



SNIascore: Deep-learning Classification of Low-resolution Supernova Spectra

Christoffer Fremling¹, Xander J. Hall¹, Michael W. Coughlin², Aishwarya S. Dahiwalé¹, Dmitry A. Duev¹, Matthew J. Graham¹, Mansi M. Kasliwal¹, Erik C. Kool³, Ashish A. Mahabal^{1,4}, Adam A. Miller^{5,6}, James D. Neill¹, Daniel A. Perley⁷, Mickael Rigault⁸, Philippe Rosnet⁹, Ben Rusholme¹⁰, Yashvi Sharma¹, Kyung Min Shin¹, David L. Shupe¹⁰, Jesper Sollerman³, Richard S. Walters¹, and S. R. Kulkarni¹

¹ Division of Physics, Mathematics, and Astronomy, California Institute of Technology, Pasadena, CA 91125, USA; fremling@caltech.edu

² School of Physics and Astronomy, University of Minnesota, Minneapolis, MN 55455, USA

³ The Oskar Klein Centre, Department of Astronomy, Stockholm University, AlbaNova, SE-10691 Stockholm, Sweden

⁴ Center for Data Driven Discovery, California Institute of Technology, Pasadena, CA 91125, USA

⁵ Center for Interdisciplinary Exploration and Research in Astrophysics and Department of Physics and Astronomy, Northwestern University, 1800 Sherman Ave., Evanston, IL 60201, USA

⁶ The Adler Planetarium, Chicago, IL 60605, USA

⁷ Astrophysics Research Institute, Liverpool John Moores University, Liverpool Science Park, 146 Brownlow Hill, Liverpool L35RF, UK

⁸ Univ Lyon, Univ Claude Bernard Lyon 1, CNRS, IP2I Lyon/IN2P3, IMR 5822, F-69622, Villeurbanne, France

⁹ Université Clermont Auvergne, CNRS/IN2P3, Laboratoire de Physique de Clermont, F-63000 Clermont-Ferrand, France

¹⁰ IPAC, California Institute of Technology, 1200 E. California Blvd., Pasadena, CA 91125, USA

Received 2021 April 27; revised 2021 July 3; accepted 2021 July 6; published 2021 August 5

Abstract

We present *SNIascore*, a deep-learning-based method for spectroscopic classification of thermonuclear supernovae (SNe Ia) based on very low-resolution ($R \sim 100$) data. The goal of *SNIascore* is the fully automated classification of SNe Ia with a very low false-positive rate (FPR) so that human intervention can be greatly reduced in large-scale SN classification efforts, such as that undertaken by the public Zwicky Transient Facility (ZTF) Bright Transient Survey (BTS). We utilize a recurrent neural network architecture with a combination of bidirectional long short-term memory and gated recurrent unit layers. *SNIascore* achieves a $<0.6\%$ FPR while classifying up to 90% of the low-resolution SN Ia spectra obtained by the BTS. *SNIascore* simultaneously performs binary classification and predicts the redshifts of secure SNe Ia via regression (with a typical uncertainty of <0.005 in the range from $z = 0.01$ to $z = 0.12$). For the magnitude-limited ZTF BTS survey ($\approx 70\%$ SNe Ia), deploying *SNIascore* reduces the amount of spectra in need of human classification or confirmation by $\approx 60\%$. Furthermore, *SNIascore* allows SN Ia classifications to be automatically announced in real time to the public immediately following a finished observation during the night.

Unified Astronomy Thesaurus concepts: [Supernovae \(1668\)](#); [Type Ia supernovae \(1728\)](#); [Neural networks \(1933\)](#); [Classification \(1907\)](#); [Surveys \(1671\)](#)

1. Introduction

Modern time-domain surveys, such as the Zwicky Transient Facility (ZTF; Bellm et al. 2019a, 2019b; Graham et al. 2019; Masci et al. 2019; Dekany et al. 2020), the All-Sky Automated Survey for Supernovae (ASAS-SN; Shappee et al. 2014) and the Asteroid Terrestrial Last-Alert System (ATLAS; Tonry et al. 2018b), are now finding tens of thousands of transients every year.

