



















# Alert Classification for the ALerCE Broker System: The Real-time Stamp Classifier

R. Carrasco-Davis<sup>1,2,16</sup> , E. Reyes<sup>1,2,16</sup> , C. Valenzuela<sup>1,3,4,5</sup> , F. Förster<sup>1,3,6</sup> , P. A. Estévez<sup>1,2</sup> , G. Pignata<sup>1,7</sup> , F. E. Bauer<sup>1,8,9</sup> , I. Reyes<sup>1,2,3</sup> , P. Sánchez-Sáez<sup>1,4,8,10</sup> , G. Cabrera-Vives<sup>1,11</sup> , S. Eyheramendy<sup>1,4</sup> , M. Catelan<sup>1,8</sup> , J. Arredondo<sup>1</sup>, E. Castillo-Navarrete<sup>1,3</sup>, D. Rodríguez-Mancini<sup>1,11</sup>, D. Ruz-Mieres<sup>1,3,12</sup> , A. Moya<sup>1,3</sup> , L. Sabatini-Gacitúa<sup>1,3</sup>, C. Sepúlveda-Cobo<sup>1,3</sup>, A. A. Mahabal<sup>1,3,14</sup> , J. Silva-Farfán<sup>6</sup>, E. Camacho-Iñiguez<sup>1,8</sup>, and L. Galbany<sup>15</sup> 

<sup>1</sup> Millennium Institute of Astrophysics (MAS), Nuncio Monseñor Sótero Sanz 100, Providencia, Santiago, Chile; [rodrigo.carrasco.davis@gmail.com](mailto:rodrigo.carrasco.davis@gmail.com), [rodrigo.carrasco.d@ing.uchile.cl](mailto:rodrigo.carrasco.d@ing.uchile.cl)

<sup>2</sup> Department of Electrical Engineering, Universidad de Chile, Av. Tupper 2007, Santiago 8320000, Chile

<sup>3</sup> Center for Mathematical Modeling, Universidad de Chile, Beauchef 851, North building, 7th floor, Santiago 8320000, Chile

<sup>4</sup> Faculty of Engineering and Sciences, Universidad Adolfo Ibañez, Diagonal Las Torres 2700, Peñalolén, Santiago, Chile

<sup>5</sup> Data Observatory, Santiago, Chile

<sup>6</sup> Departamento de Astronomía, Universidad de Chile, Casilla 36D, Santiago, Chile

<sup>7</sup> Departamento de Ciencias Físicas, Universidad Andres Bello, Av. Republica 230, Santiago 8370146, Chile

<sup>8</sup> Instituto de Astrofísica and Centro de Astroingeniería, Facultad de Física, Pontificia Universidad Católica de Chile, Casilla 306, Santiago 22, Chile

<sup>9</sup> Space Science Institute, 4750 Walnut Street, Suite 205, Boulder, Colorado 80301, USA

<sup>10</sup> Inria Chile Research Center, Av. Apoquindo 2827, Las Condes, Chile

<sup>11</sup> Department of Computer Science, Universidad de Concepción, Edmundo Larenas 219, Concepción, Chile

<sup>12</sup> Institute of Astronomy, University of Cambridge, Madingley Road, Cambridge CB3 0HA, UK

<sup>13</sup> Cahill Center for Astrophysics, California Institute of Technology, 1200 E. California Boulevard, Pasadena, CA 91125, USA

<sup>14</sup> Center for Data Driven Discovery, California Institute of Technology, Pasadena, CA 91125, USA

<sup>15</sup> Institute of Space Sciences (ICE, CSIC), Campus UAB, Carrer de Can Magrans, s/n, E-08193 Barcelona, Spain

Received 2020 June 30; revised 2021 June 23; accepted 2021 June 25; published 2021 November 5

## Abstract

We present a real-time stamp classifier of astronomical events for the Automatic Learning for the Rapid Classification of Events broker, ALerCE. The classifier is based on a convolutional neural network, trained on alerts ingested from the Zwicky Transient Facility (ZTF). Using only the *science*, *reference*, and *difference* images of the first detection as inputs, along with the metadata of the alert as features, the classifier is able to correctly classify alerts from active galactic nuclei, supernovae (SNe), variable stars, asteroids, and bogus classes, with high accuracy ( $\sim 94\%$ ) in a balanced test set. In order to find and analyze SN candidates selected by our classifier from the ZTF alert stream, we designed and deployed a visualization tool called SN Hunter, where relevant information about each possible SN is displayed for the experts to choose among candidates to report to the Transient Name Server database. From 2019 June 26 to 2021 February 28, we have reported 6846 SN candidates to date (11.8 candidates per day on average), of which 971 have been confirmed spectroscopically. Our ability to report objects using only a single detection means that 70% of the reported SNe occurred within one day after the first detection. ALerCE has only reported candidates not otherwise detected or selected by other groups, therefore adding new early transients to the bulk of objects available for early follow-up. Our work represents an important milestone toward rapid alert classifications with the next generation of large etendue telescopes, such as the Vera C. Rubin Observatory.

*Unified Astronomy Thesaurus concepts:* [Astroinformatics \(78\)](#); [Astrostatistics \(1882\)](#); [Convolutional neural networks \(1938\)](#); [Active galactic nuclei \(16\)](#); [Supernovae \(1668\)](#); [Variable stars \(1761\)](#); [Small solar system bodies \(1469\)](#); [Classification \(1907\)](#); [Surveys \(1671\)](#); [Transient detection \(1957\)](#); [Time domain astronomy \(2109\)](#)

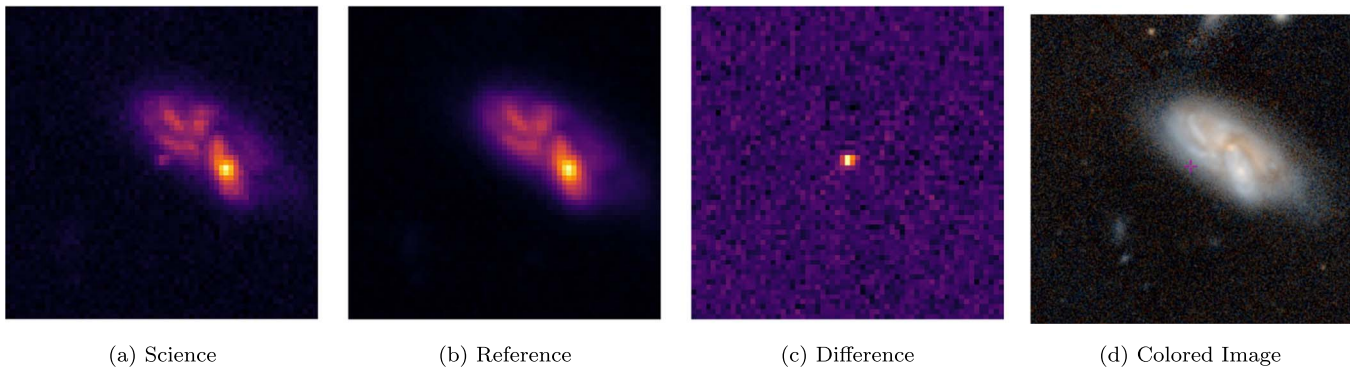
## 1. Introduction

The amount of data generated by modern survey telescopes cannot be directly handled by humans. Therefore, automatic data analysis methods are necessary to fully exploit their scientific return. A particularly challenging problem is the real-time classification of transient events. Nevertheless, the possibility to generate a quick probabilistic evaluation of which type of transient has been discovered is crucial to perform the most suitable follow-up observation, and by extension obtain the best constraints on its physics. In this work, we focus on the early detection of supernovae (SNe) by quickly discerning between SNe and various other classes of astronomical objects. Photometric and spectroscopic observations carried out soon after the explosion are fundamental to put constraints on the progenitor systems and explosion physics.

In the case of thermonuclear explosions (SNe Ia), early observations probe the outermost part of the ejecta, where it is possible to detect the material present at the surface of the progenitor (e.g., Nugent et al. 2011), evaluate the degree of mixing induced by different explosion models (e.g., Piro & Morozova 2016; Jiang et al. 2017; Noebauer et al. 2017), and estimate the size of the companion star (e.g., Kasen 2010).

For core-collapse (CC) SNe, observations carried out soon after the explosion can constrain the radius of the progenitor star, its outer structure, and the degree of  $^{56}\text{Ni}$  mixing (e.g., Tominaga et al. 2011; Piro & Nakar 2013), but also the immediate SN environment, providing a critical diagnostic for the elusive final evolutionary history of the progenitor and/or the progenitor system configuration (e.g., Moriya et al. 2011; Gal-Yam et al. 2014; Groh 2014; Khazov et al. 2016; Tanaka et al. 2016; Morozova et al. 2017; Yaron et al. 2017; Förster et al. 2018).

<sup>16</sup> These authors contributed equally to this work.



**Figure 1.** Example  $g$ -band images from a ZTF alert packet, in this case from a Type Ia supernova (ZTF19abmlyr) classified by our method. (a) The science image is the latest measurement of a source. (b) The reference image is usually a higher signal-to-noise measurement taken from an earlier epoch. (c) The third stamp is the difference between the reference and science images. (d) For context, we also show the  $gri$  color image from PanSTARRS, which is not part of the alert packet nor used in the current stamp classifier. Each image stamp is  $63 \times 63$  pixels, where 1 pixel =  $1''$ .

We propose a method to quickly classify alerts among five different classes, four of which are astrophysical, and then use the predictions to find and report SNe. This work has been developed in the framework of ALerCE<sup>17</sup> (Automatic Learning for the Rapid Classification of Events; Förster et al. 2021). The ALerCE system is able to read, annotate, classify, and redistribute the data from large survey telescopes. Such efforts are commonly called Astronomical Broker Systems (other examples include, e.g., ANTARES, Narayan et al. 2018; Lasair, Smith et al. 2019). Currently, ALerCE is processing the alert stream generated by the Zwicky Transient Facility (ZTF; Bellm et al. 2018), and its main goal is to reliably classify data of non-moving objects and make these classifications available to the scientific community.

For the purpose of classifying astronomical objects or transients, one way to discriminate among them is by computing features from the light curve of each object (e.g., Richards et al. 2011; Pichara et al. 2016; Martínez-Palomera et al. 2018; Boone 2019; Sánchez-Sáez et al. 2021), or by using the light curve directly as input to a classifier (e.g., Mahabal et al. 2017; Naul et al. 2018; Muthukrishna et al. 2019; Becker et al. 2020). In the case of an alert stream scenario such as for ZTF (whereby no forced photometry of past images is provided as of 2021 February), the light curve is built by estimating the flux from the difference image for all alerts triggered at the same coordinates.

Our model is dubbed the *stamp classifier*, since it only uses the first alert of an astronomical object. ALerCE also developed a *light-curve classifier* (Sánchez-Sáez et al. 2021) based on light curves with  $\geq 6$  detections in  $g$  or  $\geq 6$  detections in  $r$  ZTF bands. The *light-curve classifier* is able to discriminate among a richer taxonomy of astronomical objects. Both the stamp and light-curve classifiers are currently running through the ALerCE frontend (Förster et al. 2021).

Our proposed stamp classifier is based on a convolutional neural network (CNN) architecture that uses only the information available in the first alert of an astronomical object, which includes the images of the objects plus metadata regarding some of the object properties, observing conditions, and information from other catalogs. The images included in the alert correspond to the *science*, *reference*, and *difference* images, which are shown in Figure 1 and described in Section 2. The stamp classifier uses the first alert to

discriminate between active galactic nuclei (AGNs), supernovae (SNe), variable stars (VS), asteroids, and bogus alerts. The architecture was designed to exploit the rotational invariance of astronomical images. The classifier was trained using an entropy regularizer that avoids the assignment of high probability to a single class, yielding softer output probabilities that give extra information to experts, useful for further analysis of candidates. To the best of our knowledge, this is the first classifier that discriminates among five classes using a single alert, allowing a rapid, reliable characterization of the data stream to trigger immediate follow-up. Previous work on stamp classification has focused instead on the classification of real objects versus bogus detections (e.g., Goldstein et al. 2015; Cabrera-Vives et al. 2017; Reyes et al. 2018; Duev et al. 2019; Turpin et al. 2020), galaxy morphologies (e.g., Dieleman et al. 2015; Pérez-Carrasco et al. 2019; Barchi et al. 2020), or time domain classification (Carrasco-Davis et al. 2019; Gómez et al. 2020).

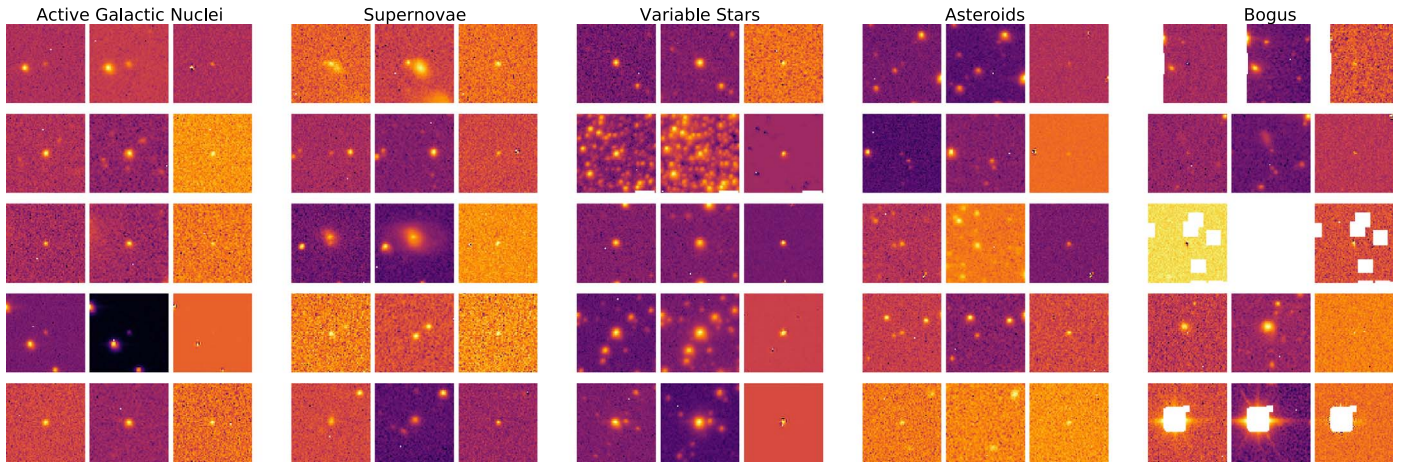
An associated contribution to the stamp classifier is the implementation of a visualization tool called *SN Hunter*,<sup>18</sup> which allows experts to explore SN candidates to further filter alerts, and choose objects to request follow-up. This visualization tool is deployed online and provides a snapshot of the current ZTF data stream within minutes of receiving new alerts.

This work is structured as follows: In Section 2 we describe the data used to train the proposed neural network model, as well as give a brief description of each class and how we gathered labeled data. In Section 3, we describe the data preprocessing, the neural network architecture, the entropy regularizer added to the optimization function, and the experiments run to find the best architecture for the problem at hand. In Section 4, we report and discuss our results in terms of the classification task. We also analyze the contribution to the classification performance of each one of the three images in the alert, as well as the metadata. In Section 5, we describe the SN Hunter visualization tool and the visual criteria used by human experts to choose good candidates to report to the Transient Name Server (TNS),<sup>19</sup> along with an analysis of reported and confirmed SNe by ALerCE using the proposed methodology since 2019 June. We finally draw our conclusions and describe future work in Section 6.

<sup>17</sup> <https://alerce.online/>

<sup>18</sup> <https://snhunter.alerce.online/>

<sup>19</sup> <https://wis-tns.weizmann.ac.il/>



**Figure 2.** Examples of the five classes to be discriminated by using only the first detection. For each class, the triplet of images in each row are science, reference, and difference images from left to right. Each row corresponds to a different candidate.

## 2. Data

An alert within the ZTF stream is defined as a source in the sky that produces a signal five standard deviations higher than the background noise (a five- $\sigma$  magnitude limit; Masci et al. 2018), and which passes a real/bogus filter designed by the ZTF collaboration (Mahabal et al. 2019). When an alert is triggered, an *alert packet* is generated with all the relevant information about the source that triggered the alert (Bellm et al. 2018). The alert packet contains three images, called stamps, which are cropped at 63 pixels on a side (1 pixel = 1") from the original image and centered on the position of the source. In addition, the alert packet contains metadata related to the source, the observation conditions of the exposure, and other useful information (Masci et al. 2018). An example of the three stamps within an alert packet is shown in Figure 1. The stamp in Figure 1(a) is called the *science image* and corresponds to the most recent measurement of the source. The stamp depicted in Figure 1(b) is the *reference image*, which is fixed for a given region and bandpass. It is usually based on images taken at the beginning of the survey and it is built by averaging multiple images to improve its signal-to-noise ratio. The stamp shown in Figure 1(c) is the *difference image*, which shows the change in flux between the science and reference frames (Masci et al. 2018), removing other sources with constant brightness.

Each alert packet represents only two samples in time, the reference and science image exposures, and often is insufficient to correctly classify objects over the full taxonomy of different variable stars, transients, or stochastic sources as in Sánchez-Sáez et al. (2021).

However, our hypothesis is that it is feasible to use the information included in a single alert packet to separate objects into several broad classes, namely AGN, SNe, VS, asteroids, and bogus alerts. Each class presents distinctive characteristics within the image triplet of the first detection alert (see Figure 2), which could be automatically learned by a CNN. In addition to the images, information in the metadata in the alert packet, along with some derived features from the metadata, are important to discriminate among the mentioned classes. The metadata used for the classification task are listed in Table 1, and the distribution of values for each feature per class is shown in Figure A1 in Appendix A. Some of the

distinctive characteristics and metadata features that help to distinguish between objects of each class are the following:

1. *AGN*: Being stochastically variable objects, an alert generated by an AGN should have flux from the source in both the reference and science stamps. Considering this feature alone, it is difficult to discriminate AGNs from other variable sources. Nevertheless, AGNs should lie at the centers of their host galaxies (based on dynamical friction arguments), or appear as (quasi-)stellar objects, in relatively lower stellar density fields. Thus, a change in flux will appear as a variable source, which may lie at the center of a galaxy, or even when the galaxy is not visible they tend to be in lower stellar density fields. In these cases, the alert is likely to be triggered by an AGN. In addition, AGNs are commonly found outside the Galactic plane, as shown in Appendix A. The important metadata features that characterize AGNs are the Star/Galaxy score, or *sgscores* of the first, second, and third closest source from the PanSTARRS1 catalog, which tend to have values closer to 0 (i.e., extended), since AGNs occur in the center of extended galaxies, and the distance of the first, second, and third closest sources in the PanSTARRS1 catalog, which should have *distpsnr1* values consistent with 0, as the nearest source should be the AGN itself combined with large *distpsnr2* and *distpsnr3* values due to the lower source density outside of the Galactic plane. The *classtar* is also useful as more weakly accreting AGN candidates tend to be classified as galaxy-shaped sources by the SExtractor classifier (Bertin & Arnouts 1996).
2. *Supernovae (SNe)*: An alert generated by a SN should appear as a change in flux where no unresolved sources were present. These transients tend to appear near their host galaxies, and their location should be consistent with the underlying host stellar population distribution (e.g., a SN will have a higher probability of arising from a location aligned with the disk than perpendicular to it). As such, most SN detections exhibit a visible host galaxy in both the science and reference stamps, with the flux from the SN arising only in the science and difference images. SN candidates tend to appear outside the Galactic plane, and so the *sgscores*, *distpsnr*, and Galactic

**Table 1**  
List of Metadata of the Alert Used as Features by the Classifier<sup>a</sup>

Feature	Description [units]
<code>sgscore{1,2,3}</code>	Star/Galaxy score of the {first, second, third} closest source from the PanSTARRS1 catalog $0 \leq \text{sgscore} \leq 1$ where a value closer to 1 implies higher likelihood of being a star, $-999$ when there is no source.
<code>distpsnr{1,2,3}</code>	Distance of the {first, second, third} closest source from the PanSTARRS1 catalog, if one exists within $30''$ , $-999$ if there is no source [arcsec].
<code>isdifffpos</code>	t (converted to 1) if the candidate is from positive (science minus reference) subtraction; f (converted to 0) if the candidate is from negative (reference minus science) subtraction.
<code>fwhm</code>	Full Width Half Max assuming a Gaussian core of the alert candidate in the science image from SExtractor (Bertin & Arnouts 1996) [pixels].
<code>magpsf</code>	The magnitude from PSF-fit photometry of the alert candidate in the difference image [mag].
<code>sigmapsf</code>	$1\sigma$ uncertainty in <code>magpsf</code> [mag].
<code>ra, dec</code>	R.A. and decl. of candidate; J2000 [deg].
<code>diffmaglim</code>	$5\sigma$ mag limit in difference image based on PSF-fit photometry [mag].
<code>classtar</code>	Star/Galaxy classification score of the alert candidate in the science image, from SExtractor.
<code>ndethist</code>	The number of spatially coincident detections falling within $1''.5$ going back to the beginning of the survey; only detections that fell on the same field and readout channel ID where the input candidate was observed are counted. All raw detections down to a photometric Signal/Noise $\approx 3$ are included.
<code>ncovhist</code>	The number of times input candidate position fell on any field and readout channel going back to the beginning of the survey.
<code>chintr, sharpnr</code>	DAOPhot (Stetson 1987) chi, sharp parameters of nearest source in reference image PSF catalog within $30''$ .
Ecliptic coordinates	Ecliptic latitude and longitude computed from the <code>ra, dec</code> coordinates of the candidate [deg].
Galactic coordinates	Galactic latitude and longitude computed from the <code>ra, dec</code> coordinates of the candidate [deg].
<code>approx nondetections</code>	<code>ncovhist</code> minus <code>ndethist</code> . The approximate number of observations in the position of the candidate, with a signal lower than Signal/Noise $\approx 3$ .

