upcoming GOTO public data releases.

REFERENCES

Aartsen M. G. et al., 2017, J. Instrum., 12, P03012
Abadi M. et al., 2015, TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. Available at: https://www.tensorflow.org/
Abbott B. P. et al., 2017a, Phys. Rev. Lett., 119, 161101
Abbott B. P. et al., 2017b, ApJ, 848, L12
Ackley K., Eikenberry S. S., Yildirim C., Klimenko S., Garner A., 2019, AJ, 158, 172
Alard C., Lupton R. H., 1998, ApJ, 503, 325
Astropy Collaboration, 2013, A&A, 558, A33
Bailey S., Aragon C., Romano R., Thomas R. C., Weaver B. A., Wong D., 2007, ApJ, 665, 1246
Becker A., 2015, Astrophysics Source Code Library, record ascl:1504.004
Bellm E. C. et al., 2019, PASP, 131, 018002
Berthier J., Vachier F., Thuillot W., Fernique P., Ochsenbein F., Genova F., Lainey V., Arlot J. E., 2006, in Gabriel C., Arviset C., Ponz D., Solano E., eds, ASP Conf. Ser. Vol. 351, SkyBoT, a new VO service to identify Solar System objects. Astron. Soc. Pac., San Francisco, p. 367
Berthier J., Carry B., Vachier F., Eggl S., Santerne A., 2016, MNRAS, 458, 3394
Bertin E., Arnouts S., 1996, A&AS, 117, 393
Bilicki M. et al., 2016, ApJS, 225, 5
Bloom J. S. et al., 2012, PASP, 124, 1175
Blundell C., Cornbise J., Kavukcuoglu K., Wierstra D., 2015 , in Bach F, Blei D, eds, Proceedings of the 32nd International Conference on Machine Learning, *Weight Uncertainty in Neural Networks*. PMLR, Lille, France, p. 1613
Breiman L., 2001, Mach. Learn., 45, 5
Brink H., Richards J. W., Poznanski D., Bloom J. S., Rice J., Negahban S., Wainwright M., 2013, MNRAS, 435, 1047
Cabrera-Vives G., Reyes I., Förster F., Estévez P. A., Maureira J.-C., 2017, ApJ, 836, 97
Carrasco-Davis R. et al., 2020, preprint (arXiv:2008.03309)
Chambers K. C. et al., 2016, preprint (arXiv:1612.05560)
Chetlur S., Woolley C., Vandermersch P., Cohen J., Tran J., Catanzaro B., Shelhamer E., 2014, preprint (arXiv:1410.0759)
Chollet F. et al., 2015, Keras. Available at: https://keras.io
Ciucă I., Kawata D., Miglio A., Davies G. R., Grand R. J. J., 2021, MNRAS. Available at: https://doi.org/10.1093/mnras/stab639
Dálya G. et al., 2018, MNRAS, 479, 2374
Dieleman S., Willett K. W., Dambre J., 2015, MNRAS, 450, 1441
Duev D. A. et al., 2019, MNRAS, 489, 3582
Fawcett T., 2006, Pattern Recognit. Lett., 27, 861
Filippenko A. V., Li W. D., Treffers R. R., Modjaz M., 2001, in Paczynski B., Chen W.-P., Lemme C., eds, ASP Conf. Ser. Vol. 246, IAU Colloq. 183: Small Telescope Astronomy on Global Scales. Astron. Soc. Pac., San Francisco. p. 121
Fossey S. J., Cooke B., Pollack G., Wilde M., Wright T., 2014, Cent. Bur. Electron. Telegrams, 3792, 1
Gal Y., Ghahramani Z., 2015a, in Balcan M. F., Weinberger K. Q., eds, Proceedings of The 33rd International Conference on Machine Learning, *Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning*. PMLR, New York, NY, USA, p. 1055
Gal Y., Ghahramani Z., 2015b, preprint (arXiv:1506.02158)
Gal Y., Islam R., Ghahramani Z., 2017, in Precup D., Teh Y. W., eds, Proceedings of the 34th International Conference on Machine Learning, *Deep Bayesian Active Learning with Image Data*. PMLR, International Convention Centre, Sydney, Australia, p. 1183
Gieseke F. et al., 2017, MNRAS, 472, 3101
Goldstein D. A. et al., 2015, AJ, 150, 82
Gompertz B. P. et al., 2020, MNRAS, 497, 726
Goodfellow I. J., Pouget-Abadie J., Mirza M., Xu B., Warde-Farley D., Ozair S., Courville A., Bengio Y., 2014, in Ghahramani Z., Welling M., Cortes C., Lawrence N., Weinberger K. Q,eds, Advances in Neural Information Processing Systems 27, *Generative Adversarial Nets* . Curran Associates Inc., Red Hook, NY, USA., 2672
Heinze A. N. et al., 2018, AJ, 156, 241
Houlsby N., Huszár F., Ghahramani Z., Lengyel M., 2011, preprint (arXiv:1112.5745)
IceCube Collaboration, 2018, Science, 361, eaat1378
Ivezić, Ž. et al., 2019, ApJ, 873, 111
Kendall A., Gal Y., 2017, in Guyon I.,Luxburg U. V., Bengio S., Wallach H., Ferguson R., eds, *Advances in Neural Information Processing Systems 30, What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision?*. Curran Associates Inc., Red Hook NY, USA
Kingma D. P., Ba J., 2014, preprint (arXiv:1412.6980)
Kulkarni S. R., 2012, preprint (arXiv:1202.2381)
Law N. M. et al., 2009, PASP, 121, 1395
Leaman J., Li W., Chornock R., Filippenko A. V., 2011, MNRAS, 412, 1419
LeCun Y., Bengio Y., 1995 , The Handbook of Brain Theory and Neural Networks. MIT Press, Cambridge, MA, USA, p. 255
LeCun Y., Bengio Y., Hinton G., 2015, Nature, 521, 436
LeNail A., 2019, J. Open Source Soft., 4, 747
Li W. et al., 2011, MNRAS, 412, 1441
Li L., Jamieson K., DeSalvo G., Rostamizadeh A., Talwalkar A., 2017, J. Mach. Learn. Res., 18, 6765
Lintott C. J. et al., 2008, MNRAS, 389, 1179
Maaten L. v. d., Hinton G., 2008, J. Mach. Learn. Res, 9, 2579
Mahabal A. et al., 2019, PASP, 131, 038002
Mariani G., Scheidegger F., Istrate R., Bekas C., Malossi C., 2018, preprint (arXiv:1803.09655)
Meegan C. et al., 2009, ApJ, 702, 791
Möller A., de Boissière T., 2020, MNRAS, 491, 4277
Mong Y.-L. et al., 2020, MNRAS, 499, 6009
Moskovitz N., Schottland R., Burt B., Wasserman L., Mommert M., Bailen M., Grimm S., 2019, EPSC-DPS Joint Meeting 2019, *Modernizing Lowell Observatory's astorb Database*
Niculescu-Mizil A., Caruana R., 2005, Proceedings of the 22nd International Conference on Machine Learning, *Predicting Good Probabilities with Supervised Learning*. Association for Computing Machinery, New York, NY, USA, p. 625
O'Malley T. et al., 2019, Keras Tuner. Available at: https://github.com/keras-team/keras-tuner
Pedregosa F. et al., 2011, J. Mach. Learn. Res., 12, 2825
Perlmutter S. et al., 1999, ApJ, 517, 565