However, without spectroscopic classifications, these discoveries are of limited value (Kulkarni 2020). The ZTF Bright Transient Survey (BTS; Fremling et al. 2020; Perley et al. 2020) is addressing this through the deployment of a fully automated very-low-resolution spectrograph, the Spectral Energy Distribution Machine (SEDM; Blagorodnova et al. 2018; Rigault et al. 2019) mounted on the Palomar 60 inch telescope. SEDM is capable of obtaining spectra of several thousands of transients per year in the magnitude range between 18 and 19 mag. Currently, the goal of the BTS is to maintain spectroscopic classification completeness for all extragalactic transients detected by the public ZTF survey that become brighter than 18.5 mag (~ 1000 SNe per year; Perley et al. 2020).

The classifications from the BTS are made public on a daily basis via the Transient Name Server (TNS¹¹). These classifications have up until now been based on manual matching of

observed spectra to spectral templates using mainly the SuperNova IDentification (SNID; Blondin & Tonry 2007) code, along with careful inspection of each obtained spectrum. This makes classification of thousands of SNe a very time-consuming endeavor.

Due to their inherent brightness, the majority of the extragalactic transients discovered by a magnitude-limited survey will be thermonuclear supernovae¹² (SNe Ia). Here we present *SNIascore*, a deep-learning-based method optimized to identify SNe Ia using SEDM spectra and determine their redshifts without any human interaction. The intended use case for *SNIascore* is to provide live spectroscopic classification of SNe Ia during the night when SEDM is observing for the BTS.

SNIascore is based on a recurrent neural network (RNN) architecture (see Sherstinsky 2020 for a review) with a combination of bidirectional long short-term memory (BiLSTM) and gated recurrent unit (GRU) layers. *SNIascore* is able to classify $>80\%$ of the SN Ia spectra that are observed by SEDM for the ZTF BTS with a false-positive rate (FPR) of $<1\%$.

SNIascore was trained on SEDM data obtained by the ZTF BTS between 2018 and 2020 March, and validated and optimized with the BTS data set published by Fremling et al. (2020). A final

¹¹ <https://wis-tns.org>

¹² In the ZTF BTS $\approx 72.5\%$ of the extragalactic transients detected are SNe Ia (Perley et al. 2020).

Table 1
Data Summary

Class	All	Training	Validation	Testing
SNIa	2619	1526	607	486
NotSNIa	2931	1997	409	525
H-rich CC SN	1285	751	312	222
H-poor CC SN	585	393	94	98
TDE	37	35	0	2
CV	325	284	0	41
Other	699	534	3	162

Note. All numbers refer to the number of unique spectra in each class. The “Other” class includes any spectra that do not fit the other classes, including galaxy spectra, active galactic nucleus (AGN) spectra, and spectra that BTS has been unable to classify as belonging to any known transient class.

test run is performed using data obtained between 2020 April and 2020 August, neither of which were part of the training or validation data sets. Our data sets are described in detail in Section 2. The network architecture is described in Section 3, and the training and optimization procedure used for SNIAscore is described in Section 4. The performance is evaluated and compared to SNID and the previously published deep-learning method DASH (Muthukrishna et al. 2019b) in Section 5. A discussion on the most likely false positives can be found in Section 6. The implementation of SNIAscore as part of the ZTF BTS is described in Section 7. Future development possibilities are also discussed in this section.

2. Data Sets

Our data consist of SEDM spectra of transients detected by ZTF, which were followed up and classified by the BTS (Fremling et al. 2020; Perley et al. 2020) using the GROWTH Marshal (Kasliwal et al. 2019). These spectra are of low resolution ($R \sim 100$) and cover a typical wavelength range of 3800–9150 Å within 209 wavelength bins (see Section 2.2). Although the BTS focuses on extragalactic transients, some spectra eventually turn out to be of Galactic sources (e.g., cataclysmic variables; CVs). We include both spectra of Galactic and extragalactic transients in our data sets. The data set we use here contains 5550 SEDM spectra of 3463 individual transients obtained between 2018 March and 2020 August. A breakdown of the various classes of transients included in our data set can be found in Table 1. We split our full data set into three parts: training data, validation and optimization data, and final performance testing data.

For the purpose of training SNIAscore (Section 4) we group the data into two classes: real SNe Ia (SNIa; 1526 spectra of 1090 SNe Ia) and everything else (NotSNIa; 1997 spectra of 1121 transients). The training data were collected between 2018 March 7 and 2020 March 1, but we exclude any data that are part of our validation set, which was also collected during 2018.