**Note.** The definitions are from the ZTF avro schemas.

<sup>a</sup> <https://zwickyscience.github.io/ztf-avro-alert/>

latitude features have similar distributions to AGN candidates. However, there are other features that might help to classify SN candidates correctly. For instance, the `chi` and `sharp` parameters from DAOPhot (Stetson 1987), or `chintr` and `sharpnr` of the nearest source in the reference image PSF (point-spread function) catalog within  $30''$ , have different distributions for the SN class, compared to the other classes (see Appendix A). Furthermore, the `isdifffpos` value, which measures whether the candidate is positive or negative in the science minus reference subtraction, should always be 1 for new SN candidates.

3. *Variable Stars (VS)*: The flux coming from variable stars usually appears in both the reference and science stamps. With ZTF's sensitivity, variable stars can be detected within the Milky Way or the Local Group, and thus the alert will typically not be associated with a visible host galaxy in the stamp but rather with other point-like sources. In addition, such alerts will have a higher probability of residing at lower Galactic latitudes and in crowded fields with multiple-point sources within the stamps, given the high concentration of stars in the disk and bulge of our Galaxy. Therefore, VS candidates present a distribution of higher `sgscores`, lower `distpsnr`, and Galactic latitude closer to 0 compared to AGN and SN candidates (see Figure A1).

4. *Asteroids*: Alerts from moving solar system objects will appear only one time at a given position, and thus will show flux only in the science and difference images. Depending on their distance and apparent speed, they may appear elongated in the direction of motion. In addition, such alerts should have a higher probability of residing at lower ecliptic latitudes as shown in Figure A1. Also, new asteroid candidates should always have an `isdifffpos` feature equal to 1.

5. *Bogus alerts*: Camera and telescope optics effects, such as saturated pixels at the centers of bright sources, bad columns, hot pixels, astrometric misalignment in the subtraction to compute the difference image, un baffled internal reflections, etc., can produce bogus alerts with no interesting real source. Bogus alerts are characterized by the presence of NaN pixels due to saturation, single or multiple bright pixels with little or no spatial extension (i.e., smaller than the telescope PSF and nightly seeing), or lines with high or low pixel values that extend over a large portion of the stamp (hot or cold columns/rows). We are currently working to include satellites in this class. However, they may share some image traits with asteroids but are not confined to the ecliptic. According to the estimates shown in Appendix E, bogus alerts comprise a big portion of the total of alerts generated each night by ZTF, with  $\sim 20\%$  of all alerts being bogus,

and  $\sim 60\%$  of them having a single detection. These estimates were carried out by applying the stamp classifier over 176,376 alerts generated by ZTF’s stream. We present a more thorough characterization and definition of the nine types of bogus alerts we have found in ZTF’s alert stream.

We built a training set of ZTF alerts using the labeled set from Sánchez-Sáez et al. (2021), which is a result of cross-matching with other catalogs, such as the ASAS-SN catalog of variable stars (Jayasinghe et al. 2018, 2019a, 2019b, 2020), the Roma-BZCAT Multi-Frequency Catalog of Blazars (Massaro et al. 2015), the Million Quasars Catalog (version 2019 June, Flesch 2015, 2019), the New Catalog of Type 1 AGNs (Oh2015; Oh et al. 2015), the Catalina Surveys Variable Star Catalogs (Drake et al. 2014, 2017), the LINEAR catalog of periodic light curves (Palaversa et al. 2013), Gaia Data Release 2 (Mowlavi et al. 2018; Rimoldini et al. 2019), the SIMBAD database (Wenger et al. 2000), and spectroscopically classified SNe from the TNS database. The asteroid subset was built by selecting the alerts that were near a solar system object, requiring that the `ssdistnr` field in the alert metadata exists. Each sample corresponds to the triplet of science, reference, and difference images of the first detection. The number of samples of AGN, SN, VS, asteroid, and bogus alerts are 14,966 (29%), 1620 (3%), 14,996 (29%), 9899 (19%), and 10,763 (20%), respectively, with a total of 52,244 examples. Under-sampling the labeled set from Sánchez-Sáez et al. (2021) for a better balance between classes, since 3% of SNe would not be considered balanced compared to the rest. The bogus class was built in two steps: We first used *step 1 bogus*, composed of 1980 bogus examples reported by ZTF (based on human inspection), and ran an initial iteration of the proposed classifier detailed in Section 3.2. Then, we added *step 2 bogus*, where another 8783 bogus samples were labeled by our team of experts using the SN Hunter and added to the training set by manually inspecting the samples predicted by an early version of the model as SNe.

Appendix E contains an analysis of bogus alerts present in the training set. Briefly, Figure E1 shows the distribution of different types of bogus alerts in our labeled set, whereby an expert manually assigned type labels to a representative subset of 1000 bogus samples. We stress here that bogus class generation is an ongoing process with different stages that involves labeling by hand. To highlight the current state, we made a 2D U-MAP projection of the bogus samples alongside SNe, differentiating both stages of a two-step bogus-labeling system. This projection, shown in Figure E2, groups alerts with similar triplet images as neighboring or adjacent points. Bogus alerts categorized by step 2 bogus are within a big cluster that mainly overlaps with SNe, which reflects the bias on how step 2 bogus alerts are selected samples that were confused with SNe by early versions of the stamp classifier. We continue adding new bogus alerts in this way. In Appendix E, we analyze in greater detail which type of bogus alerts are the most representative of each of the three clusters present in the U-MAP of Figure E2.

One final point to stress is that a key aim of the stamp classifier is the fast detection of SNe, and therefore the training set consists only of the initial alert from each object, which allows us to estimate probabilities of objects as soon as we receive the alert.

### 3. Methodology

#### 3.1. Data Preprocessing

The standard shape for each stamp within an alert is  $63 \times 63$  pixels; 650 nonsquare stamps were removed from the data set. After removing misshaped stamps, we obtained 14,742 (29%) AGN, 1596 (3%) SN, 14,723 (29%) VS, 9799 (19%) asteroids, and 10,734 (20%) bogus alerts, with a total of 51,594 examples. Some pixels have NaN values due to pixel saturation, bad columns, or stamps from the edges of the camera; all NaN pixels were replaced by a value of 0, giving information about NaNs content within the stamp to the classifier. Preliminary tests showed that smaller images for training led to better results, therefore we cropped all the stamps at the center, getting  $21 \times 21$  pixels images; this size was selected by the hyperparameter random search discussed in Section 3.5. Better results with a small stamp size may be explained by the fact that smaller stamps mean a dimensionality reduction with respect to the original image size in the input of the CNN, and this may be easier for the model to handle. Further analysis of the optimal stamp size for the classification task at hand must be carried out since it might be important for the design of future alert stream-based surveys. Each stamp was normalized independently between 0 and 1 by subtracting the minimum pixel value in the image, then dividing by the maximum pixel value. Finally, a three-channel cube is assembled as input to the classifier, built by stacking the resulting science, reference, and difference images as separate channels, resulting in a  $21 \times 21 \times 3$  image. The metadata are clipped differently for each feature following the values in Table A1, then each feature is normalized by subtracting the mean value of the training set and dividing by the standard deviation.

#### 3.2. Classifier Architecture

The classification model is a CNN based on the real/bogus classifier proposed by Reyes et al. (2018), which is an improvement over Deep-HITS (Cabrera-Vives et al. 2017) by adding rotational invariance to the CNN and analyzing the predictions of the model using Layer-wise Relevance Propagation (LRP; Bach et al. 2015). The specific CNN architecture used in this work is shown in Table 2. In these previous works, metadata were not included for classification.

The input of the neural network has a shape of  $21 \times 21 \times 3$  as explained in Section 3.1. Following the architecture of Reyes et al. (2018), a zero padding is applied to the input, to then augment the batch with rotated versions of itself, as described in Section 3.3. For the convolutional layers, the parameters shown in Table 2 are the filter dimensions and the number of output channels. All convolutional layers, except for the first one, have zero padding (filling the edges of the images with zeros) that preserves the input shape after the convolution. Moreover, all the convolutional layers and fully connected layers have a Rectified Linear Unit (ReLU; Nair & Hinton 2010) activation function (except for the last fully connected one that has a softmax output). The output of the last max-pooling layer, which reduces the dimensionality of the image by selecting the largest values of non-overlapping windows of  $2 \times 2$  pixels, is rearranged (flattened) to a single dimension array for each sample in the batch, to feed the fully connected layers. Then, the rotation concatenation step takes place, stacking the fully connected output representation of the

**Table 2**  
Convolutional Neural Network Architecture

Layer	Layer Parameters	Output Size
Input	...	$21 \times 21 \times 3$
Zero padding	...	$27 \times 27 \times 3$
Rotation augmentation	...	$27 \times 27 \times 3$
Convolution	$4 \times 4, 32$	$24 \times 24 \times 32$
Convolution	$3 \times 3, 32$	$24 \times 24 \times 32$
Max pooling	$2 \times 2, \text{stride } 2$	$12 \times 12 \times 32$
Convolution	$3 \times 3, 64$	$12 \times 12 \times 64$
Convolution	$3 \times 3, 64$	$12 \times 12 \times 64$
Convolution	$3 \times 3, 64$	$12 \times 12 \times 64$
Max pooling	$2 \times 2, \text{stride } 2,$	$6 \times 6 \times 64$
Flatten	...	2304
Fully connected	$2304 \times 64$	64
Rotation concatenation	...	$4 \times 64$
Cyclic pooling	...	64
Concat with BN <sup>a</sup> features	...	$64 + 23$
Fully connected with dropout	$90 \times 64$	64
Fully connected	$64 \times 64$	64
Output softmax	$64 \times 5$	5 (n° classes)

**Note.**

<sup>a</sup> BN stands for batch normalization.

rotated versions of each sample, and passing them through the cyclic pooling layer, where an average is applied in the stacked dimension. The metadata features are first processed by a batch normalization layer that learns an optimal bias and scale to normalize the data. The normalized features are concatenated to the output of the cyclic pooling. The concatenated representation passes through two fully connected layers. Finally, a softmax function is applied to the output of the last fully connected layer to obtain the estimated probabilities for each of the five classes. A glossary about CNNs and their training is presented in Appendix B.

### 3.3. Rotational Invariance

Astronomical objects present within a stamp usually have a random orientation. It has been shown that imposing rotational invariance on a classifier improves its cumulative accuracy for some classification problems (e.g., Dieleman et al. 2015, 2016; Cabrera-Vives et al. 2017; Reyes et al. 2018). In this work, rotational invariance is achieved by feeding the neural network with  $90^\circ$ ,  $180^\circ$ , and  $270^\circ$  rotated versions of the original input batch  $x$ . Defining  $r$  as a  $90^\circ$  rotation operation, then the samples within the extended batch will be  $B(x) = [x, rx, r^2x, r^3x]$  after applying the rotations. At the last step of the architecture before the softmax output layer, a cyclic pooling operation is performed, which is an average pooling over the representation of the dense layer for each rotated example. A scheme of the procedure described in this section is shown in Figure 3.

### 3.4. Entropy Regularization

When the CNN model is trained using cross entropy as the loss function to be minimized, the classification confidence of the model is very high, resulting in a distribution of output probabilities with saturated values of 0s and 1s without populating the values in between, even for wrong classifications. In this case, there is no insight of certainty (relative probabilities between classes) of the prediction because most estimated probabilities for each class were either 0 or 1. In

order to provide more granularity to the astronomers, who revise SN candidates based on the probability of the classification reported by the model to later request follow-up observing time, we added the entropy of the predicted probabilities of the models as a regularization term, to be maximized during training (Pereyra et al. 2017). By maximizing the entropy of the output probabilities, we penalize predictions with high confidence in order to get better insight into cases where the stamps seem equally likely to belong to more than one class. The loss function  $\mathcal{L}$  per sample is as follows:

$$\mathcal{L} = - \underbrace{\sum_{c=1}^N y_c \log(\hat{y}_c)}_{\text{cross-entropy}} + \beta \underbrace{\sum_{c=1}^N \hat{y}_c \log(\hat{y}_c)}_{\text{entropy regularization}}, \quad (1)$$

where  $N$  is the number of classes,  $y_c$  is the one-hot encoding label (a value of 1 in the corresponding index of class and 0 for the rest) indexed by  $c$ ,  $\hat{y}_c$  is the model prediction for class  $c$ , and  $\beta$  controls the regularization term in the loss function. Further explanation on the role of the loss function in the training process of a neural network is given in Appendix B.

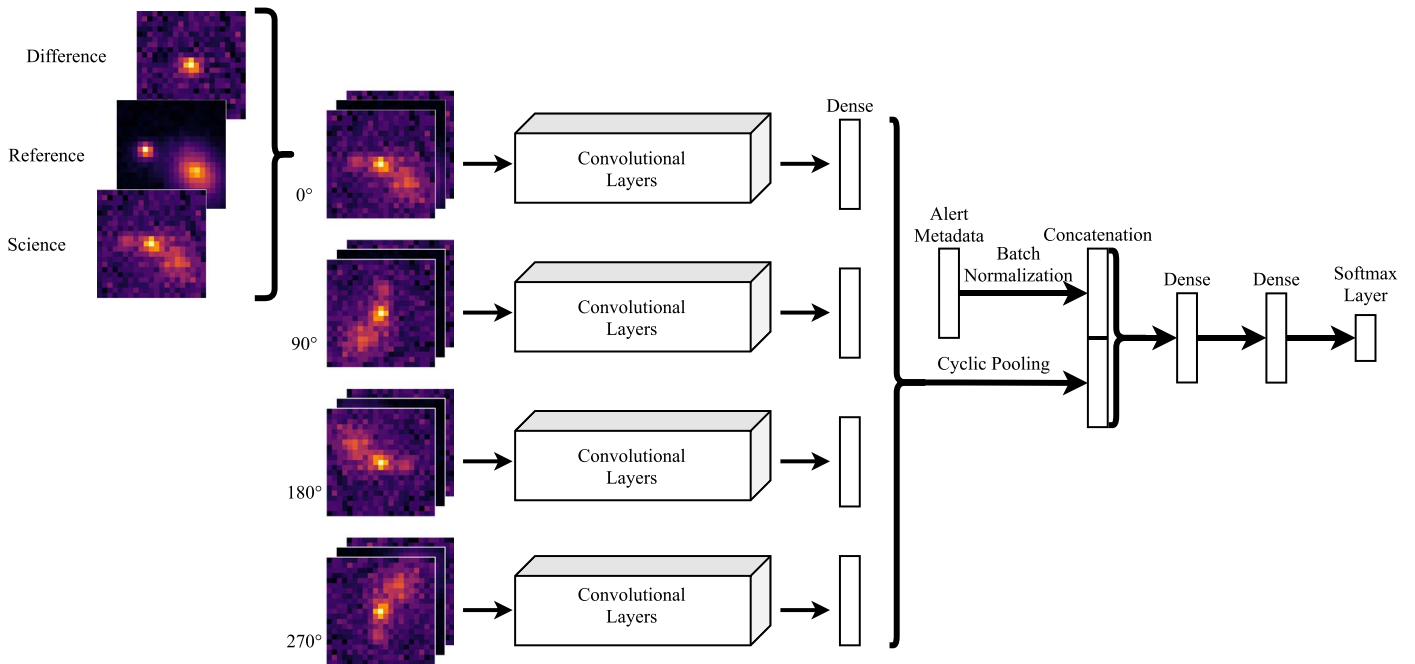
### 3.5. Experiments

A hyperparameter search was done by randomly sampling 133 combinations of the parameters shown in Table 3. For each combination of hyperparameters, we trained five networks with different initial random weights. The initial maximum number of iterations (presenting a single batch per iteration) was 30,000, evaluating the loss in the validation set every 10 iterations to save the best model thus far. After the first 20,000 iterations, if a lower loss is found on the validation set, 10,000 more iterations are performed. The validation and testing subsets were sampled randomly only once, taking 100 samples per class from the whole data set, obtaining 500 samples for each of the mentioned subsets. The remaining samples were used in the training set. For each training iteration, the batch was built to contain roughly the same number of samples per class. We used Adam (Kingma & Ba 2017) as the updating rule for the network parameters during training, with  $\beta_1 = 0.5$  and  $\beta_2 = 0.9$ . Further details on the updating rules of a neural network and the Adam optimizer are described in Appendix B.

To account for the relevance of each part of the input, which is comprised of the three images and metadata features, we trained several versions of the stamp classifier, each of them with a different combination of images and metadata features. First, we trained the stamp classifier using combinations of the three images (without features). Second, we trained a random forest (Breiman 2001) to classify our training set but using the features only, in order to obtain a feature importance ranking. Once we got the feature importance, we trained different stamp classifier models (with the three images) by adding one feature at a time, from the most important to the least important according to the ranking, and measured the accuracy for the corresponding model with the aggregated feature. For each of these models, we trained the model five times to account for variance due to random initialization parameters.

## 4. Results

In this section, we first describe our results in terms of the classification task for the five classes. Then, we change our



**Figure 3.** CNN enhanced with rotational invariance. The box *Convolutional Layers* refers to those described in Table 2, from the first convolutional layer to the last pooling layer. For each sample, the science, reference, and difference images are concatenated in the channel dimension, obtaining an image input of dimension  $21 \times 21 \times 3$ . For each sample within the sampled batch, rotated versions are generated as described in Section 3.2 and fed to the CNN. After the first dense layer, the Cyclic pooling is performed. The metadata features are passed through a batch normalization layer, and its output is concatenated with the cyclic pooling output (clipping values for each of the metadata features are specified in Table A1). Then, the concatenation goes through two fully connected layers, and finally, a softmax function is applied to estimate the output probabilities.