[7] http://www.astropy.org

Pian E. et al., 2017, Nature, 551, 67

Price-Whelan A. M. et al., 2018, AJ, 156, 123

Reyes E., Estévez P. A., Reyes I., Cabrera-Vives G., Huijse P., Carrasco R., Forster F., 2018, 2018 International Joint Conference on Neural Networks (IJCNN), *Enhanced Rotational Invariant Convolutional Neural Network for Supernovae Detection*. IEEE, New York, NY, USA,p. 1

Rhodes B., 2019, Astrophysics Source Code Library, record ascl:1907.024

Romano R. A., Aragon C. R., Ding C., 2006, 2006 5th International Conference on Machine Learning and Applications (ICMLA'06), *Supernova Recognition Using Support Vector Machines*. IEEE, Orlando, FL, USA, p. 77

Shappee B. J. et al., 2014, ApJ, 788, 48

Simonyan K., Zisserman A., 2014, preprint (arXiv:1409.1556)

Singer L. P. et al., 2015, ApJ, 806, 52

Smith K. W. et al., 2020, PASP, 132, 085002

Soumagnac M. T., Ofek E. O., 2018, PASP, 130, 075002

Srivastava N., Hinton G., Krizhevsky A., Sutskever I., Salakhutdinov R., 2014, J. Mach. Learn. Res., 15, 1929

Tanvir N. R. et al., 2009, Nature, 461, 1254

Tompson J., Goroshin R., Jain A., LeCun Y., Bregler C., 2015, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, *Efficient Object Localization using Convolutional Networks*. IEEE, New York, NY, USA, p. 648

Tonry J. L. et al., 2018, PASP, 130, 064505

Turpin D. et al., 2020, MNRAS

Villar V. A., Berger E., Metzger B. D., Guillochon J., 2017, ApJ, 849, 70

Walmsley M. et al., 2020, MNRAS, 491, 1554

Wozniak P. R., 2000, Acta Astron., 50, 421

Wright D. E. et al., 2015, MNRAS, 449, 451

Yip K. H. et al., 2019, AAS/Division for Extreme Solar Systems Abstracts. p. 305.04

Yosinski J., Clune J., Bengio Y., Lipson H., 2014, in Ghahramani Z., Welling M., Cortes C., Lawrence N., Weinburger K. Q., eds., Advances in Neural Information Processing Systems 27, *How Transferable are Features in Deep Neural Networks?*, MIT Press, Cambridge, MA, USA

Zackay B., Ofek E. O., Gal-Yam A., 2016, ApJ, 830, 27

This paper has been typeset from a TEX/LATEX file prepared by the author.