To validate and optimize SNIAscore (Section 4) we use 1016 spectra of 632 SNe, which are part of the BTS sample from 2018 published in Fremling et al. (2020; hereafter the BTS18 sample). The BTS18 sample was chosen for validation because the classifications and redshifts of the SNe in this sample have been carefully vetted by humans.

For final performance testing we use all BTS spectra collected between 2020 March 2 and 2020 August 5

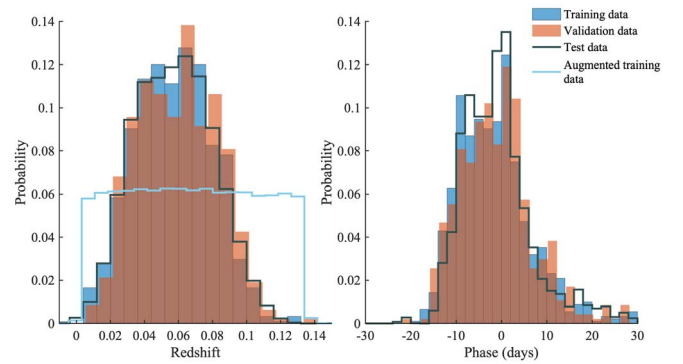


Figure 1. Redshift (left) and spectral phase (right) distributions for SNe Ia in the unaugmented SNIAscore training (blue) and validation (red) data sets used to optimize SNIAscore for classification. The SN Ia distributions for the testing data set are shown in black. The blue line in the left panel shows the redshift distribution of the training set after augmentation (Section 2.1), which is used to train SNIAscore for redshift regression. The phase is relative to the time of maximum light in the g or r band, depending on which band is brighter.

(1011 spectra of 632 transients). The classifications in this testing sample are sufficiently accurate to evaluate the performance of SNIAscore, but they are subject to minor future changes as work on the BTS proceeds.¹³

For training SNIAscore for redshift prediction, we only use spectra of SNe Ia and also impose a cut on the quality of the known redshifts. We only use SNe Ia where the redshift is known to three decimal places or more.¹⁴ Both redshifts derived from broad SN features and host-galaxy emission lines are used. The training set for redshift prediction consists of 891 spectra of 630 SNe Ia, which is increased to 12,810 spectra through data augmentation (Section 2.1).

The properties of the training, validation, and testing data sets are illustrated in Figure 1. The redshift range covered by our full data set is $z = 0.008$ to $z = 0.126$. However, 98% of our data fall within the range $z = 0.01$ to $z = 0.11$. The phases of our spectra with respect to maximum light determined from the lightcurves of our transients fall within -20 days to $+30$ days, with 97% of the data within -20 days to $+20$ days. Thus, the range we can expect SNIAscore to perform reliably within is $z = 0.01$ to $z = 0.11$ for spectra obtained within ± 20 days of maximum brightness. We use the time of maximum light from the BTS Sample Explorer (Perley et al. 2020), which records the brightest actual measurement in the lightcurve of each transient. This measurement can be in the g or r band, depending on which band is brighter.

2.1. Training Set Augmentation

For the classification component of SNIAscore we can achieve excellent performance without any need for data augmentation (Section 5). We also do not need to de-redshift our spectra as part of the preprocessing procedure that is performed before training or when classifying new data (Section 2.2). However, in order to use regression to predict the redshifts of SNe Ia identified by SNIAscore without

¹³ The live BTS sample, which is updated daily, can be accessed through the BTS sample explorer at <https://sites.astro.caltech.edu/ztf/bts/explorer.php>.

¹⁴ This typically means that a spectrum of the host galaxy exists in the literature (e.g., on NED), the SEDM spectrum was of very high signal-to-noise ratio (S/N) with very good SNID template matches or clear host-galaxy lines, or that we have obtained a spectrum of the SN from a telescope that offers a higher resolution compared to SEDM.

introducing systematic bias, a weighting scheme, or data augmentation, is needed. This is due to the shape of the redshift distribution of our data set (Figure 1); there are very few spectra for SNe Ia at both $z < 0.03$ and $z > 0.09$.