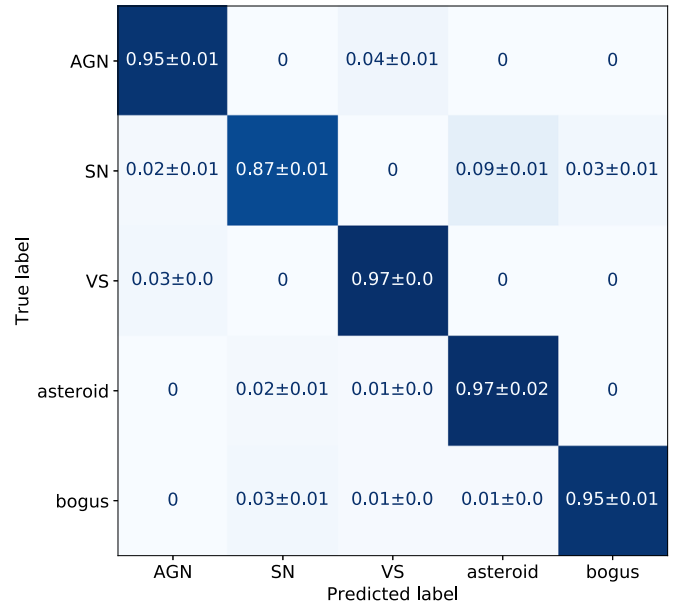
**Table 3**  
Hyperparameter Random Search Values

Hyperparameter	Random Search Values
Learning rate	$5e-3$ , $1e-3$ , $5e-4$ , $1e-4$ , $5e-5$
Regularization parameter ( $\beta$ )	0, 0.3, 0.5, 0.8, 1.0
Batch size	16, 32, 64
Image size	21, 41, 63
Dropout rate	0.2, 0.5, 0.8
CNN kernel size	3, 5, 7

focus to the detection of SN candidates, since our main interest in this work is to discover extremely young transient candidates to be observed with follow-up resources. Further applications of this early classification system might include rapid detection of extreme variability in AGN or tracking solar system objects.

The following results correspond to the best model (including metadata) in the search for hyperparameters, which adopts a batch size of 64 samples, a learning rate of  $1e-3$ , a dropout rate of 0.5, a CNN kernel size of 5, an image size of  $21 \times 21$  pixels, and a regularization parameter of  $\beta = 0.5$ . Appendix D contains the details on how this model was selected. We use accuracy<sup>20</sup> to compare models since the validation and test sets are balanced; achieving  $0.95 \pm 0.005$  in the validation set and  $0.941 \pm 0.004$  in the test set.

Figure 4 shows the confusion matrix for the test set consisting of using five realizations of the proposed model. With our five-class model, we recover  $87\% \pm 1\%$  of the SNe, with only  $5\% \pm 2\%$  of false positives. For completeness, we also report the confusion matrix of the stamp classifier when no metadata features are included in the fully connected layers (see

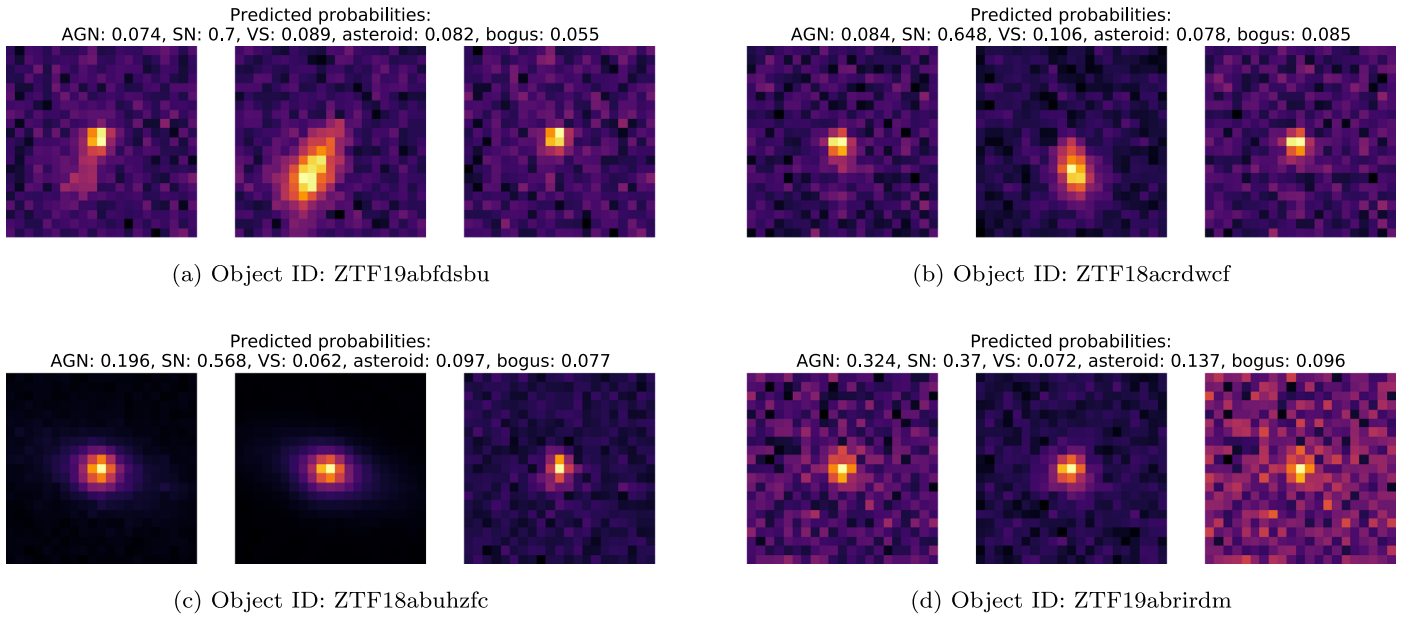


**Figure 4.** Average confusion matrix for the test set using five different realizations of the stamp classifier with metadata.

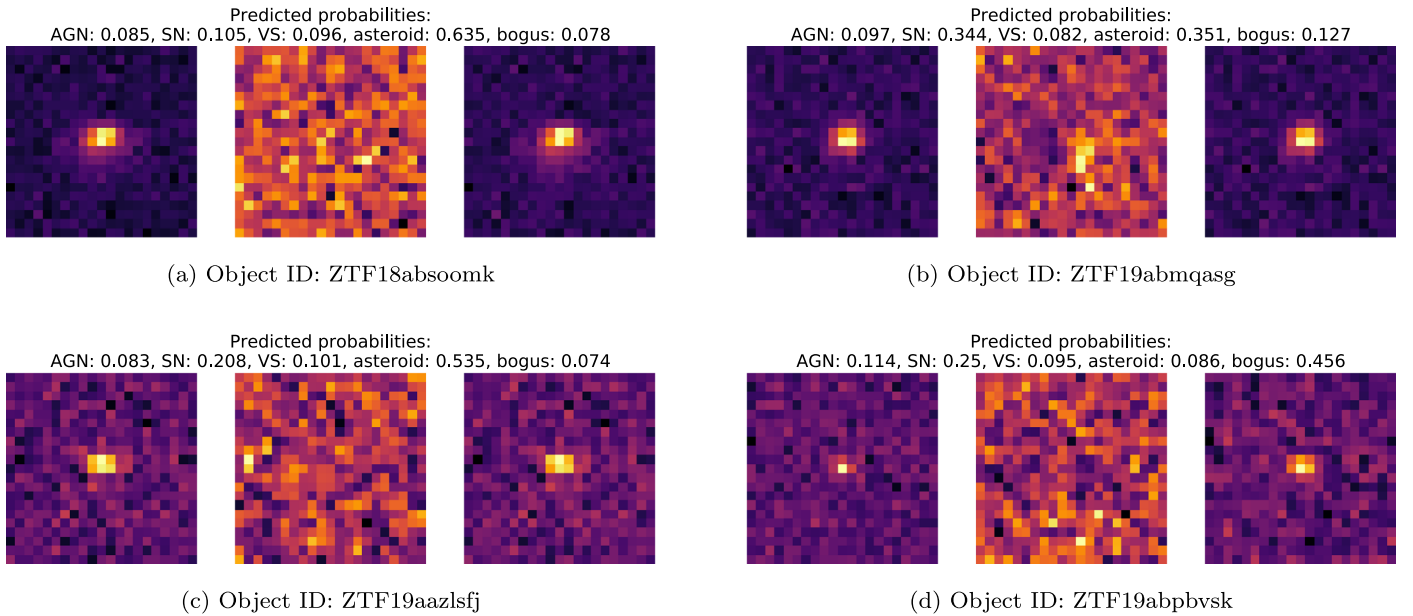
Figure C1), which has a test set accuracy of  $0.883 \pm 0.006$ , recovering  $80\% \pm 2\%$  of the SNe in the test set, with  $10\% \pm 4\%$  of false positives.

By inspecting the predictions made by our model for each SN sample in the test set, we found that the results are in agreement with our initial expectations regarding the class discrimination described in Section 2, and the characteristics presented within the three stamps for each sample. Figure 5 shows SNe examples from TNS that have been correctly classified by our model, where in most cases a host galaxy is

<sup>20</sup> accuracy =  $\frac{N^{\circ} \text{ correct classifications}}{\text{Total } N^{\circ} \text{ of samples}}$



**Figure 5.** Correctly classified SN examples, with their respective predicted probabilities according to the proposed model. Panels (a) and (b) show typical examples of well-classified SNe, where the presence of a host galaxy within the stamps increases the chances of an SN alert being triggered. Panels (c) and (d) show small confusions between SN and AGN, due to the spatial coincidence of the transient with the center of the host galaxy.



**Figure 6.** Incorrectly classified SN examples with their respective predicted probabilities by the proposed model. In Panels (a)–(c), the SNe are classified as asteroids. The SN in panel (d) is classified as a bogus alert, which might be caused by the small size of the PSF, confusing the classifier with a hot pixel or a cosmic ray, which usually occupies a very narrow portion of the stamp at the center. In all cases, the absence of a clear host galaxy within the stamps reduces the probability of a SN alert being triggered.

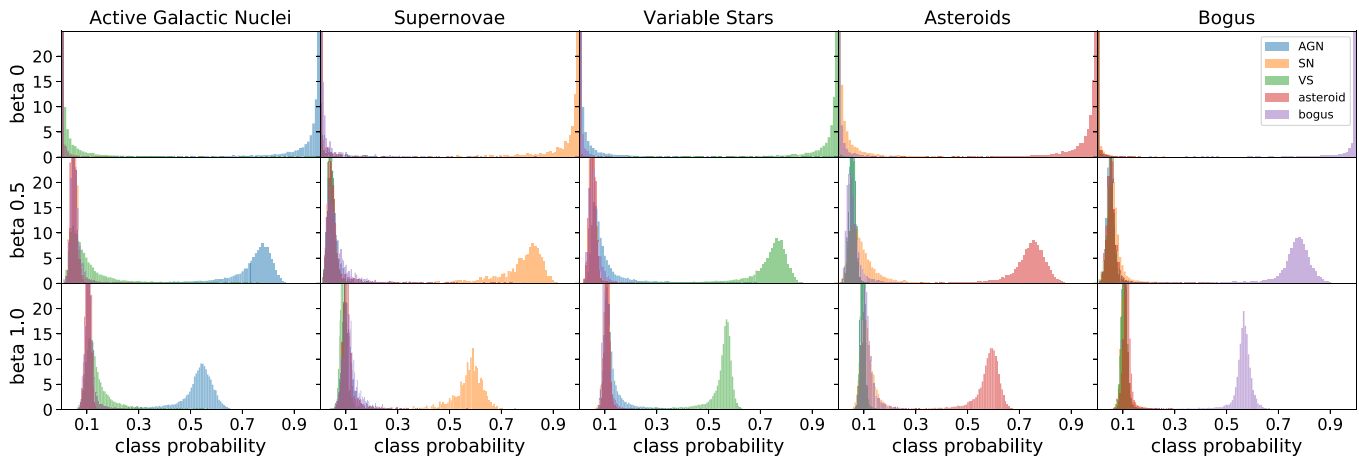
present, which is a good indicator of an alert triggered by a SN. In the examples shown in Figures 5(c) and (d), the second most likely class is AGN, due to the spatial coincidence of the transient with the center of the host galaxy.

In Figure 6, incorrectly classified examples are shown. The examples in Figures 6(a)–(c) are SNe from TNS classified as Asteroids by our model. The absence of a host galaxy in these cases reduces the probability of an alert to be triggered by a SN. In the samples shown in Figure 6(d), confusion occurs between SN and bogus alerts. In this case, the confusion is likely due to the small size of the PSF for some observations, which leads to

confusion between true variables and hot pixels or cosmic rays (these often appear as single or a few adjacent bright pixels on the image).

It is worth highlighting again that the results of our model are achieved using the first alert only. According to the confusion matrix, the most probable misclassification for SN candidates is asteroid and bogus classes. This confusion between SN, asteroid, and bogus alerts could be fixed by looking at the second alert of the same object. If the second alert exists, it is safe to discard the bogus and asteroid classes, since it is extremely unlikely that the same bogus error or a





**Figure 7.** Probability distribution for each of the classes in the training set, for different values of the regularization constant  $\beta = \{0, 0.5, 1.0\}$ . For the model without regularization ( $\beta = 0$  shown on the top plot), the probability distribution saturates to 1 or 0. Increasing  $\beta$  to 0.5 or 1.0 decreases the saturation and spreads the distribution of predictions made by the model (mid and bottom plots).

moving object will appear in the exact same location in consecutive images, unless the alert is near a bright star that produces pixel errors due to saturation.

An example of the effect of the regularization term discussed in Section 3.4 is depicted in Figure 7. Considerable differences in the distribution of the predicted probability for each class can be observed by varying  $\beta$  between 0 and 1, since both terms in Equation (1) are expected values of log probabilities. In the case of  $\beta = 0$ , the predictions are mostly saturated around 0 or 1 for the SN, VS, Asteroids, and Bogus alert classes, creating difficulties to identify stamps that seem equally likely to belong to more than one class, because every sample is mapped to similar levels of high certainty. As the value of  $\beta$  increases, the saturation of predicted values decreases, spreading the predicted probability distributions and emphasizing the different levels of certainty between predictions of different samples. The order of predicted probabilities for each sample does not change significantly by varying  $\beta$ , achieving 99% of accuracy in the test set by checking whether the correct label lies in the highest two predicted probabilities for different  $\beta$ . The use of regularization to find noticeable differences in the predicted probabilities could be helpful to an expert for evaluating the output of the classifier, gaining better insight into how reliable the classifications are.

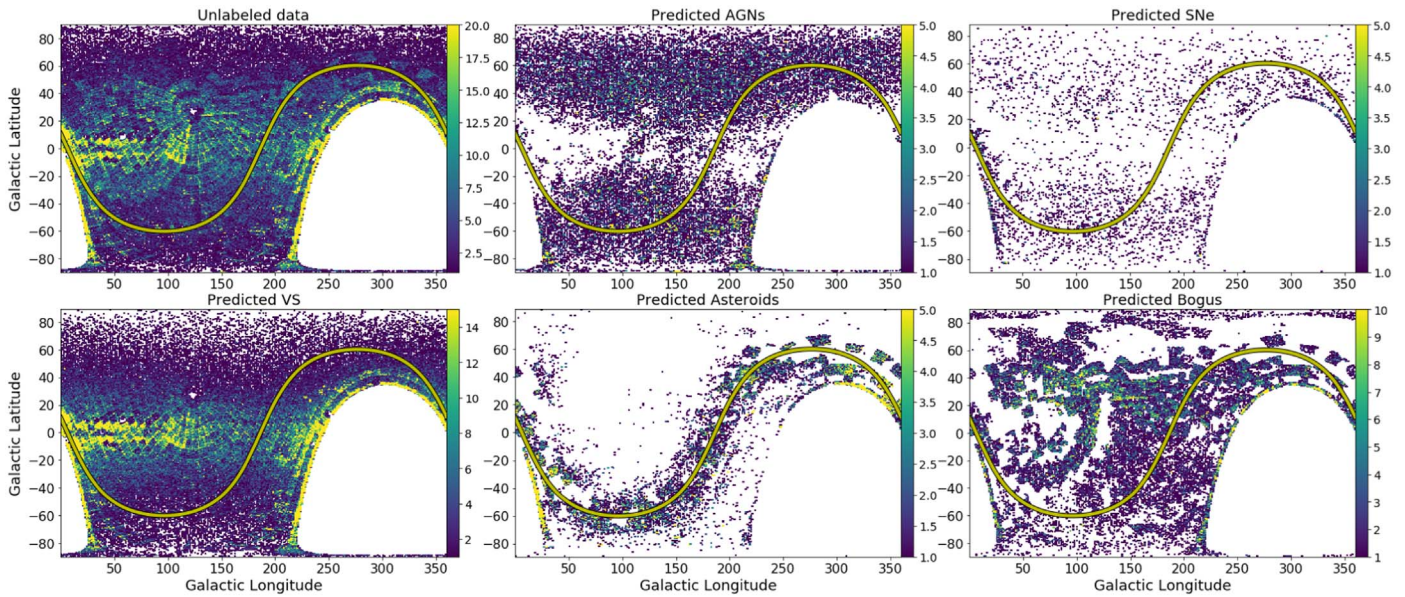
As a consistency check, we predicted the classes of unlabeled candidates using the stamp classifier, in order to compare their spatial distribution to the expected spatial locations for each class as mentioned in Section 2. To gather the unlabeled candidates, we queried objects using the ALERCE API<sup>21</sup> by selecting 390,498 first alerts of different objects, chosen to be uniformly distributed over the full sky coverage of ZTF, where 325,582 of the alerts come from objects with more than one alert (SNe, AGN, and VS) and 64,916 come from objects with only one alert to have a better representation of asteroids and bogus candidates. Figure 8 shows the spatial distribution of the predictions made by our model over the unlabeled data. As expected, due to the extinction of extragalactic sources (SNe and AGNs) in the Galactic plane, the spatial distribution for these sources has a lower density of predicted candidates at low Galactic latitudes.

On the contrary, the spatial distribution of VS candidates is more concentrated toward the Galactic plane. In the case of asteroids, these are found near the ecliptic. It is also possible to see a slight trend of predicted SNe near the ecliptic due to the confusion with the asteroid class when there is no apparent host galaxy in the stamp, as shown in Figure 6. In addition, we show in Figure C2 in Appendix C the distribution for the same unlabeled data classified by the CNN without including features. The presence of predicted extragalactic objects (SNe and AGNs) within the Galactic plane and a higher density of predicted asteroids far from the ecliptic are noticeable. Even though the images alone have important information to classify the five classes, the metadata features are essential to improve the accuracy of the classifier in the labeled data set as shown in Appendix C, in addition to obtaining the expected spatial distribution for each class.

We further extended our analysis by comparing the predictions of our model in the unlabeled data set, with the ones made by the feature-based light-curve classifier from Sánchez-Sáez et al. (2021), which is able to classify a finer taxonomy of objects but requires at least six detections in one of the two bands. For more details, see Section 3.6 of Förster et al. (2021). Here we summarize the important results. The stamp classifier predictions strongly agree with the ones of the light-curve classifier. The stamp classifier finds 78% of the SN classified by the light-curve classifier, 85% for AGNs, and 96% for VS. The main confusions in the SN class (false positives) are 9% of AGNs, 6% of VS, 4% of asteroids, and 3% of bogus alerts. The false positives of AGNs are 4% of SN and 1% of VS, and the confusion of VS is only 3% with AGNs. Classifications of objects in the unlabeled set comparing the light-curve classifier and stamp classifier are shown in Table D2. To further account for the performance of the stamp classifier in the asteroids and bogus classes, we randomly selected 20,000 asteroids and 20,000 bogus objects predicted by the stamp classifier. The proportion of objects with a single detection as of 2021 February is 98% and 96%, for asteroids and bogus alerts respectively, for which we expect a single detection (except for a small proportion of bogus alerts).

To understand how the stamp classifier performs in different conditions, we computed the probability assigned to alerts of each class in the training set as a function of the specific value

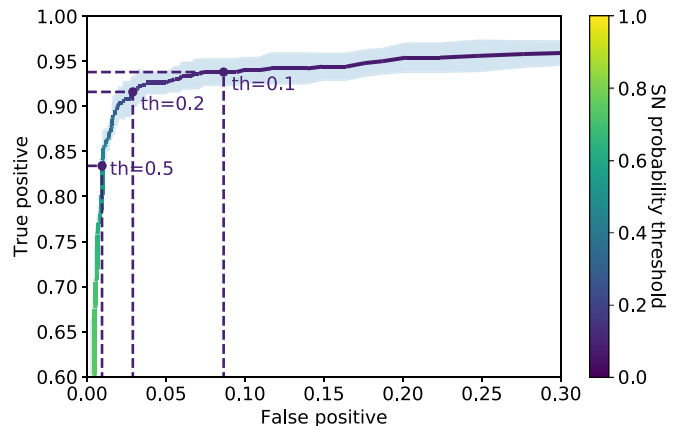
<sup>21</sup> <https://alerceapi.readthedocs.io/en/latest/>



**Figure 8.** Spatial distribution for the unlabeled data, and distribution of predictions per class. The colorbar indicates the density of points. The ecliptic is shown with a yellow line with black edges. The distributions are shown as a 2D histogram of the density of alerts. Extragalactic sources (SNe and AGNs) are found outside the Galactic plane. On the contrary, VS are concentrated in the Galactic plane. Asteroids are near the ecliptic.

of each of the features given to the classifier; these probabilities are shown in Figures E3 and E4 in Appendix E. Here we can inspect the features where the probability assigned to the correct class decreases, showing the classifier performance in different regimes. In what follows, we remark on some examples that support our hypotheses for class separability according to features mentioned in Section 2. For example, variable stars with low  $sgscore1$  are less likely to be classified correctly, as well as variable stars with higher  $distpsnr$  (Figures E3(b), (d), and (f)), since we anticipate that variable stars are in higher stellar density regions (i.e., in the plane or bulge of the Milky Way). In the case of photometry measurements, variable stars have a lower probability of being correctly classified at higher magnitudes, while AGNs have a higher probability of getting correctly classified. The inverse is true for  $sigmapsf$  (Figures E3(i) and (j), respectively), which is probably due to the higher error in the estimated magnitude owing to the host galaxy in the case of AGNs, which is not present in variable stars. We can see that the assigned probability to asteroids decays far from the ecliptic in Figure E4(e) in line with their known distribution, while the probability assigned to AGN is lower near the Galactic plane in Figure E4(g) due to dust extinction. In Figure E4(l), the probability assigned to each class is shown for different values of signal-to-noise ratio. Higher values of signal-to-noise ratio mean less probability of being an AGN, since these objects are distant and are usually found within a host galaxy; the signal-to-noise ratio for the AGN source is smaller. Note that the signal-to-noise ratio is not added as a feature to the classifier, but it is encoded in the photometry features  $magpsf$  and  $sigmapsf$  from which we computed the signal-to-noise ratio.

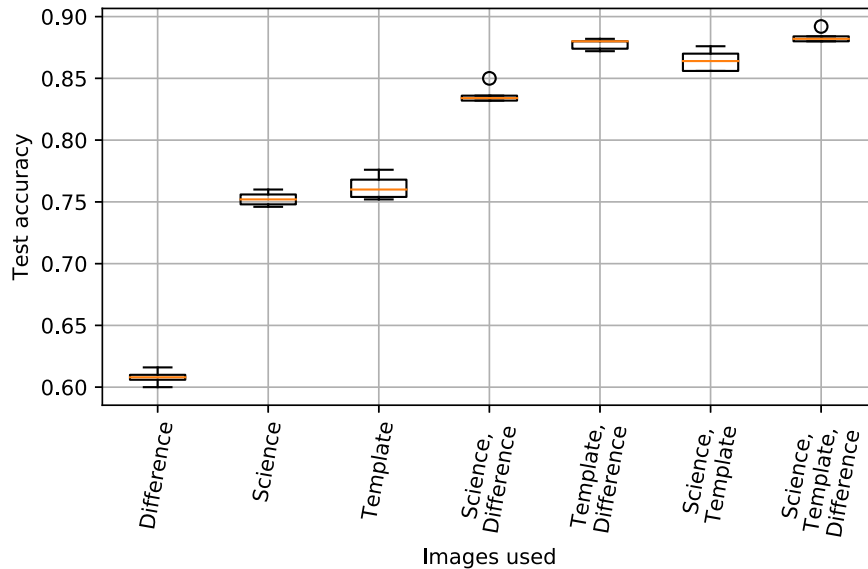
We use the stamp classifier on a daily basis to filter suitable SN candidates to report for follow-up. The filtered candidates are inspected by experts<sup>22</sup> to choose the most reliable candidates to report among the ones indicated by the classifier.



**Figure 9.** ROC curve with SN detection threshold. The colorbar shows the threshold that a sample’s predicted SN probability must surpass in order to be assigned as a member of the SN class. For a SN probability threshold of 0.1, 0.2, and 0.5, the false-positive ratio is 0.87, 0.03, and 0.01, respectively, while the true positive ratio is 0.94, 0.92, and 0.83, respectively.

Therefore, it is important to control the false-positive ratio and the amount of classified SNe events. To understand this trade-off, we computed the Receiver Operating Characteristic (ROC) curves depicted in Figure 9. To build the ROC curve, we converted the classification problem into a detection problem by making a binary classification between SN versus the rest of the classes (AGN, VS, asteroids, and bogus alerts). Using the predicted probabilities in the test set of each alert being a SN, we varied the threshold value (minimum probability) necessary to assign the SN class to an alert and change the operation regime of the model. By choosing a high SN probability threshold, the false-positive ratio can be reduced in order to decrease the number of false candidates in the list for inspection by experts, while keeping a high true positive ratio. For instance, for a SN probability threshold of 0.1, 0.2, and 0.5, the false-positive ratio is 0.87, 0.03, and 0.01, respectively, while the true positive ratio is 0.94, 0.92, and 0.83,

<sup>22</sup> For a full list of reporters, please check [the list of reported objects by ALeRCE in TNS](https://www.wis-tns.org/object/2021mfa), such as <https://www.wis-tns.org/object/2021mfa>.



**Figure 10.** Accuracy of stamp classifier model when varying the images available at the input, isolated points are outliers. As we can observe, the most important image is the template, which gives information about nearby objects and context to the classifier, as well as the science image, which is slightly less informative according to the test accuracy, probably due to the higher noise compared to the template. Using the three images at the same time is better, but surprisingly not much better compared with using the science plus template, or difference plus template. This might be important to consider for classification purposes when designing alert-based surveys such as LSST.

respectively. Our model is suitable to be used to process large volumes of alerts when limited resources for manual inspection and confirmation by means of follow-up observations are available.

#### 4.1. Images and Metadata Feature Relevance

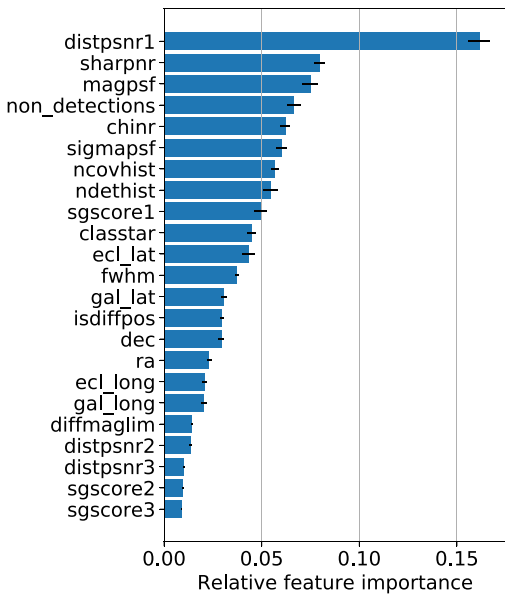
In this section, we explain the results of the experiments designed to account for the relevance of each image in the input (science, difference, and template) and the importance of each feature in the classification. Figure 10 shows the results of training the stamp classifier with a different combination of the input images. Here we highlight the main conclusions from these experiments. When using a single image as input, the image that gives the most accuracy is the template image. We hypothesize that the template image gives information about the context of the alert (e.g., stellar density, host galaxy, bright star, no counterpart), which is valuable information for the specific classes we are trying to correctly classify, as explained in Section 2. Notably, the template image has a better signal-to-noise ratio than the science image (which also contains contextual information), explaining the difference in the accuracy of the stamp classifier model when trained on each of these. Surprisingly, using two images (science and template, or difference and template) as inputs considerably improves the classification accuracy, almost reaching the accuracy of using the three images at the same time. This result could be considered when designing alert-based surveys where bandwidth or storage of the alert streaming is an important restriction, but still, the best accuracy is achieved when using the three images combined to train the model.

For analyzing the relevance of each metadata feature for the classification of the alerts, we trained a random forest to classify the alerts only using the features, as mentioned in Section 3.5, to obtain a feature importance ranking. The feature importance ranking is shown in Figure 11(a), where the highest score feature is `distpsnr1`, which indicates the distance to the first closest source from the PanSTARRS1 catalog, giving a

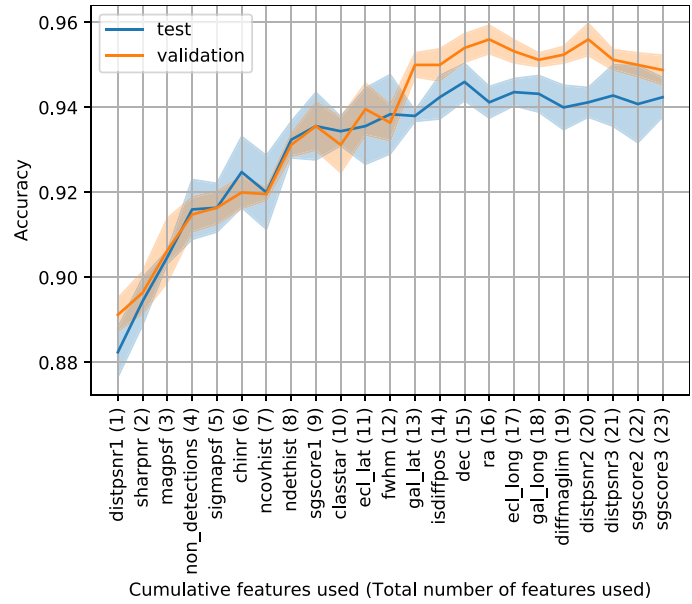
measure of the density of objects near the alert and providing important context information. Another relevant feature is `sharpnr`, which is useful to discriminate the SN class among the others. Notice the relevance of the Galactic and ecliptic latitudes. The former provides context for stars (which have a higher probability of lying at low Galactic latitude) and extragalactic sources (which have a lower source density at low Galactic latitudes due to extinction), while the latter provides context for asteroids (which have a higher probability of lying at low ecliptic latitude). To assess the impact of each feature in the stamp classifier accuracy, we trained different models by adding one feature at a time in the order given by the feature ranking (from more important to less important) as detailed in Section 3.5. The change in accuracy for each combination of accumulated features is shown in Figure 11(b). The model gets higher accuracies for both validation and test sets by adding more features up to the galactic latitude feature, where the accuracy in the validation set goes further up compared to the test set. We interpret this as a sort of overfitting to the validation set, which in this particular case is not harmful to the performance of the model because the accuracy on the test set still increases and converges to a value when including additional features, without any statistically significant drop. We argue that, while the accuracy on the test set does not drop, adding a new feature might be valuable extra information to the classifier, but further checking is needed in a larger set. For instance, using the predictions by the light-curve classifier by Sánchez-Sáez et al. (2021) in the cases of SN, AGN, and VS classes, or the number of detections in the asteroid and bogus classes, as mentioned in Section 4.

## 5. Model Deployment and SN Hunter

The SN Hunter is a visualization tool that allows the user to inspect SN candidates classified by the model in real time in order to select good targets for follow-up observations. The interface of the SN Hunter is shown in Figure 12. At the left of the interface, a celestial map shows the position of each candidate

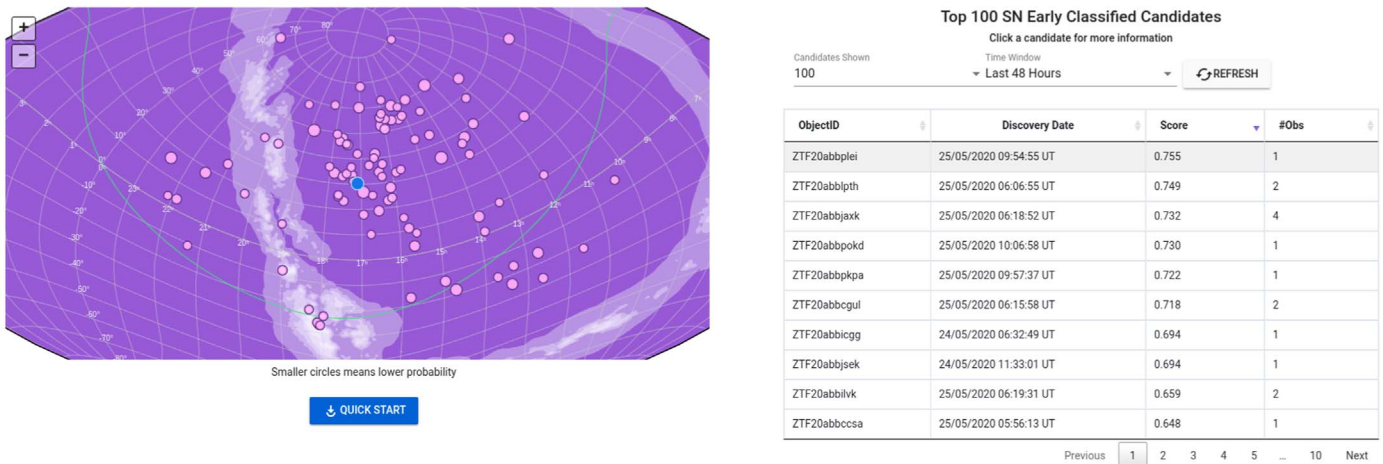


(a) Average feature importance according to 10 random forest classifiers trained to predict the label only using the metadata features.



(b) Accuracy of stamp classifier models when cumulative adding one feature at a time according to the feature ranking in (a).

**Figure 11.** Feature relevance analysis. The feature ranking in (a) was used to build stamp classifier models by aggregating one feature at a time and evaluate its accuracy (b). We can see, for instance, that the galactic and ecliptic latitudes are more important than longitude, since the former indicates the distance to the Galactic plane and ecliptic, which is useful for classification. Also, we see that `distpsnr1` is the most important feature as mentioned in Section 2, and also `sharpnr` which helps to separate the SN class from the rest. This feature analysis is useful when using classification models that do not provide the relevance of each input dimension explicitly.

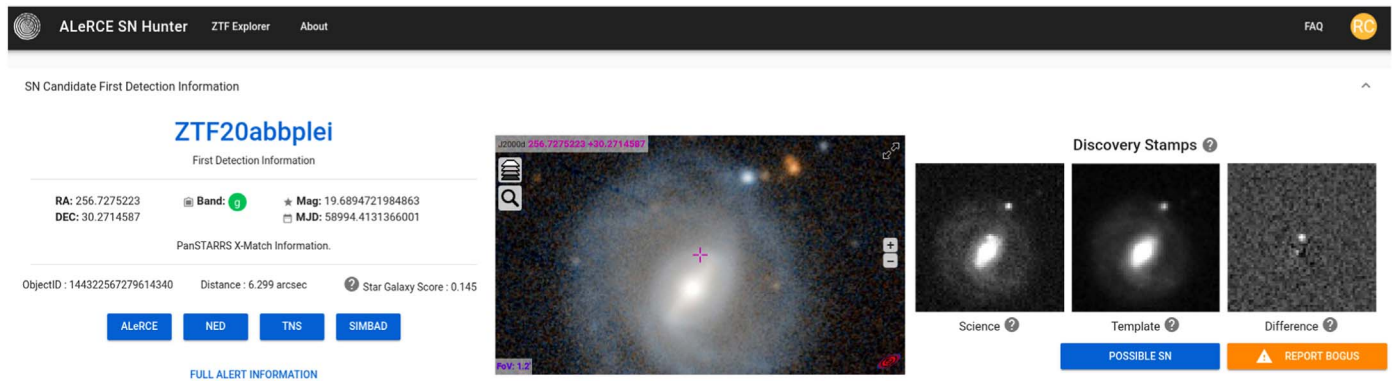


**Figure 12.** SN Hunter, a tool for the visualization of SNe candidates. On the left side, the location of each candidate in the sky with respect to the Galactic plane and the ecliptic are depicted. On the right side, a selection of the top candidates is listed, initially ordered by SN probability score from the stamp classifier. The list of candidates can be sorted by other parameters, and updated/refreshed to include newly ingested alerts.

with a circle, where the size of the circle is proportional to the class probability assigned by our model, with the map centered on the R.A. (`ra`) and decl. (`dec`) coordinates of the alert. The Milky Way plane is highlighted by the regions with lighter shades of purple. The green curve in the map represents the ecliptic, where SN candidate alerts are more likely to be triggered by asteroids instead of real SNe. The right side of the interface provides a table where the highest probability SN candidates are listed. The table shows the ZTF Object ID which uniquely identifies each astronomical alert, the discovery date specifying the day, month, year, and time where the first alert was triggered, the corresponding SN probability (score) from the stamp

classifier, and the number of available alerts in the  $r$  and  $g$  bands (`#Obs`) since the discovery date. The list can be sorted by object, discovery date, score, or the number of alerts. The total number of high probability candidates shown in the table and the maximum age of the candidates can be modified by the user. By clicking on a given candidate row, a new visualization panel is deployed as shown in Figure 13, with detailed information for the selected candidate.

The visualization panel contains some metadata on the left side (see Figure 13), including the ZTF Object ID (which is a clickable link to open the objects as a new tab within the ALerCE online main frontend, `ra` and `dec` coordinates of the



**Figure 13.** Candidate visualization in the Supernova Hunter tool. On the left side, the SN candidate ID is shown as a clickable link to the ALeRCE frontend, with relevant metadata such as `ra`, `dec`, magnitude, date, etc. At the bottom, there are links to other sources of information, including ALeRCE, NED, TNS, and the SIMBAD astronomical database. In the middle of the figure, there is a colored image from Aladin. On the right side, the stamps of the first detection are shown, along with buttons for reporting the candidate as an eventual bogus alert or as a possible SN.

alert, the filter in which the first detection was made, the PSF magnitude, and the observation date. Below is the PanSTARRS cross-match information, containing the Object ID, distance to the first closest known object, and the `classtar` score of the first closest known object, where a score closer to 1 implies a higher likelihood of it being a star. The buttons below this information, from left to right, correspond to queries with the ALeRCE frontend, the NASA Extragalactic Database (NED<sup>23</sup>), TNS, and the SIMBAD astronomical database (Wenger et al. 2000) around the position of the candidate. Finally, the full metadata associated with the first alert of the SN candidate are linked below these buttons. The middle panel of Figure 13 contains an interactive color image from PanSTARRS DR1 (Chambers et al. 2019), centered around the source using Aladin (Bonnarel et al. 2000; Boch & Fernique 2014); this image is also available in the main frontend of ALeRCE. The right panel of Figure 13 provides the science, reference, and difference stamps of the first detection. It is also possible to sign in with a user account and label candidates as either a possible SN or bogus alert by clicking the corresponding buttons below the image stamps. These can be used to build up larger training sets, as well as select candidates for the Target and Observation Managers (TOMs).

We implemented the CNN stamp classifier using TensorFlow 1.14 (Martín et al. 2015) and deployed it to classify the streaming alerts from ZTF’s Kafka server.<sup>24</sup> The timespan between a ZTF exposure and its first arrival as an alert from the stream is  $14.6 \pm 4.5$  minutes. Once the alert is received by ALeRCE, it takes a few seconds for the candidate to be listed in the Supernova Hunter tool for expert inspection. Further details about the complete processing pipeline are described in Förster et al. (2021).