We have found that a simple augmentation procedure, which equalizes the redshift distribution of the training sample, can offer excellent redshift prediction performance with negligible bias. We perform the following steps: for each redshift bin of size 0.001 from zero to 0.13 we search for spectra in our unaugmented data set with similar redshifts (within $z - 0.005$ to $z + 0.01$). We then randomly pick spectra from the matches and apply new redshifts by shifting the wavelengths of the original spectra to randomly selected redshifts within ± 0.001 of the redshift bin center (the spectra are not de-redshifted to the rest frame as part of this process to reduce loss of information at the edges of the spectra). We repeat this process until we have roughly 100 spectra in each redshift bin, for a total of 12810 spectra. To reduce overfitting and lessen the impact of repeating a small number of spectra at the edges of the redshift distribution, we introduce noise from a normal distribution to each added spectrum.¹⁵ The final redshift distribution for our training data used for redshift regression is shown in Figure 1.

2.2. Preprocessing of Spectra

Due to small variations in the final wavelength coverage of the SEDM spectra included in our data set, we interpolate all spectra to a uniform wavelength grid, which spans the largest possible wavelength range without a need for extrapolation for any individual spectrum. This range is 3800–9150 Å with 209 wavelength bins of size 25.6 Å.

After interpolation we normalize each spectrum by division with its median value. The normalized spectrum is then divided by its continuum (a heavily smoothed version of the normalized spectrum computed with robust¹⁶ local polynomial regression using weighted linear least squares and a second-degree polynomial model). Finally, to center the preprocessed spectra around zero, we subtract a constant value of one from each normalized and continuum divided spectrum. This process is illustrated in Figure 2.

We have found that this preprocessing procedure significantly outperforms a simple division by the median, directly dividing by a smoothed continuum, or dividing by the median and then subtracting the smoothed continuum. We have also investigated the effect of suppressing the bluest and reddest parts of the spectra (which can be very noisy in data from many instruments), as in Muthukrishna et al. (2019b). This decreases the overall performance and is not needed for SEDM data.

Effectively our preprocessing procedure flattens the spectra (removes any temperature gradient or host-galaxy continuum emission contribution), makes emission lines positive and absorption lines negative, and normalizes the strengths of the features.

3. Neural Network Architecture

For our final SNIAscore network that we have arrived at after following the optimization procedure described in

¹⁵ We add between zero and one standard deviation of noise. The standard deviation is estimated as the standard deviation of each spectrum after subtraction of the broad features using a heavily smoothed spectrum.

¹⁶ Lower weights are assigned to outliers in the regression; outliers at $> 6\sigma$ are assigned zero weight.

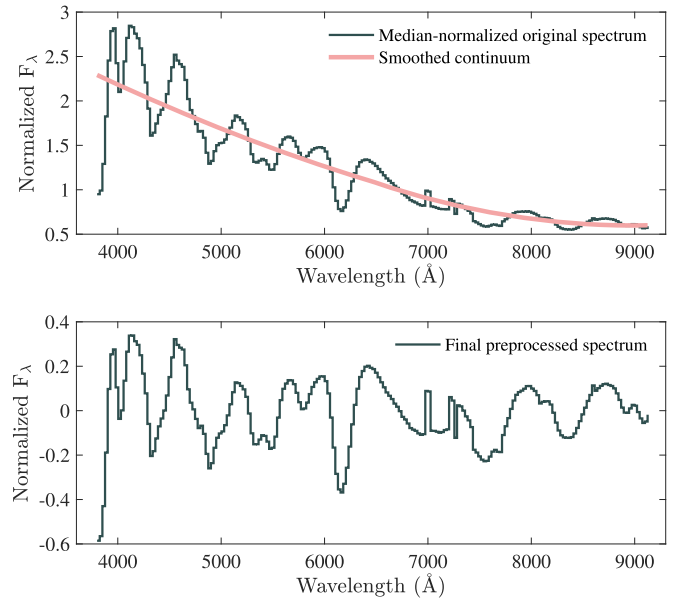


Figure 2. Preprocessing procedure used in SNIAscore. The top panel shows an example SN Ia spectrum, normalized by the median value (black line). This normalized spectrum is divided by a smoothed continuum (computed with robust local polynomial regression; red line) to create the final preprocessed spectrum (black line, bottom panel), which is centered around zero by subtracting a constant value of 1.

Section 4, we have used an RNN architecture (see e.g., Sherstinsky 2020) consisting of a combination of BiLSTM and GRU layers, with 32 hidden units in each layer (Figure 3). We employ significant dropout both during training and prediction. We use a dropout of 40% immediately following the input layer. This is followed by a BiLSTM layer and 45% dropout. After this we use a GRU layer and 35% dropout and then another BiLSTM layer and 25% dropout. This last BiLSTM layer is followed by a fully connected layer with two outputs, a softmax layer and a classification layer that computes the cross-entropy loss for the SNIa and NotSNIa classes.