### 5.1. Additional Visual Selection Criteria

We note that the SN candidate sample presented in this section and Section 5.2 resulted from an older version of the Stamp Classifier which relied only on the three images within the first alert and did not use features for SN classification. Moreover, some of the filtering steps we applied manually are no longer

necessary now that features are included (we note these below). As shown in Appendix C, even without the metadata features, the classifier provides reasonably high accuracy (only 6% worse than the model with features). Regardless of whether features are included or not, we found it critical to visually inspect the predicted SNe candidates in order to weed out misclassifications and submit more reliable candidates to TNS.

There are some common characteristics among the higher confidence SN candidates. As mentioned in Section 2, most confirmed SNe are located on top or near an extended galaxy. If there is no galaxy within the stamps, then it is more likely that it is a variable star or asteroid when the candidate is located near the Milky Way or the ecliptic, respectively, or a bogus alert. In some cases, it is difficult to tell if the nearest source to the alert in the science image is an extended galaxy or star; for these, a search of archival catalogs and/or an assessment of the spectral energy distribution can further aid classification. Therefore, the star galaxy score from PanSTARRS in Figure 13 should be closer to 0, indicating that the extended source is more likely to be an extended galaxy. Real SN should have a positive flux in the difference image, so we removed candidates that have negative flux in the latter by checking if the field `isdiffpos` value in the metadata is false; this is automatically done in the current pipeline. It is also important to check that the object is visible in the difference image.

Another relevant feature is the shape of the candidate, which should be similar to other stars with fuzzy edges and generally symmetric in shape. If the shape of the candidate is sharp (pixelized) or very localized, it might be a cosmic ray or a defect of the CCD camera. Alternatively, if it is elongated, it could be an asteroid or a satellite (often seen as a streak or multiple small dashes due to rotational reflections). After doing all of these checks, if the candidate is not convincing enough, then it is helpful to look at the next detections when available and search for the characteristics mentioned, which can be done using the ALeRCE frontend by clicking the ALeRCE button in the SN Hunter tool and querying directly that specific candidate’s data. The 100 highest probability SNe candidates each day are manually inspected by astronomers of the ALeRCE team, and all of them must be in agreement before a candidate is reported to TNS. As a qualitative analysis, we report that the confusion of the SN class will depend on the weather. In optimal weather conditions, the PSF size could be a few pixels long and the classifier confuses SN with bogus

<sup>23</sup> The NASA/IPAC Extragalactic Database (NED) is operated by the Jet Propulsion Laboratory, California Institute of Technology, under contract with the National Aeronautics and Space Administration. <https://ned.ipac.caltech.edu/>.

<sup>24</sup> <https://kafka.apache.org/>

**Table 4**

Spectroscopically Observed Candidates Discovered by ALERCE, with a Total of 971 SNe and 24 non-SN Objects (5 TDE, 5 Galaxies, 4 Nova, 2 Other, 2 Cataclysmic Variables (CV), 2 AGNs, 2 Unknowns, 1 Variable Star, 1 M Dwarf)

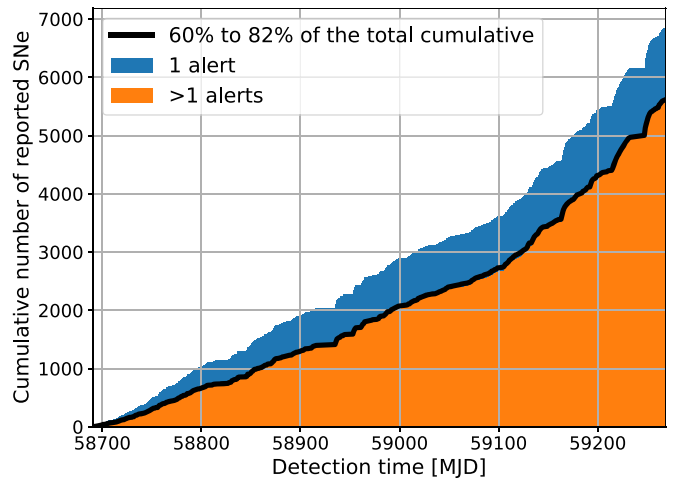
Confirmed Class	Spectroscopically Observed Candidates
SN Ia	676
SN II	148
SN Ic	24
SN Ia-91T-like	22
SN IIn	21
SN IIP	16
SN Ib	14
SN Iib	13
SN Ic-BL	10
TDE	5
Galaxy	5
Nova	4
SN Ia-pec	4
SN Iax[02cx-like]	4
SN I	3
SN Ia-91bg-like	3
SN	3
SN Ib/c	3
SLSN-II	3
Other	2
CV	2
unknown	2
AGN	2
SN Ib-pec	1
Varstar	1
SN Ibn	1
M dwarf	1
SLSN-I	1
SN Icn	1

samples of type cosmic rays. On regular nights, asteroids and satellites are a key source of contamination, and then a small fraction of image subtraction issues. Alerts triggered near known variable stars or asteroids and classified as SNe by the SN Hunter are removed from the list of 100 candidates to be visually inspected by astronomers.

### 5.2. Reported and Confirmed Supernovae

From 2019 June 26 to 2021 February 28, we have reported 6846 new SN candidates to TNS, increasing this number by 11.8 SNe per day on average, of which 995 have been observed spectroscopically. Table 4 shows the number of candidates for each confirmed class, of which 971 were confirmed as SNe. Non-SNe objects reported were five TDEs, five galaxies, four Nova, two Other, two cataclysmic variables (CV), two AGNs, two unknowns, one variable star, and one M dwarf. Even though TDEs are not SNe, follow-up of these events is still of significant interest due to their relative scarcity (van Velzen et al. 2020). In summary, taking into consideration the conservative final candidate selection done by the team of astronomers to perform spectroscopic confirmation, our reported and confirmed candidates have around 2% contamination by non-SNe objects.

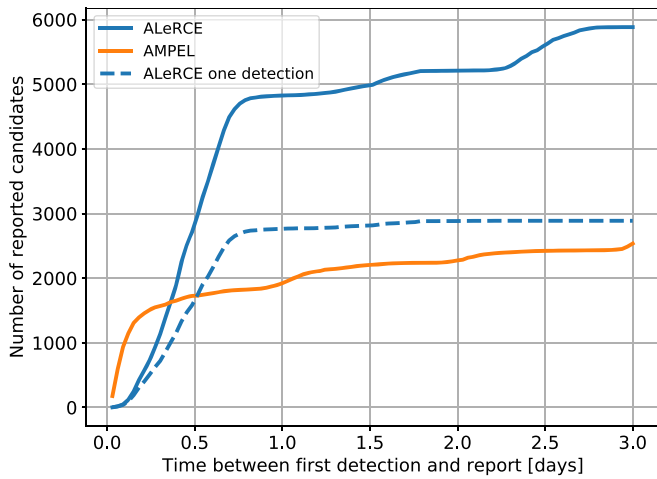
In Figure 14, we show the cumulative distribution of candidates reported to TNS from 2019 June 26 to 2021



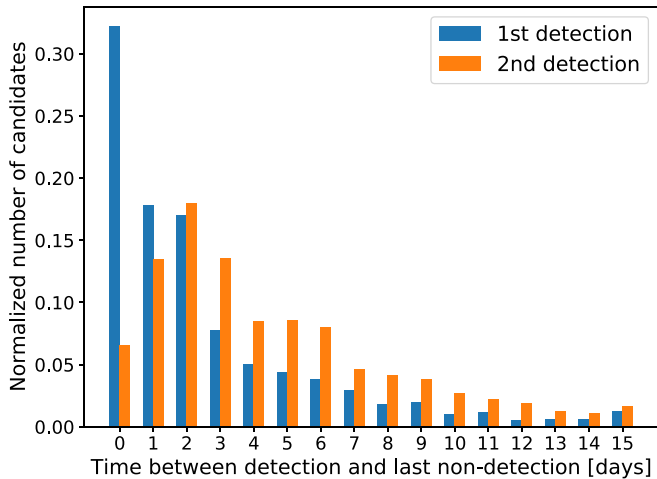
**Figure 14.** Cumulative number of reported SNe since we started reporting on 2019 June 26 to 2021 February 28. The average rate of reporting is 11.8 candidates per day. Currently,  $\approx 82\%$  of our reported candidates are detected with multiple alerts (purity), implied they are true SNe, while  $\approx 12\%$  have only one detection and thus less certainty. We have been increasing the purity of the reported candidates roughly linearly from  $\approx 60\%$  to  $\approx 82\%$ .

February 28. The cumulative distribution is separated into two parts, namely the alerts with more than one detection to date (orange) and the alerts with a single detection to date (blue). We can consider the percentage of candidates with more than one detection to be a lower bound of real non-moving astronomical objects since we do not have the true label for reported alerts; we define this as “purity,” since multiple associated detections are a clear sign of a real non-moving astronomical object rather than a moving object or bogus alert. Candidates with only single detections to date could be due to several reasons: moving objects, bogus alerts, relatively short transients that were only above the detection threshold for a short period of time, and objects in locations that have not been visited again by the public ZTF Survey since object detection. We have been increasing the purity approximately linearly, from  $\approx 60\%$  to  $\approx 82\%$  (reported candidates detected with multiple alerts to date).

For comparison purposes, we gathered the objects reported by both ALERCE and AMPEL (Alert Management, Photometry, and Evaluation of Lightcurves, which is an internal ZTF classification effort; Nordin et al. 2019) to TNS, and compared the reporting times within 3 days after the first detection. Figure 15 shows the cumulative histogram of reporting times to TNS for ALERCE and AMPEL, along with the cases where reports were done by ALERCE before having the second detection in the public stream (one detection). Approximately 42% of the candidates reported by ALERCE were based on a single detection. An important difference between both systems is the visual inspection by experts in the reporting process to TNS. According to Nordin et al. (2019), AMPEL reports candidates automatically using their “TNS channel,” which produces more reported candidates than our system, within 12 hr after the first detection. As described in this work, our system’s final stage so far relies on human inspection, checking, and reporting, which occurs within 10–24 hr after the first detection, without reporting transients already reported by AMPEL (only two cases were reported after AMPEL). Therefore, we report new candidates to TNS within a day after



**Figure 15.** Cumulative distribution of time between the first detection and the reporting time from TNS, for candidates reported by ALerCE and AMPEL. The full distributions are shown with solid lines, and the distributions of reporting time for candidates with a single detection are shown with segmented lines.



**Figure 16.** Histogram of time between first (second) detection and the last nondetection of the ALerCE reported candidates.

the first detection. Additionally, since ALerCE is largely reporting candidates with a single detection, 70% (4825) of the reports were sent within 1 day after the first detection, compared to 25% (1925) from AMPEL.

Figure 16 shows the distribution of time between first or second detection and last nondetection for candidates reported by ALerCE. Based on the data shown in Figure 16, the average time between the last nondetection and the first detection is 4.2 days, with 8.1 days for the second detection. Reporting candidates only after the second detection would result in a delay of 3.9 days on average, which represents a potentially critical timespan to measure the spectra at the early stages of the transient event, as required in order to achieve the science goals described in Section 1. As mentioned before, ALerCE currently does not report candidates that were previously reported by other groups using data from ZTF, therefore our candidates reported using a single detection

increase the bulk of objects available for early follow-up of transients that were not found by other groups. We will report already reported candidates in the near future, since this adds the additional information that the candidates passed our visual inspection test. In addition, the work presented is a starting point toward our goal of developing an automatic reporting system of the most highly confident subset of SN candidates.

## 6. Conclusion and Future Work

As part of the ALerCE Broker processing pipeline, we identified characteristics of the images and metadata within the ZTF alert stream that allow us to discriminate among SN, AGN, VS, asteroids, and bogus alerts using the first detection only. In order to solve this classification problem automatically and quickly identify the best SN candidates to perform follow-up, we trained a CNN. The inputs to this classifier are the science, reference, and difference images, and part of the metadata of the first detection alert. In addition, our CNN architecture is invariant to rotations within the stamps and was trained using an entropy regularized loss function. The latter is useful to improve human readability in predicted probabilities per class, in terms of certainty assigned to each sample, so an expert can gain better insights into the actual nature of the transients when inspecting SN candidates.

Among all five classes that our CNN can classify, it achieves an accuracy of  $0.941 \pm 0.004$  on a balanced test set, while in the SN class reaches a true positive rate and a false-positive rate of  $87\% \pm 1\%$  and  $5\% \pm 2\%$ , respectively. By manually inspecting the classification of each sample, we found that the incorrectly classified objects are in concordance with our hypotheses regarding the separability of classes using only the first detection images. Moreover, the CNN model successfully classified the alerts in the labeled set by using the images only, but when applied to unlabeled data we found some flaws by inspecting the spatial distribution of each predicted class, for instance, a concentration of extragalactic sources within the Galactic plane, and a higher density of asteroids far from the ecliptic. By giving the alert metadata as additional features to the classifier, we find that the spatial distributions of the events are in agreement with the expectations, according to their tentative nature. More specifically, extragalactic classes (SNe and AGNs) are found outside the Galactic plane, VS have a higher density of predicted objects within the Galactic plane, and asteroids are found around the ecliptic.

The proposed CNN classifier is deployed and its predictions are publicly available in an especially designed visualization tool for inspection of candidates with high SN class probability, called the SN Hunter. The predictions are also available in the ALerCE main frontend. This tool shows relevant information about the SN candidates in order to facilitate their analysis, and we used it to report SN candidates on TNS for follow-up. We also presented a visual inspection methodology that relies on the information presented in the Supernova Hunter tool, such as the probability assigned by our classifier to the SN class, alert metadata, position in the sky with respect to the Galactic plane and the ecliptic, number of detections, etc. By manually inspecting candidates using the SN Hunter tool, from 2019 June 26 to 2021 February 28, our team has reported 6846

candidates for follow-up, out of which 995 were tested, and 971 were spectroscopically confirmed as SN. Additionally, the interface allows experts to manually label bogus alerts and SN candidates alike, which helps improve our training set.

As many as 70% of the candidates reported to TNS by ALERCE were reported within 1 day after the first detection, and 42% of all the candidates reported by ALERCE were done by using a single detection, where 82% of the total alerts reported by ALERCE have multiple detections to the date of writing this document, confirming extragalactic nature. Since ALERCE does not report objects that had been previously reported, these results correspond to the transients that were not detected or chosen by other groups, therefore adding new early transient reports to TNS.

We are currently working on improving the training set by adding more examples from confirmed SNe and manually adding bogus candidates to the training set. We run simple but insightful experiments to understand the contribution of each image (science, reference, and bogus) and features (metadata) to the classification task. Furthermore, we are exploring ideas for model interpretability, adding visualization tools that may help understand why the model predicts a given class for a given event. We are working on using LRP (Bach et al. 2015; Montavon et al. 2019) and occlusion techniques (Zeiler & Fergus 2014) to show what part of the input influences the decision in a specific way, so the expert can use it to choose better candidates.

Regarding the performance of the model, an additional step would be to extend the system to be able to process more than a single alert while keeping the capability of performing well with only one alert. New approaches about how to achieve this have been explored in Carrasco-Davis et al. (2019) and Gómez et al. (2020), feeding a neural network sequentially with the data available so far, and improving the prediction every time a new measurement arrives. In the near future our efforts regarding alert classification, and particularly the SN detection problem, will aim toward the automatization of the entire process of classification of the data stream and reporting objects for follow-up, eliminating or bringing expert assistance to a minimum.

The methodology proposed in this work is suitable to other streams of data based on alerts, such as ATLAS (The Asteroid Terrestrial-impact Last Alert System; Tonry et al. 2018) and the Vera C. Rubin Observatory Legacy Survey of Space and Time (LSST; Ivezić et al. 2019). The latter presents a further challenge in terms of the amount of data generated, restrictions in processing time due to the data generation rate, the larger number of filters, the lack of comparison catalogs at the survey’s depth, the smaller field of view per stamp (currently planned to be only 6” × 6”) and limited contextual information, and the possibility that either the science or reference image may not be contained in the alerts. We think our work on ZTF data will be a valuable precursor for the next generation of large etendue telescopes.

The authors acknowledge support from the National Agency of Research and Development’s Millennium Science Initiative through grant IC12009, awarded to the Millennium Institute of Astrophysics (RC, ER, CV, FF, PE, GP, FEB, IR, PSS, GC, SE, Ja, EC, DR, DRM, MC) and from the National Agency for Research and Development (ANID) grants: BASAL Center of

Mathematical Modelling AFB-170001 (CV, FF, IR, ECN, CS, ECI) and Centro de Astrofísica y Tecnologías Afines AFB-170002 (FEB, PSS, MC); FONDECYT Regular #1171678 (PE), #1200710 (FF), #1190818(FEB), #1200495 (FEB), #1171273 (MC), #1201793(GP); FONDECYT Postdoctorado #3200250 (PSS); FONDECYT Iniciación #11191130 (CV); Magíster Nacional 2019 #22190947 (ER). This work was funded in part by project CORFO 10CEII-9157 Inria Chile (PS). The authors acknowledge financial support from the Spanish Ministry of Science, Innovation, and Universities (MICIU) under the 2019 Ramón y Cajal program RYC2019-027683 (LG).

*Software:* Aladin (Bonnarel et al. 2000), Apache ECharts<sup>25</sup>, Apache Kafka<sup>26</sup>, Apache Spark (Zaharia et al. 2016), ASTROIDE (Brahem et al. 2018), Astropy (Astropy Collaboration et al. 2013), catsHTM (Soumagnac & Ofek 2018), Dask (Rocklin 2015), Jupyter<sup>27</sup>, Keras (Chollet et al. 2018), Matplotlib (Hunter 2007), NED (Steer et al. 2016), Pandas (McKinney 2010), Prometheus<sup>28</sup>, Python<sup>29</sup>, scikit-learn (Pedregosa et al. 2011), Simbad-CDS (Wenger et al. 2000), Tensorflow (Martín et al. 2015), Vue<sup>30</sup>, Vuetify<sup>31</sup>, PostgreSQL<sup>32</sup>.

## Appendix A

### Exploring the Relationship between Features and Classes

In Figure A1 the distribution of each feature for each of the different classes in the training set are shown. In Table A1 the limiting values for each of the metadata features, applied right before feeding the stamp classifier.

**Table A1**  
Clipping Values for Each Feature

Feature	[Min Value, Max Value]
sgscore1	[-1, max]
distpsnr1	[-1, max]
sgscore2	[-1, max]
distpsnr2	[-1, max]
sgscore3	[-1, max]
distpsnr3	[-1, max]
ifwhm	[min, 10]
ndethist	[min, 20]
ncovhist	[min, 3000]
chintr	[-1, 15]
sharpnr	[-1, 1.5]
nondetections	[min, 2000]

**Note.** Max or min in the clipping range for each feature means that the maximum and minimum value is preserved for that feature, respectively.

<sup>25</sup> <https://echarts.apache.org>

<sup>26</sup> <https://kafka.apache.org/>

<sup>27</sup> <https://jupyter.org/>

<sup>28</sup> <https://prometheus.io/>

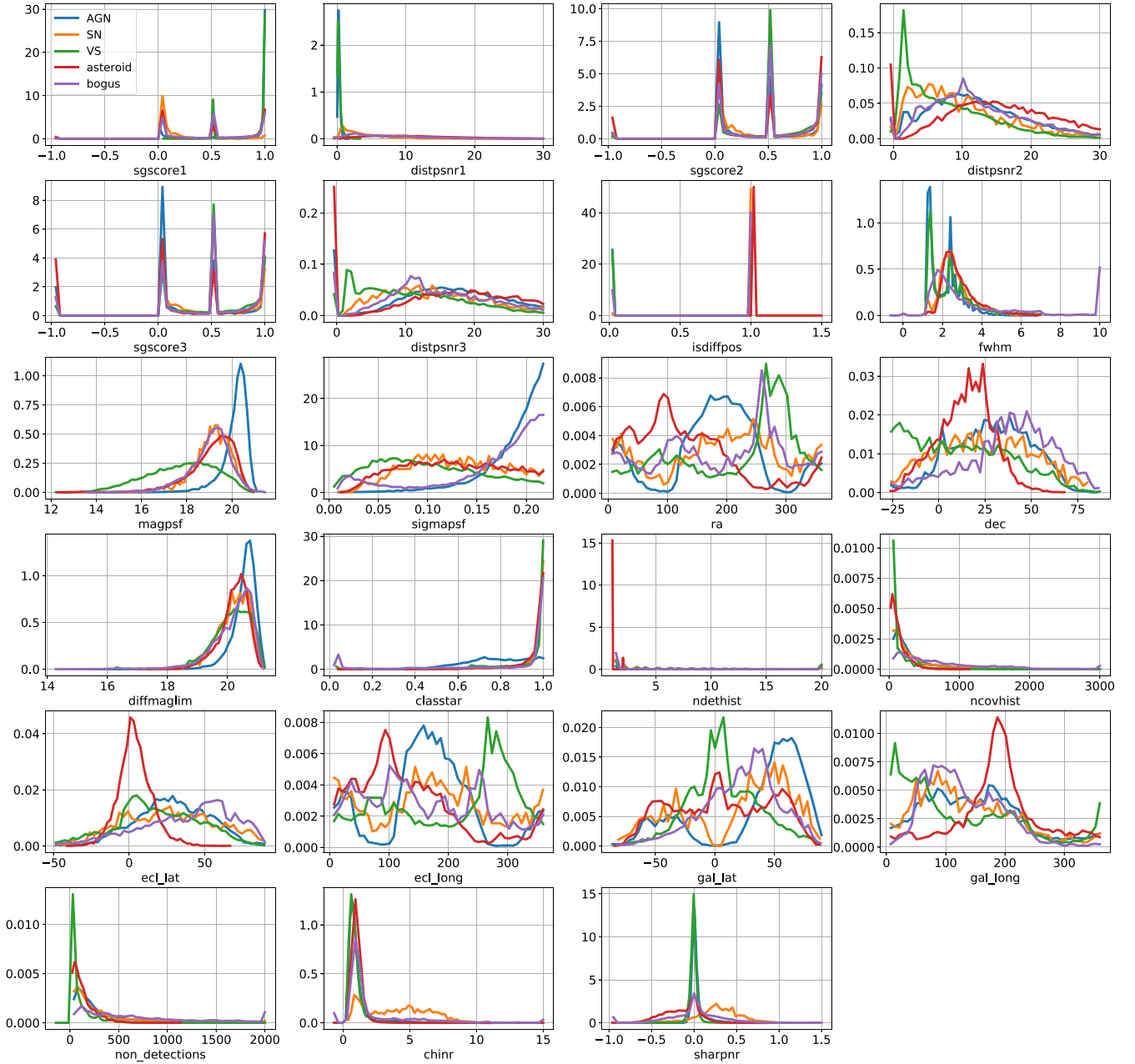
<sup>29</sup> <https://www.python.org/>

<sup>30</sup> <https://vuejs.org/>

<sup>31</sup> <https://vuetifyjs.com/>

<sup>32</sup> <https://www.postgresql.org/>





**Figure A1.** Feature distribution per class of the labeled data set. Each feature was clipped to the values given in Table A1.

## Appendix B CNN Glossary and Training

### B.1. CNN Architecture

1. *Fully connected layer:* Artificial neural networks (ANNs) are mathematical models that are mostly used for classification or regression. ANNs make use of basic processing units called *neurons*, which receive vectors  $\mathbf{x}$  of data as input, then apply a linear function to them, followed by a nonlinear activation function. These neurons are grouped in *layers*, which are called *fully connected layers*. The output produced by a set of neurons of a specific fully connected layer is calculated as:

$$\mathbf{y} = \phi(\mathbf{W}\mathbf{x} + \mathbf{b}), \quad (\text{B1})$$

where  $\mathbf{x} \in \mathbb{R}^n$  is the input of the layer,  $\mathbf{y} \in \mathbb{R}^m$  is the output of the layer,  $\mathbf{W} \in \mathbb{R}^{m \times n}$  is a matrix of parameters called *weights*,  $\mathbf{b} \in \mathbb{R}^m$  is a vector which contains the so-called *biases* of the layer, and  $\phi(\cdot)$  is a nonlinear activation function that follows the linear transformation of  $\mathbf{x}$ . There are many types of nonlinear activation functions, the most commonly used are:

$$\text{sigmoid}(x) = \frac{1}{1 + e^{-x}}, \quad (\text{B2})$$

$$\text{tanh}(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}, \quad (\text{B3})$$

$$\text{ReLU}(x) = \max\{0, x\}. \quad (\text{B4})$$

To be precise,  $\mathbf{W}$  and  $\mathbf{b}$  are referred to as the parameters of a fully connected layer, and they are modified during

training to be optimized for the task at hand. Fully connected layers can be sequentially stacked one after the other to integrate an ANN model. For instance, an ANN of two layers, is defined as:

$$\mathbf{z} = \phi(\mathbf{W}^{(1)}\mathbf{x} + \mathbf{b}^{(1)}), \quad (\text{B5})$$

$$\mathbf{y} = \phi(\mathbf{W}^{(2)}\mathbf{z} + \mathbf{b}^{(2)}). \quad (\text{B6})$$

The parameters of the ANN are  $\theta = (\mathbf{W}^{(1)}, \mathbf{b}^{(1)}, \mathbf{W}^{(2)}, \mathbf{b}^{(2)})$ . The way of grouping neurons and layers in an ANN is called the *architecture* of an ANN.

2. *Softmax output layer*: A commonly used activation function at the output of ANN models is the sigmoid ( $x$ ), whose output is bounded by (0, 1), and can be interpreted as the probability of activation of a neuron, a property useful for binary classification. A generalization of the aforementioned function, useful for multiclass classification models, is the *softmax* activation function, usually referred to as *softmax output layer*, where there are  $K$  neurons  $x_i, i \in \{1, \dots, K\}$ , and it is desired to assign a probability to each one, hence, requiring that they add up to one. This is done by the softmax activation function, defined as:

$$\text{softmax}(x_i) = \frac{e^{x_i}}{\sum_{j=1}^K e^{x_j}}, \quad i \in \{1, \dots, K\}. \quad (\text{B7})$$

3. *Convolutional layer*: ANN with fully connected layers are limited to vector-like inputs, and they do not take into account the presence of correlation between adjacent features. To overcome this limitation, and preserve a degree of spatial or temporal correlation in the input of models, CNNs were proposed. The main component of CNNs is *convolution layers*, which apply a filter or kernel to the input of the layer by a convolution operation. Similar to a fully connected layer, convolutional layer outputs are calculated as follow:

$$\mathbf{y} = \phi(\mathbf{W} * \mathbf{x} + \mathbf{b}), \quad (\text{B8})$$

where  $\mathbf{x}$  stands for the input of the layer,  $\mathbf{y}$  is the output of the layer,  $\mathbf{W}$  is a set of filters to apply by convolution to the input,  $\mathbf{b}$  is the vector of biases for each filter, and  $\phi(\cdot)$  is the activation function. In this case, the  $*$  operation between  $\mathbf{x}$  and  $\mathbf{W}$  is a convolution. The model used in this work applies convolutions to images, so  $\mathbf{x} \in \mathbb{R}^{e \times f \times g}$  and  $\mathbf{y} \in \mathbb{R}^{u \times v \times l}$  are 3d tensors, while  $\mathbf{W} \in \mathbb{R}^{d \times d \times t \times l}$  and  $\mathbf{b} \in \mathbb{R}^l$ . The calculation of every element  $y_{i,j,k}$  of  $\mathbf{y}$  is derived from the operation of the convolutional layer as follows:

$$y_{i,j,k} = \sum_{m,n,p} x_{i-m,j-n,p} W_{m,n,p,k} + b_k, \quad (\text{B9})$$

where every element  $i, j, k$  of the tensor  $\mathbf{y}$  is calculated by moving the filters of  $\mathbf{W}$  over the tensor  $\mathbf{x}$  and applying Equation (B9). Each time  $\mathbf{W}$  moves over the first two dimensions of  $\mathbf{x}$ , it skips  $S$  pixels, where  $S$  is called stride. After applying the convolutional layer, the first two dimensions of  $\mathbf{y}$  are smaller in size than the ones of  $\mathbf{x}$ . The spatial dimensions (first and second dimension) of the tensors  $\mathbf{x}$  and  $\mathbf{y}$  relate to each other as follows:

$$U = \frac{E - D}{S} + 1, \quad (\text{B10})$$

where  $U$  is the size of any of the spatial dimensions of  $\mathbf{y}$ ,

$E$  is the size of the respective spatial dimension of  $\mathbf{x}$ ,  $D$  is the respective spatial dimension of  $\mathbf{W}$  and  $S$  is the stride used in the convolution operation.

4. *Zero padding*: It is a commonly used technique to preserve the spatial dimensions of the input  $\mathbf{x} \in \mathbb{R}^{e \times f \times g}$  at the output  $\mathbf{y} \in \mathbb{R}^{u \times v \times l}$  of a convolutional layer. Zero padding consists on adding 0s to the edges of the spatial dimensions of  $\mathbf{x}$ . For a convolutional layer of stride  $S = 1$ , and kernel size  $D$ , the zero-padded input to the convolutional layer must have dimensions such as  $\mathbf{x} \in \mathbb{R}^{(e+[D/2]) \times (f+[D/2]) \times g}$ , where  $D/2$  is the amount of zero padding included to achieve the same spatial dimensions of  $e = u \wedge f = v$ , between the layer's input  $\mathbf{x}$  and output  $\mathbf{y}$ .
5. *Max pooling*: Pooling layers are used in CNNs to reduce the spatial dimensionality of their inputs. The max pooling used in the model shown in Figure 3 returns the maximum value within a window of its input  $\mathbf{x}$ , in the same way as a convolutional filter, this maximum value extraction window is rolled across the spatial dimensions of the input. In the case of the architecture shown in Figure 3, the pooling window is of dimension  $2 \times 2$  with a stride of 2, i.e., without overlapping of the window, yielding a spatial dimensionality reduction by half each time max pooling is applied.
6. *Batch normalization layer*: It works as a trainable normalization layer that has different behavior during training and evaluation of the model. During training, the batch normalization layer calculates the mean and variance of each feature to normalize them and compute an exponential moving average of mean and variance of the training set. After training the model, for its evaluation, the whole population statistics adjusted during training are used to normalize evaluation inputs. Batch normalization not only normalizes input values to have a mean value near 0 or a variance value near 1, but it also contains a linear ponderation of these inputs that allows their scaling and shifting. This layer allows the model to emphasize or ignore specific inputs, acting as a regularizer and speeding up training.
7. *Dropout*: It is an operation that is usually applied at the output of fully connected layers, and it is used as a regularizer of the model to avoid overfitting layers with a large number of neurons. Similar to a batch normalization layer, dropout performs different operations during training and evaluation. The dropout operation is defined by the dropout rate  $DR \in [0, 1]$ , which is a parameter that, at the training phase of the model, defines the probability of setting each of its inputs to 0, and multiplying the values not set to 0 by  $1/(1 - DR)$ , such that the sum over all the input values remains the same. At each training step, a percentage  $DR$  of the outputs of a fully connected layer will not be used, reducing the effective size of that layer. On the other hand, when using the model after training, dropout is deactivated. The desired effect of dropout is to enforce the model to not depend on specific units of every layer.

The model described in Figure 3 is based on Enhanced Deep-HITS Reyes et al. (2018), a state-of-the-art classifier for binary classification of real astronomical objects and bogus samples. The architecture of this model introduced total

rotational invariance, which empirically proved to enhance performance on the classification of astronomical images.

### B.2. Neural Network Training

1. *Procedure to train a neural network:* The objective of using a neural network  $f_\theta$  of parameters  $\theta \in \Theta$ , is to approximate a function  $y = f(x)$ , with  $x \in \mathcal{X}$ . In practice, there is no access to the whole data distribution  $\mathcal{X}$ , but to a subset of  $N$  data samples of the function to approximate  $\{(x^{(i)}, y^{(i)})\}_{i=1}^N$ , called *training set*. Finding the best parameters  $\theta$  for the neural network  $f_\theta(x)$  requires solving the optimization problem:

$$\theta^* = \arg \min_{\theta \in \Theta} \mathcal{C}(\theta) = \arg \min_{\theta \in \Theta} \frac{1}{N} \sum_{i=1}^N \mathcal{L}(y^{(i)}, f_\theta(x^{(i)})), \quad (\text{B11})$$

where  $\mathcal{C}$  is an error functional defined by the function  $\mathcal{L}$  that is called a loss function. The optimization depicted in Equation (B11), is achieved by the training of the model through optimization techniques based on gradient descent, when  $\mathcal{L}$  is chosen as a differentiable function (e.g., cross entropy). The parameters  $\theta$  are iteratively adjusted by the following rule, until convergence:

$$\theta_k = \theta_{k-1} - \mu \nabla_{\theta} \mathcal{C}(\theta). \quad (\text{B12})$$

Because neural networks are composed of many consecutive layers, the direct calculation of  $\nabla_{\theta} \mathcal{C}(\theta)$  is computationally expensive. However, they can be calculated efficiently by back-propagation, which is an algorithm that *propagates* the error from the output of the model until it reaches the first layer of the neural network, back-propagation is based on the differentiation chain rule. Even when using back-propagation, for neural networks trained on large amounts of data, the calculation of the exact gradient  $\nabla_{\theta} \mathcal{C}(\theta)$  becomes computationally expensive. As a

solution to this problem, a nonbiased estimation of the gradient  $\nabla_{\theta} \tilde{\mathcal{C}}(\theta)$  is used, where the gradient is calculated over a small random fraction of data, this is called a batch, and the number of data samples in the batch is the batch size. Therefore, the optimization rule for a batch  $\mathcal{B} \subset \mathcal{X}$  is:

$$\theta_k = \theta_{k-1} - \mu \frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \nabla_{\theta} \mathcal{L}(y^{(i)}, f_{\theta}(x^{(i)})), \quad (\text{B13})$$

where  $\mu$  is a constant called learning rate, and establishes how large is the performed training step. This technique of training by batches is a form of *stochastic gradient descent*, and it guarantees convergence when  $\mu$  is a well-defined sequence in  $k$  that satisfies  $\sum_k \mu_k = \infty$  and  $\sum_k \mu_k^2 < \infty$ .

2. *Adam optimizer:* An alternative to the optimization rule of Equation (B13), is Adam Kingma & Ba (2017), which is an adaptive learning rate optimization algorithm that automatically adjusts  $\mu_k$ . It uses the squared gradients to scale the learning rate and it includes the moving average of the gradients in its formulation, a strategy that is known as momentum, and it is used to avoid convergence to a local minima in the optimization. The main hyperparameters of Adam are  $\beta_1$  and  $\beta_2$ , which relate to the moving averages of the gradients and the squared gradients, respectively, and they regulate the rate at which the learning rate  $\mu_k$  is adjusted.

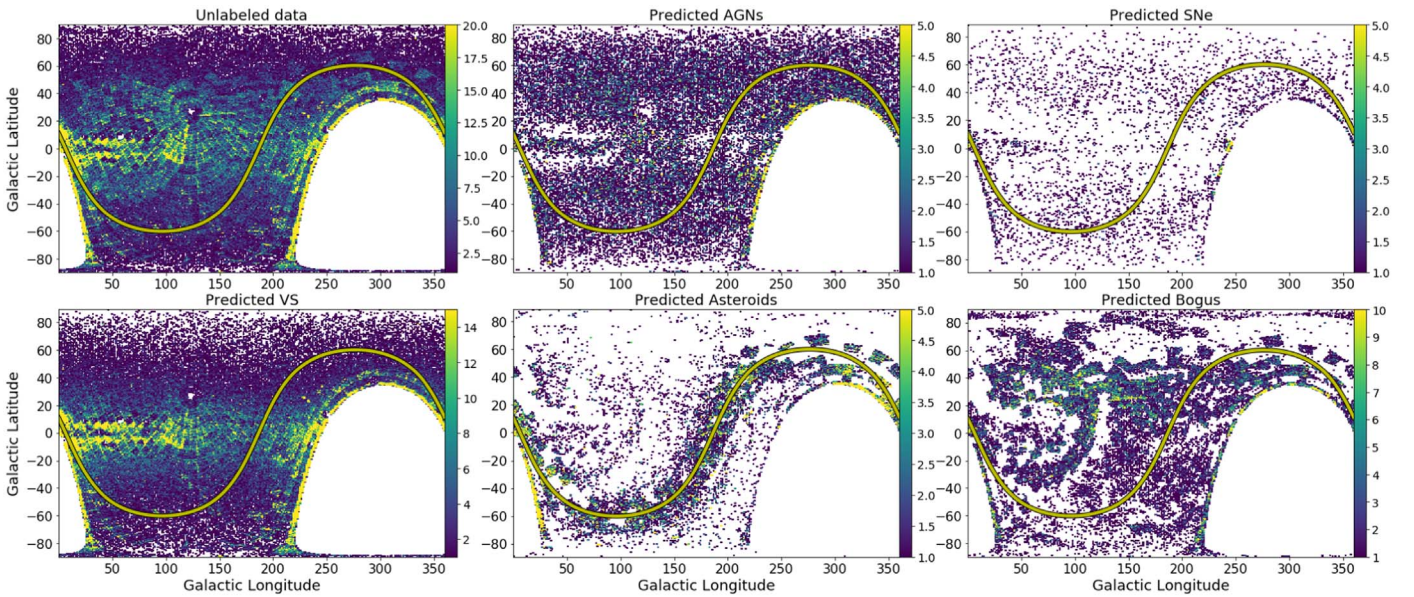
## Appendix C Additional Results

For completeness, we also report the confusion matrix of the stamp classifier when no metadata features are included in the fully connected layers (see Figure C1), which has a test set accuracy of  $0.883 \pm 0.006$ , recovering  $80\% \pm 2\%$  of the SNe in the test set, with  $10\% \pm 4\%$  of false positives. In addition, we show in Figure C2, the distribution for the unlabeled data classified by the CNN without including features.

Test set

AGN	0.91±0.02	0.02±0.01	0.06±0.01	0	0.02±0.01
SN	0.03±0.01	0.8±0.02	0.01±0.01	0.14±0.01	0.02±0.01
VS	0.14±0.02	0	0.85±0.02	0	0
asteroid	0	0.04±0.02	0.01±0.0	0.95±0.02	0
bogus	0	0.04±0.01	0.01±0.0	0.04±0.02	0.91±0.02
True label	AGN	SN	VS	asteroid	bogus
	Predicted label				

**Figure C1.** Confusion matrix for the test set when using the stamp classifier only on the three images, without alert metadata features.