For redshift regression the softmax and classification layers are replaced by a custom regression layer, which uses the root of the mean bias error (MBE; Equation (A2)) squared as the forward loss function and the derivative of the mean absolute error (MAE; Equation (A4)) as the backward loss function. We have found that this custom regression layer gives the least systematic bias and best overall performance among the loss functions we have tested (Section 4).

To estimate uncertainties when predicting both the SNIAscore ($\sigma_{\text{SNIAscore}}$) and redshift (σ_z), we use a simple Monte Carlo (MC) method. Because dropout is used during prediction, the results of subsequent predictions based on the same input spectrum will be different. We rerun each prediction 100 times and adopt the standard deviation of the results as the uncertainty of the prediction (Gal & Ghahramani 2015).

4. Training and Optimization

4.1. Classification Optimization

We have implemented the architecture for SNIAscore described in Section 3 using the MATLAB Deep Learning ToolboxTM. For training we use the adaptive learning rate optimization algorithm Adam (Kingma & Ba 2014) with the following settings:

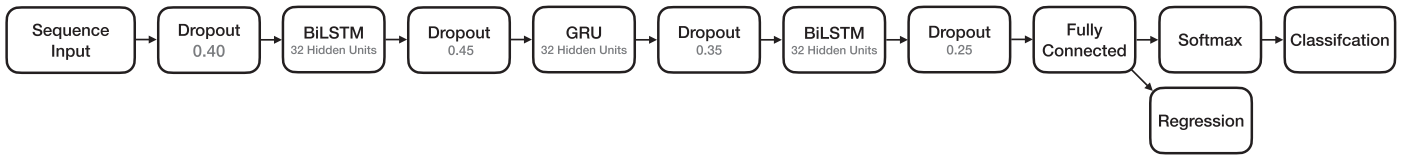


Figure 3. Network architecture of `SNIascore`. We utilize heavy dropout throughout the network and a combination of two BiLSTM layers surrounding one GRU layer. For regression the final softmax and classification layers are replaced by a regression layer.

1. MiniBatchSize: 256
2. InitialLearnRate: 0.005
3. LearnRateSchedule: none
4. L2Regularization: 10^{-4}
5. GradientThreshold: 2
6. Shuffle: every-epoch
7. SequenceLength: longest.

For validation and optimization we use the BTS18 data set. To arrive at the architecture described in Section 3 and the optimal hyperparameters listed above, we have followed this procedure:

1. We construct a grid of networks with an initial dropout layer following the input layer and then one to five BiLSTM layers with dropout layers between each layer.
2. For each network we create a grid of subnetworks with a range of hyperparameters, dropout from 0.15 to 0.45 for each dropout layer, and hidden units from 8 to 128 for each BiLSTM layer.
3. We explore learning rates in the range 0.001–0.01 and mini batch sizes of 2^n in the range 16–512 for each network.

This optimization procedure showed that a constant learning rate of 0.005 and a mini batch size of 256 seems to perform universally well on our data, regardless of the other hyperparameters. Changes to the learning rate (including introducing adaptive learning rate schemes), gradient threshold, and L2 regularization listed above have negligible impact on the final performance. As long as the network retains the ability to converge during training, it remains possible to pick an optimal epoch from the training sequence where the performance is very similar.

We do not try to optimize for the required physical training time or number of epochs required to achieve a good performance. We do not employ any stopping condition based on the loss. Instead, each network is trained well past the point of overfitting (at least 325 epochs), and we save the state of each network at every training epoch. We then choose and compare the optimal epoch (typically found between epochs 200 and 300) for every individual network based on the performance on the BTS18 validation set, according to a combination of two metrics: (i) the total number of successful classifications for an FPR of 0%, and (ii) the total number of successful classifications for an FPR of 1%. To determine the optimal FPR and true-positive rate (TPR) of an individual network, all possible cuts on `SNIascore` and $\sigma_{\text{SNIascore}}$ are evaluated. These metrics direct the optimization procedure toward hyperparameters and a network structure that results in a low FPR, with a high TPR being secondary. We consider $\text{FPR} < 1\%$ to be a hard requirement for autonomous spectroscopic classification. It also allows all the trained networks to be quantitatively compared based on their optimal epochs and `SNIascore` cuts.