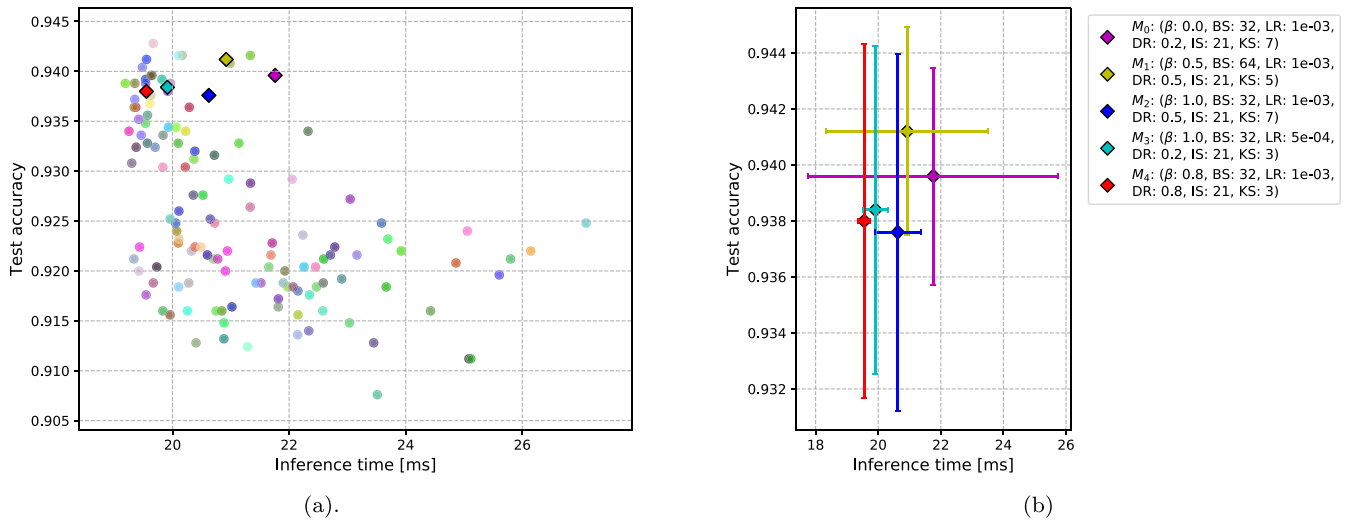


**Figure C2.** Space distribution for the unlabeled data, and distribution of predictions per class using the stamp classifier only on the three images, without alert metadata features. The ecliptic is shown with a yellow line with black edges. Extragalactic classes (SNe and AGNs) still show a lower density at low galactic latitudes but are found with higher density compared to the predictions of the stamp classifier with features, shown in Figure 8. Also, the predicted asteroids by the classifier without using alert metadata features have higher density far from the ecliptic compared to the classifier that uses features.

## Appendix D Hyperparameter Random Search Results

For the hyperparameter random search, 133 different combinations of hyperparameters values were sampled from Table 3. We trained five models for each hyperparameter set and then evaluated the test and validation set with every model. In addition, for each model the inference time over a single sample was measured. The training procedure took  $\approx 3$  days of continuous training on five NVIDIA GTX 1080Ti GPUs. From now on, every time we refer to the accuracy or inference time of a model, they are the average measurement of five models trained with the same hyperparameters.

The selection of the best model from the hyperparameter random search was done by looking at the performance in the validation set. We took the five models with the highest validation accuracy and performed a Welch's hypothesis test between the model with the highest and lowest validation accuracy among the top-5 models, obtaining a  $p$ -value of 0.594, which means that the accuracy differences among the models are not statistically significant. Another important factor when processing the volume of data encountered in astronomy is the inference time of the used models, we measure inference time for the top-5 models and a Welch's hypothesis test between the fastest and the slowest model got a  $p$ -value of 0.330, i.e., no statistically meaningful difference.



**Figure D1.** Accuracy of 133 models from hyperparameter random search. For each model, results consider five trainings and respective evaluations on the test set. (a) Test accuracy versus inference time, where each dot is a model with different hyperparameters, the closer to the left top corner is a model, the better its performance. Models represented with diamonds correspond to the five models with the best validation accuracy. (b) Test accuracy versus inference time for the models represented as a diamond in (a), each model has its error bars corresponding to one standard deviation, and every model is denoted as  $M_i$ , where  $i$  corresponds to the ranking of its validation accuracy performance among all the 133 models of (a).

**Table D1**  
Top-5 Models with Highest Validation Accuracy from the Hyperparameter Random Search, Ranked from  $M_0$  to  $M_4$

Model Name	Model's Hyperparameters	Validation Accuracy	Test Accuracy	Inference Time (ms)
$M_0$	$\beta: 0$ , BS: 32, LR: 1e-03, DR: 0.2, IS: 21, KS: 7	<b><math>0.950 \pm 0.003</math></b>	$0.940 \pm 0.004$	$21.8 \pm 4.0$
$M_1$	$\beta: 0.5$ , BS: 64, LR: 1e-03, DR: 0.5, IS: 21, KS: 5	<u><math>0.950 \pm 0.005</math></u>	<b><u><math>0.941 \pm 0.004</math></u></b>	<u><math>20.9 \pm 2.6</math></u>
$M_2$	$\beta: 1.0$ , BS: 32, LR: 1e-03, DR: 0.5, IS: 21, KS: 7	$0.949 \pm 0.002$	$0.938 \pm 0.006$	$20.6 \pm 0.7$
$M_3$	$\beta: 1.0$ , BS: 32, LR: 5e-04, DR: 0.2, IS: 21, KS: 3	$0.948 \pm 0.003$	$0.938 \pm 0.006$	$19.9 \pm 0.4$
$M_4$	$\beta: 0.8$ , BS: 32, LR: 1e-03, DR: 0.8, IS: 21, KS: 3	$0.949 \pm 0.003$	$0.938 \pm 0.006$	<b><math>19.6 \pm 0.2</math></b>
Welch's t-test $p$ -value $M_0$ versus $M_4 - M_1$ versus $M_4 - M_0$ versus $M_4$		0.594	0.364	0.330

**Note.** There is no statistical difference between the accuracy and inference time of the displayed models.  $M_1$  is chosen as the best model because it has  $\beta = 0.5$ , which shows the most interpretable range of prediction probabilities, according to astronomers. Metrics of the best model  $M_1$  are underlined, whereas bold metrics are the best of their respective column.

**Table D2**

Classes Assigned to the Unlabeled Set Described in Section 4, by the Light Curve Classifier (LC) from Sánchez-Sáez et al. (2021) and the Stamp Classifier (SC)

SC/LC	SN Ia	SN Ibc	SN II	SLSN	AGN	Blazar	QSO	CV/ Nova	YSO	DSCT	RRL	Ceph	LPV	EB	Periodic- Other
SN	355	124	246	257	657	87	13	241	76	1	22	4	58	7	18
AGN	5	2	27	83	4057	1310	9553	592	309	227	1586	67	61	1133	1738
VS	10	7	22	38	393	623	691	2545	5023	5098	29635	9657	36478	56587	16539
Asteroid	19	8	6	17	1	4	1	38	10	0	7	1	107	8	1
Bogus	7	1	9	19	47	16	13	84	36	1	8	4	23	14	22
Recall	90%	87%	79%	62%	79%	64%	93%	73%	92%	96%	95%	99%	99%	98%	90%

**Note.** The stamp classifier finds 78% of the SN classified by the light curve classifier, 85% for AGNs, and 96% for VS. The false positives in the SN class are 9% of AGNs, 6% of VS, 4% of asteroids, and 3% of bogus alerts. The false positives of AGNs are 4% of SN and 1% of VS, and the false positive of VS is only 3% with AGNs.

Finally, the best model among the top-5 is chosen as the one with  $\beta = 0.5$ , because it shows the most interpretable range of prediction probabilities, according to our team of astronomers. The model chosen as the best has a validation accuracy of  $0.950 \pm 0.003$ , test accuracy of  $0.941 \pm 0.004$ , and inference time of  $20.5 \pm 2.6$  [ms]. The performance of the top-5 models over the test set is shown both in Figure D1 and Table D1. Every time we refer to the top-5 models we mean the aforementioned models.

Figure D1 shows plots of test accuracy versus the inference time. Figure D1(a) shows the results of all the 133 models from the random hyperparameter search: In Figure D1(a), the nearest to the top left corner, the better the model, the diamond shapes in this figure correspond to the models with top-5 validation accuracy. Figure D1(b) shows in detail the performance of these top-5, where each model has its one standard deviation error bar. These models are named  $M_i$ , where  $i$  corresponds to the position of its validation accuracy with respect to all the

133 models, the lower its validation accuracy, the higher is  $i$ . Table D1 shows validation, test accuracy, and inference time for the models with top-5 validation accuracy of Figure D1(b), where  $M_1$  is selected as the best model, which is used for experiments shown in previous sections. In Table D1, metrics of the best model  $M_1$  are underlined, whereas bold metrics correspond to the highest of their respective column. Coincidentally, model  $M_1$  chosen as the best has the higher test accuracy among the top-5 models, and when compared to model  $M_2$  with the worst test accuracy, Welch’s hypothesis test  $p$ -value is 0.364, meaning that the difference is not statically meaningful.

Figure D1(b) shows that test accuracy and inference time error bars of each of the top-5 models contain each other.

### Appendix E Bogus Alert Analysis

We elaborate on an analysis of both the bogus samples from ZTF’s alert stream and the ones present in our training set.

With our model, we estimate the proportion of bogus events present in ZTF’s alert stream, a proportion that is unfeasible to accurately calculate via direct visualization due to the large number of alerts generated each night. We run 176,376 alerts chosen at random from ZTF’s alert stream on 10 nights through the stamp classifier; each of these nights are also chosen at random between 2020 August 18 and 2021 March 3. Our model classifies 34,438 alerts as bogus ( $\sim 20\%$  of the alerts).

As the stamp classifier used by the SN Hunter only processes first detection alerts, to estimate the number of bogus alerts processed in a night from the previous 176,376 alerts we only take the 51,481 alerts that have 1 detection. Our model identifies 31,001 samples as bogus ( $\sim 60\%$  of the total 51,481 alerts with 1 detection).

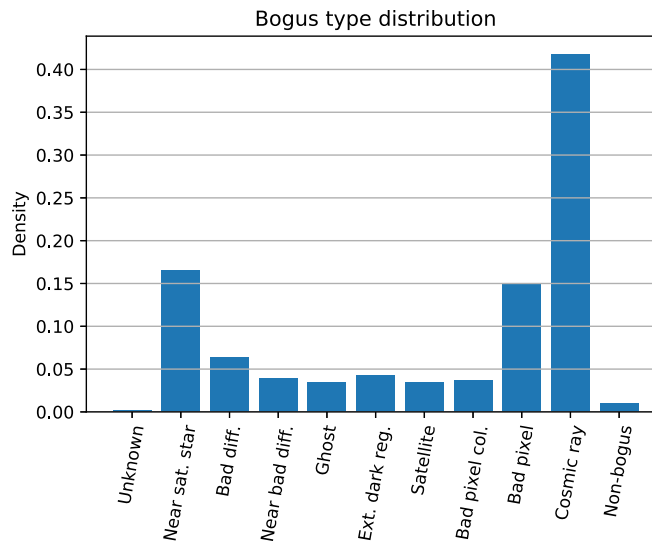
Based on the manual observations of alerts classified as bogus by our experts, we were able to identify nine different types of bogus alerts that most commonly appear in ZTF’s stream:

1. *Region near saturated star*: Bright sources systematically found around very bright targets. When stars are too

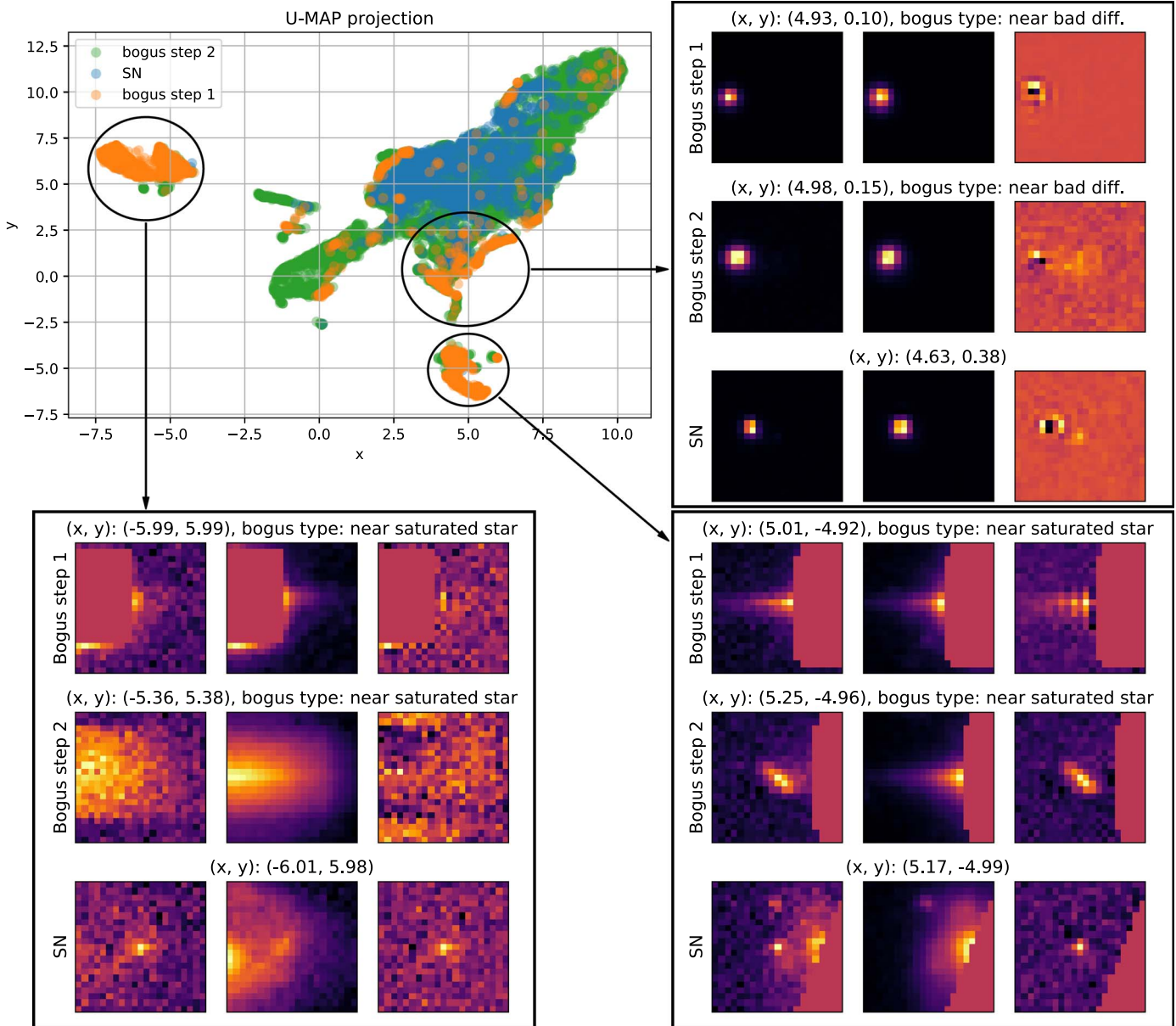
bright, they can saturate the camera of the telescope and produce artifacts that appear as light flux variations and trigger alerts.

2. *Bad difference*: Produced by a misalignment of the bright source in the template and science images, which translates into artifacts in the difference image. They appear in the images as dark/light dipoles around every source in the field.
3. *Near bad difference*: A bad difference can produce artifacts that affect wide portions of an image and trigger alerts far away from the misaligned source.
4. *Ghost*: Residuals from saturated observations. They usually look like large extended circles with diffraction spikes.
5. *Extended dark region*: Extended dark regions in the science image.
6. *Satellite*: We consider alerts triggered by passing human-made satellites as bogus. They trigger alerts that often show multiple point-like or extended sources on the same image in a line (due to rotation and reflection of the satellite) and appear convolved with the PSF.
7. *Bad pixel columns*: Science images captured by regions of the telescope’s camera where whole columns of pixels go bad. They look like a clear change in the background (up or down) over a small portion of columns in the field of the images.
8. *Bad pixel*: Science images captured by regions of the telescope’s camera where a single pixel is bad.
9. *Cosmic ray*: High energy particles that interact with a few pixels of the telescope’s camera. They generate alerts that look smaller than or have shapes distinctly different from a point source or moving source convolved with PSF.

We elaborate on a recognition of the previous types of bogus alerts present in our training set. A subset of 1000 bogus samples randomly taken from the training set was analyzed by an astronomer, where all the nine different types of bogus events described above were identified. A distribution of the 1000 identified bogus events can be seen in Figure E1. Although we did not characterize types of bogus events for all



**Figure E1.** Normalized distribution of different types of bogus events present in 1000 bogus samples from our training set. Bogus types were visually analyzed by an astronomer.



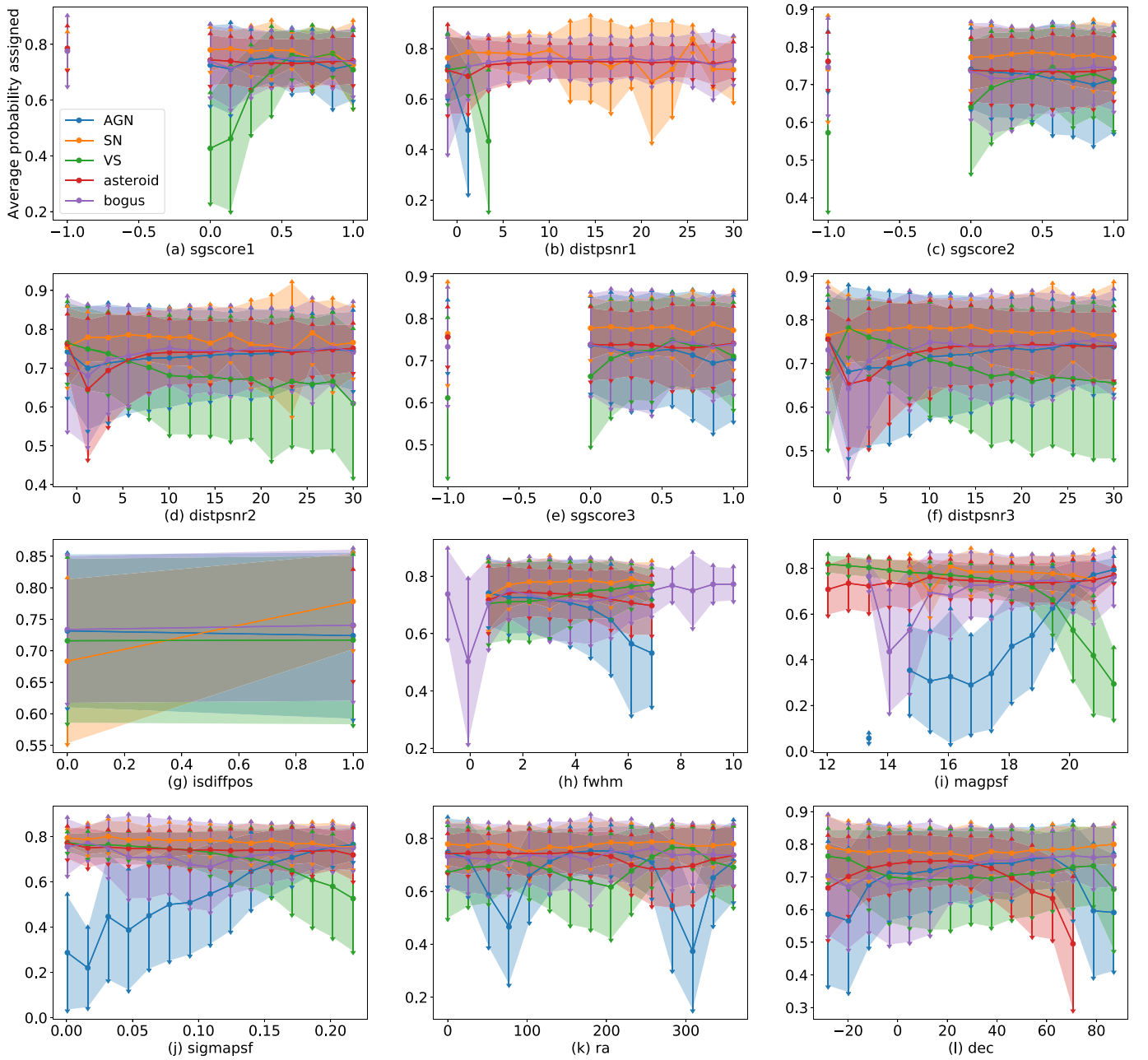
**Figure E2.** U-MAP projection of image triplets of bogus alerts labeled in step 1 and step 2, alongside SNe alerts. Three main clusters can be identified in the projection. For each of them, images of samples from each class are shown, and their projection coordinates are displayed to denote that they are close in the U-MAP projection space; close-by samples tend to look alike. Bogus types for each sample are assigned by an astronomer. Each triplet shows images of science, reference, and difference, in that order. The analyzed regions are enclosed by circles labeled from 1 to 3.

the bogus alerts in the training set (because of time restrictions involved in manually analyzing  $\sim 10,000$  bogus events), as the 1000 bogus samples from Figure E1 were randomly sampled, it is an approximation of the true distribution of bogus types in the whole training set.