The architecture that resulted from this optimization procedure was 3 BiLSTM layers with 32 hidden units in each

layer, in combination with heavy dropout (0.40, 0.45, 0.35, and 0.25 for the 4 dropout layers from first to last). As a final step, we investigated the effect of replacing the BiLSTM layers with GRU layers in all possible combinations and eventually arrived at our final `SNIascore` network described in Section 3, where the middle BiLSTM layer is replaced by a GRU layer. The effect of this is a reduced number of false positives with high `SNIascore` values.

For our optimized classification network, Figure 4 shows the `SNIascore` distribution for known SNe Ia and other transients in the BTS18 validation sample. We find that 90% of the SNe Ia in this sample are identified by `SNIascore` with a 0.6% FPR. The performance of our final `SNIascore` network is investigated in detail and tested on independent data in Section 5.

4.2. Redshift Regression Optimization

For redshift regression we use the same optimal `SNIascore` architecture found above with the softmax and classification layers replaced with a regression layer (Figure 3). However, we have found that the standard deep-learning mean square error (MSE; Equation (A1)) or MAE loss functions result in significant redshift-dependent systematic bias, even after data augmentation to equalize the redshift distribution of the training data (Section 2.1). Because of this, we perform an additional optimization step for the redshift regression network, which consists of an investigation of various forward and backward loss functions in the final regression layer of the network.

The forward loss functions that we have investigated are the MAE, the mean percentage error (MPE; Equation (A3)), the MSE, the MBE, and all possible combinations of these in pairs, with both the derivative of the MAE and MSE as the backward loss.

The MPE, MBE, and the root square of these can all give good results. Differences are minor, but we have selected the root square bias error ($\sqrt{\text{MBE}^2}$) as the forward loss for our final network and the derivative of the MAE as the backward loss, based on optimizing (i) the percentage of SNe with redshift residuals less than 0.005 and 0.01 for the BTS18 validation sample, and (ii) how close the predictions on the full validation set adhere to a linear relationship with respect to the known redshifts ($y = ax + b$ where a should be as close to 1 as possible and b as close to zero as possible). Each network was trained past overfitting (at least 525 epochs) and the optimal epoch (typically found between epochs 400 and 500) was selected after the fact for comparison among the networks.

5. Performance

In order to evaluate the classification performance of `SNIascore` we have performed comparisons to SNID (Blondin & Tonry 2007) and DASH (Muthukrishna et al. 2019b) (Section 5.1). We also evaluate the accuracy of the