Figure E1 shows that the most common type of bogus events in our training set are cosmic rays. This can be explained by how most of the bogus samples were obtained. Bogus events misclassified as SN by the stamp classifier, and cosmic rays, are the most common type of bogus events misclassified as SN. Figure E1 also shows that 10 samples were identified as non-bogus (alerts triggered by a real astronomical source), which can be interpreted as the fact that our manual bogus event labeling process has an estimated error/contamination of 1%. One of the bogus events analyzed does not match any known type, so it is assigned the *unknown* tag in Figure E1.

The previous analysis was performed over the bogus events in our training set. We propose as future work to perform a similar study over alerts from the ZTF’s stream that are classified as bogus by our stamp classifier. It is worth mentioning that the distribution of bogus events types shown in Figure E1 is not representative of the actual distribution from ZTF’s alert stream, since it is biased by our labeling process.

As bogus events in the training set were manually labeled in two steps, we analyzed how the distribution of bogus events varies from one step to the other. Step 1 is composed of 1980 bogus examples reported by ZTF (based on human inspection). Many bogus events coming from step 1 are characterized for containing many NaN patches, being near saturated sources or subtraction misalignments—bogus types that are easily identifiable by eye. These bogus events were used to train an early version of the stamp classifier and detect



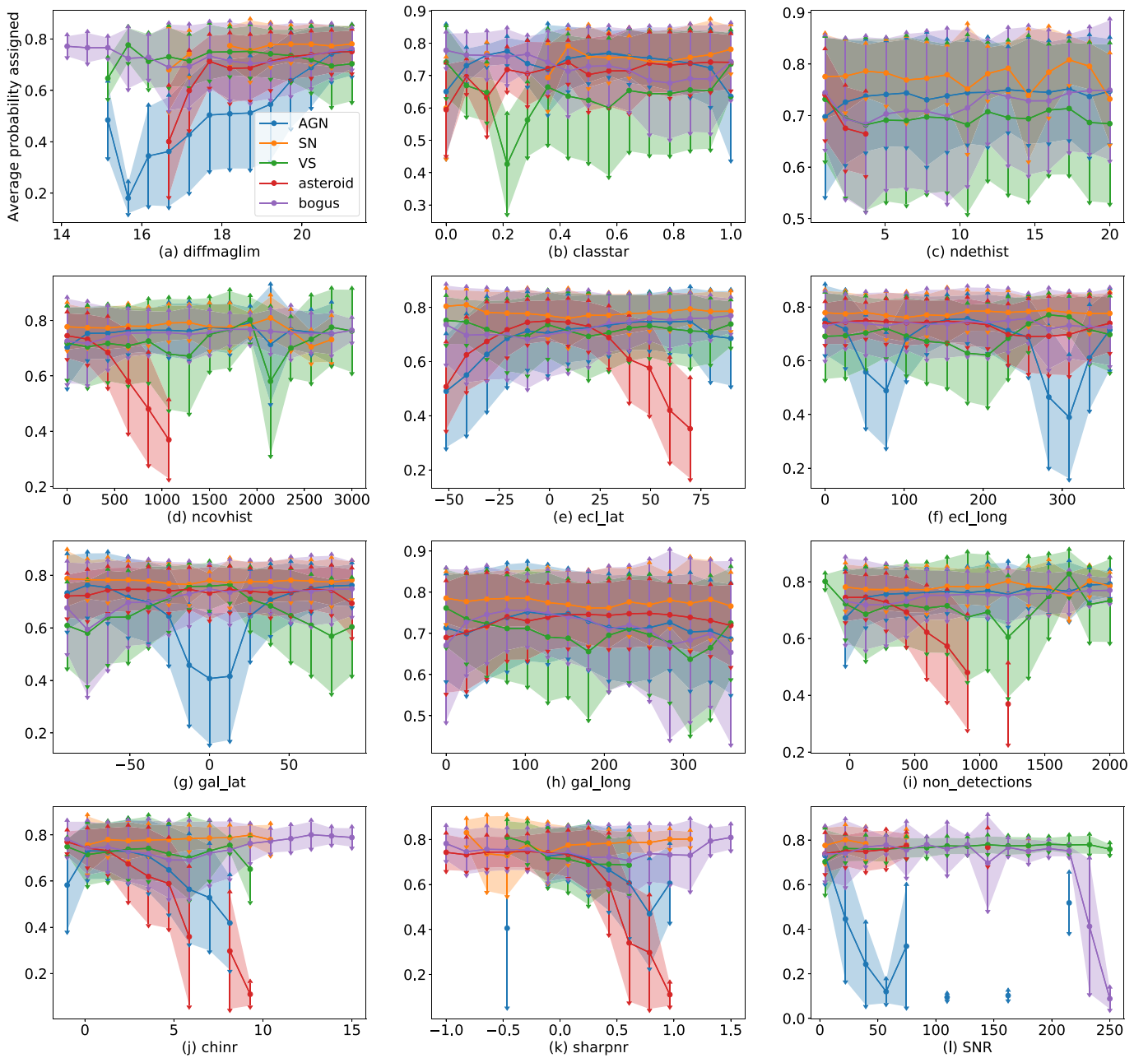
**Figure E3.** Average model probabilities assigned to the correct class in the training set versus feature values. Each plot from (a) to (l) contains the probabilities for a specific feature.

SNe. Step 2 bogus examples include 8783 samples and they are alerts that were confused with a SN by the early version of the stamp classifier, and astronomers visually confirmed them as bogus. As we increased the amount of labeled bogus alerts, we iteratively improved the stamp classifier and kept visually identifying bogus alerts that were misclassified as SNe. Because of the way we obtained our bogus samples, we expect them not to be representative or have the same distribution of all the bogus events in ZTF’s alert stream. To avoid these type of biases, in the future we plan to label by hand bogus alerts that are confused with the rest of the classes (VS, AGN, and asteroids).

The effect of using bogus events coming from these two steps can be seen in Figure E2, where we used U-MAP

(McInnes et al. 2020) to project alert image triplets in a 2D space; alerts with similar images should appear as neighbor points in the projection. Alongside each cluster of alerts, we plotted images of bogus events and SNe samples to visualize the similarity between SNe and step 2 bogus alerts. Figure E2 shows three main clusters, two small clusters at the right and bottom of the U-MAP projection, composed mainly by step 1 bogus alerts, and a big cluster mainly composed of SNe alerts and step 2 bogus alerts, with a few step 1 bogus alerts in it, the ones that most resemble SNe. The big cluster shows that SNe samples overlap with step 2 bogus alerts, where the latter fill spaces between and around SNe. The behavior of step 2 bogus alerts with respect to SNe is no surprise to us. Given the way step 2 bogus alerts were generated (from misclassified SNe by





**Figure E4.** Average model probabilities assigned to the correct class in the training set versus feature values. Each plot from (a) to (l) contains the probabilities for a specific feature. Note that the signal-to-noise ratio was not added as a feature to the classifier—this is shown for explanatory purposes mentioned in Section 4.

old versions of the stamp classifier), they are expected to resemble or look like SNe.

For all the classes, we computed the probability assigned to alerts of each class in the training set as a function of the specific value of each of the features given to the classifier; these probabilities are shown in (Figures E3 and E4).

By visual inspection of the clusters, we verified that close-by samples in the U-MAP projection do look alike and have similar geometric structures. For the three main clusters of Figure E2, we analyzed the regions enclosed by dark circles (labeled from 1 to 3 in the figure) and visualized the samples of SNe and bogus events of steps 1 and 2:

1. Circle 1 encloses the small cluster at the left, which is dominated by samples with NaN patches or bright

sources at the left of the images. The displayed samples correspond to a near saturated star type bogus alert with a NaN patch at its left, for step 1 bogus events. The step 2 bogus sample is also a near saturated star type with the bright source at its left. The displayed SN sample has a bright source at the left of the template image (middle image).

















2. Circle 2 encloses a region of the biggest cluster. This region is the one with the most step 1 bogus events, and it is mainly composed of samples with a bad difference at the left or top of the images. The displayed samples of type step 1 and step 2 bogus events are near bad differences located at the left of the images, while the SN sample also has a bad difference at its left.

3. Circle 3 encloses the small cluster at the bottom of Figure E2, which is mainly composed of samples with NaN patches or bright sources at the right of the images. The displayed step 1 and step 2 bogus samples correspond to near saturated star type bogus events with NaN patch at their right, while the SN sample has a bright source and a NaN patch at its right.

For all the regions analyzed, clustering was dominated by geometric compositions within images. Additionally, bogus clusters can be related to the different bogus types previously described, where bogus alerts within a region tend to be of the same type. In the future it would be useful to use a visualization technique invariant to the geometric orientation of samples, to avoid cases where clusters with bogus samples of the same type but different geometric orientations are positioned far away in the projection.

We do not show all five classes together in the U-MAP projection because trying to project so many classes of high-dimensional data in a 2D embedding is a problem that is too hard to solve for U-MAP and not enough insightful clusters arise—only blobs of highly overlapped samples can be seen.

### ORCID iDs

R. Carrasco-Davis  <https://orcid.org/0000-0003-4673-8791>  
 E. Reyes  <https://orcid.org/0000-0003-3455-9358>  
 C. Valenzuela  <https://orcid.org/0000-0001-5306-1390>  
 F. Förster  <https://orcid.org/0000-0003-3459-2270>  
 P. A. Estévez  <https://orcid.org/0000-0001-9164-4722>  
 G. Pignata  <https://orcid.org/0000-0003-0006-0188>  
 F. E. Bauer  <https://orcid.org/0000-0002-8686-8737>  
 I. Reyes  <https://orcid.org/0000-0003-3627-0216>  
 P. Sánchez-Sáez  <https://orcid.org/0000-0003-0820-4692>  
 G. Cabrera-Vives  <https://orcid.org/0000-0002-2720-7218>  
 S. Eyheramendy  <https://orcid.org/0000-0003-4723-9660>  
 M. Catelan  <https://orcid.org/0000-0001-6003-8877>  
 D. Ruz-Mieres  <https://orcid.org/0000-0002-9455-157X>  
 A. Moya  <https://orcid.org/0000-0002-7003-5087>  
 A. A. Mahabal  <https://orcid.org/0000-0003-2242-0244>  
 L. Galbany  <https://orcid.org/0000-0002-1296-6887>

### References

- Astropy Collaboration, Robitaille, T. P., Tollerud, E. J., et al. 2013, *A&A*, 558, A33
- Bach, S., Binder, A., Montavon, G., et al. 2015, *PLoS*, 10, 1
- Barchi, P. H., de Carvalho, R. R., Rosa, R. R., et al. 2020, *Astronomy and Computing*, 30, 100334
- Becker, I., Pichara, K., Catelan, M., et al. 2020, *MNRAS*, 493, 2981
- Bellm, E. C., Kulkarni, S. R., Graham, M. J., et al. 2018, *PASP*, 131, 018002
- Bertin, E., & Arnouts, S. 1996, *A&AS*, 117, 393
- Boch, T., & Fernique, P. 2014, *Astronomical Data Analysis Software and Systems XXIII*, 485, 277
- Bonnarel, F., Fernique, P., Bienaymé, O., et al. 2000, *A&AS*, 143, 33
- Boone, K. 2019, *AJ*, 158, 257
- Brahem, M., Zeitouni, K., & Yeh, L. 2018, *IEEE Transactions on Big Data*, 6, 477
- Breiman, L. 2001, *Machine Learning*, 45, 5
- Cabrera-Vives, G., Reyes, I., Förster, F., Estévez, P. A., & Maureira, J.-C. 2017, *ApJ*, 836, 97
- Carrasco-Davis, R., Cabrera-Vives, G., Förster, F., et al. 2019, *PASP*, 131, 108006
- Chambers, K. C., Magnier, E. A., Metcalfe, N., et al. 2019, arXiv:1612.05560 [astro-ph]
- Chollet, F., et al. 2018, Keras: The Python Deep Learning Library, Astrophysics Source Code Library, ascl:1806.022
- Dieleman, S., De Fauw, J., & Kavukcuoglu, K. 2016, arXiv:1602.02660
- Dieleman, S., Willett, K. W., & Dambre, J. 2015, *MNRAS*, 450, 1441
- Drake, A. J., Djorgovski, S. G., Catelan, M., et al. 2017, *MNRAS*, 469, 3688
- Drake, A. J., Graham, M. J., Djorgovski, S. G., et al. 2014, *ApJS*, 213, 9
- Duev, D. A., Mahabal, A., Masci, F. J., et al. 2019, *MNRAS*, 489, 3582
- Flesch, E. W. 2015, *PASA*, 32, e010
- Flesch, E. W. 2019, arXiv:1912.05614[astro-ph]
- Förster, F., Cabrera-Vives, G., Castillo-Navarrete, E., et al. 2021, *AJ*, 161, 242
- Förster, F., Moriya, T. J., Maureira, J. C., et al. 2018, *NatAs*, 2, 808
- Gal-Yam, A., Arcavi, I., Ofek, E. O., et al. 2014, *Natur*, 509, 471
- Goldstein, D. A., D'Andrea, C. B., Fischer, J. A., et al. 2015, *AJ*, 150, 82
- Gómez, C., Neira, M., Hoyos, M. H., Arbeláez, P., & Forero-Romero, J. E. 2020, *MNRAS*, 499, 3130
- Groh, J. H. 2014, *A&A*, 572, L11
- Hunter, J. D. 2007, *CSE*, 9, 90
- Jayasinghe, T., Kochanek, C. S., Stanek, K. Z., et al. 2018, *MNRAS*, 477, 3145
- Jayasinghe, T., Stanek, K. Z., Kochanek, C. S., et al. 2019a, *MNRAS*, 486, 1907
- Jayasinghe, T., Stanek, K. Z., Kochanek, C. S., et al. 2019b, *MNRAS*, 485, 961
- Jayasinghe, T., Stanek, K. Z., Kochanek, C. S., et al. 2020, *MNRAS*, 491, 13
- Jiang, J.-A., Doi, M., Maeda, K., et al. 2017, *Natur*, 550, 80
- Ivezić, Ž., Kahn, S. M., Tyson, J. A., et al. 2019, *ApJ*, 873, 111
- Kasen, D. 2010, *ApJ*, 708, 1025
- Khazov, D., Yaron, O., Gal-Yam, A., et al. 2016, *ApJ*, 818, 3
- Kingma, D. P., & Ba, J. 2017, arXiv:1412.6980[cs]
- Mahabal, A., Rebbapragada, U., Walters, R., et al. 2019, *PASP*, 131, 038002
- Mahabal, A., Sheth, K., Gieseke, F., et al. 2017, in 2017 IEEE Symp. Series on Computational Intelligence (SSCI) (Piscataway, NJ : IEEE), 1
- Martín Abadi, A., Agarwal, A., Barham, P., et al. 2015, *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*, <https://www.tensorflow.org/>
- Martínez-Palomera, J., Förster, F., Protopapas, P., et al. 2018, *AJ*, 156, 186
- Masci, F. J., Laher, R. R., Rusholme, B., et al. 2018, *PASP*, 131, 018003
- Massaro, E., Maselli, A., Leto, C., et al. 2015, *Ap&SS*, 357, 75
- McInnes, L., Healy, J., & Melville, J. 2020, UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction, arXiv:1802.03426
- McKinney, W. 2010, in Proc. 9th Python in Science Conf., Austin, Texas June 28, 2010, 56
- Montavon, G., Binder, A., Lapuschkin, S., Samek, W., & Müller, K.-R. 2019, in *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*, ed. W. Samek et al. (Cham: Springer), 193
- Moriya, T., Tominaga, N., Blinnikov, S. I., Baklanov, P. V., & Sorokina, E. I. 2011, *MNRAS*, 415, 199
- Morozova, V., Piro, A. L., & Valenti, S. 2017, *ApJ*, 838, 28
- Mowlavi, N., Lecoœur-Taïbi, I., Lebzelter, T., et al. 2018, *A&A*, 618, A58
- Muthukrishna, D., Parkinson, D., & Tucker, B. E. 2019, *ApJ*, 885, 85
- Nair, V., & Hinton, G. E. 2010, Proc. 27th Int. Conf. on Int. Conf. on Machine Learning, ICML'10 (Haifa: Omnipress), 807
- Narayan, G., Zaidi, T., Soraisam, M. D., et al. 2018, *ApJS*, 236, 9
- Naul, B., Bloom, J. S., Pérez, F., & van der Walt, S. 2018, *NatAs*, 2, 151
- Noebauer, U. M., Kromer, M., Taubenberger, S., et al. 2017, *MNRAS*, 472, 2787
- Nordin, J., Brinell, V., Santen, J. v., et al. 2019, *A&A*, 631, A147
- Nugent, P. E., Sullivan, M., Cenko, S. B., et al. 2011, *Natur*, 480, 344
- Oh, K., Yi, S. K., Schawinski, K., et al. 2015, *ApJS*, 219, 1
- Palaversa, L., Ivezić, Ž., Eyer, L., et al. 2013, *AJ*, 146, 101
- Pedregosa, F., Varoquaux, G., Gramfort, A., et al. 2011, *Journal of Machine Learning Research*, 12, 2825
- Pereyra, G., Tucker, G., Chorowski, J., Kaiser, L., & Hinton, G. 2017, arXiv:1701.06548[cs]
- Pérez-Carrasco, M., Cabrera-Vives, G., Martínez-Marin, M., et al. 2019, *PASP*, 131, 108002
- Pichara, K., Protopapas, P., & León, D. 2016, *ApJ*, 819, 18
- Piro, A. L., & Morozova, V. S. 2016, *ApJ*, 826, 96
- Piro, A. L., & Nakar, E. 2013, *ApJ*, 769, 67
- Reyes, E., Estévez, P. A., Reyes, I., et al. 2018, in 2018 Int. Joint Conf. on Neural Networks (IJCNN), Rio de Janeiro, Brazil, 8-13 July 2018, (Piscataway, NJ: IEEE), 1
- Richards, J. W., Starr, D. L., Butler, N. R., et al. 2011, *ApJ*, 733, 10
- Rimoldini, L., Holl, B., Audard, M., et al. 2019, *A&A*, 625, A97
- Rocklin, M. 2015, in Proc. 14th Python in Science Conf., Austin, Texas, July 11, 2015, 126
- Sánchez-Sáez, P., Reyes, I., Valenzuela, C., et al. 2021, *AJ*, 161, 141
- Smith, K. W., Williams, R. D., Young, D. R., et al. 2019, *RNAAS*, 3, 26
- Soumagnac, M. T., & Ofek, E. O. 2018, *PASP*, 130, 075002
- Steer, I., Madore, B. F., Mazzarella, J. M., et al. 2016, *AJ*, 153, 37

- Stetson, P. B. 1987, [PASP](#), **99**, 191
- Tanaka, M., Tominaga, N., Morokuma, T., et al. 2016, [ApJ](#), **819**, 5
- Tominaga, N., Morokuma, T., Blinnikov, S. I., et al. 2011, [ApJS](#), **193**, 20
- Tonry, J. L., Denneau, L., Heinze, A. N., et al. 2018, [PASP](#), **130**, 064505
- Turpin, D., Ganet, M., Antier, S., et al. 2020, [MNRAS](#), **497**, 2641
- van Velzen, S., Gezari, S., Hammerstein, E., et al. 2021, [ApJ](#), **908**, 26
- Wenger, M., Ochsenbein, F., Egret, D., et al. 2000, [A&AS](#), **143**, 9
- Yaron, O., Perley, D. A., Gal-Yam, A., et al. 2017, [NatPh](#), **13**, 510
- Zaharia, M., Xin, R. S., Wendell, P., et al. 2016, [Communications of the ACM](#), **59**, 56
- Zeiler, M. D., & Fergus, R. 2014, in *Computer Vision ECCV 2014*, ed. D. Fleet et al. (Cham: Springer), 818