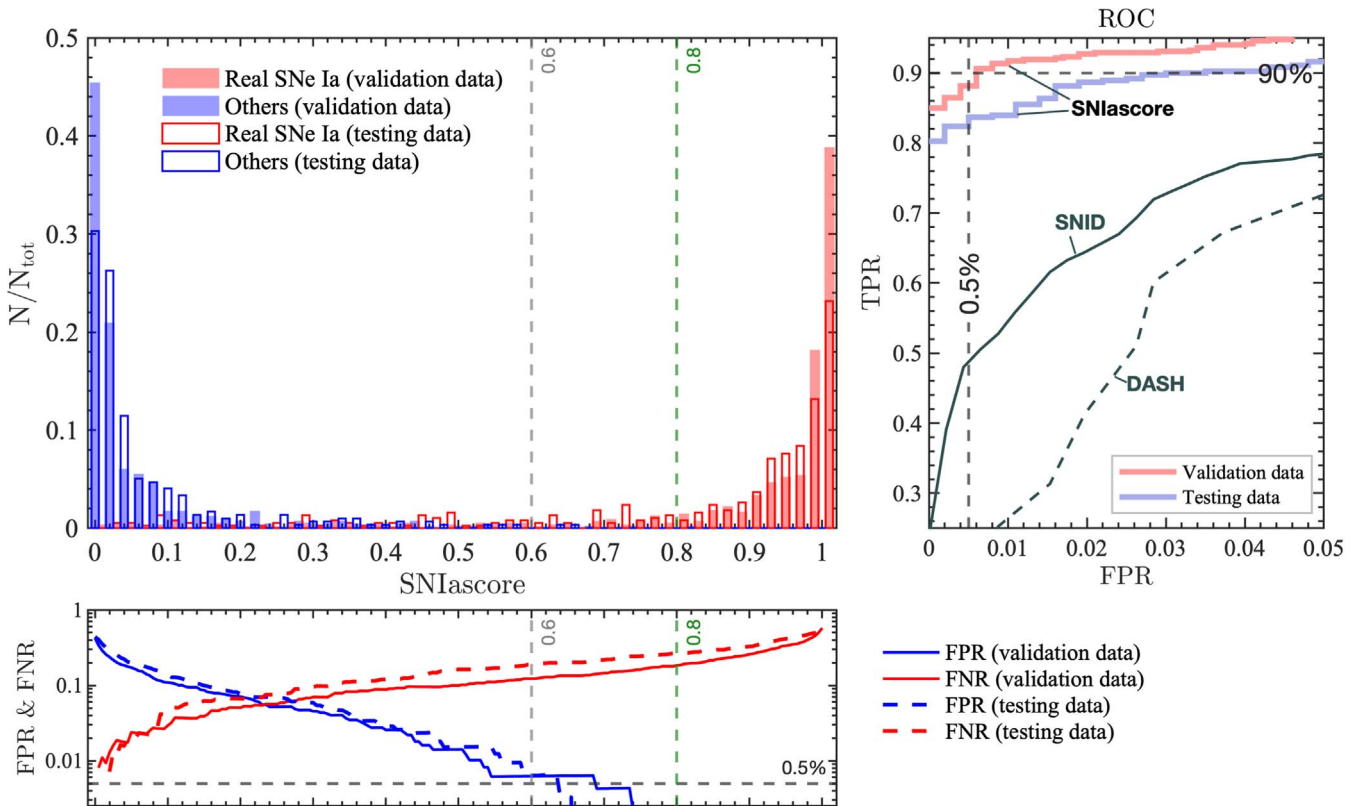


Figure 4. Classification performance of SNIAscore on the BTS18 validation sample and on the testing data set. Top-left panel: the SNIAscore distributions for both “Real SNe Ia” spectra (red) and spectra of “Other” transients (blue). The validation set is shown as solid bars, and the testing data set is shown as open bars. N_{tot} is the total for each class separately (i.e., the sum of the “Real SNe Ia” bins add up to unity). Bottom panel: FPR and FNR as a function of SNIAscore for the validation (solid red and blue lines) and for the testing (dashed red and blue lines) data sets. A cut on SNIAscore at 0.6 results in an FPR of roughly 0.5% and a TPR of 90% on the validation data set and 83% on the testing data set. A cut at an SNIAscore of 0.8 results in an FPR of zero on both the validation and testing data sets, with a TPR of 86% and 80%, respectively. Right panel: ROC curve for SNIAscore on the validation data set (red line) and the testing data set (blue line). For comparison we also show ROC curves for SNID (solid black line) and DASH (dashed black line).

associated redshift predictions for the SNe Ia that are identified by SNIAscore by comparing to host-galaxy redshifts and redshifts determined after manual inspection with SNID (Section 5.2).

5.1. Classifications

Our goal with SNIAscore is a low-enough FPR ($<1\%$) so that the classifications do not require any human confirmation. To achieve this, using the standard value of $\text{SNIAscore} > 0.5$ to classify a spectrum as that of an SN Ia is not adequate. Instead, we evaluate all possible cuts on SNIAscore and $\sigma_{\text{SNIAscore}}$ to find a combination of cuts that result in the desired performance.

Receiver operating characteristic (ROC) curves for SNIAscore (computed by varying SNIAscore and $\sigma_{\text{SNIAscore}}$), SNID (by varying the rlap parameter threshold), and DASH (by varying rlap and the softmax score threshold) are shown in Figure 4. Based on this comparison it is clear that SNIAscore significantly outperforms these alternatives¹⁷. Between 83% (testing data) and 90% (validation data) of the SN Ia spectra that SEDM observes for the ZTF BTS can be classified with $\text{FPR} \approx 0.5\%$. This performance is achieved for the cuts $\text{SNIAscore} > 0.6$ and $\sigma_{\text{SNIAscore}} < 0.3$, which

produce F1-scores of 0.95 and 0.91 for the validation and testing data sets, respectively.

SNID can achieve a very low FPR for SN Ia spectra when $\text{rlap} > 10$. However, only $\approx 50\%$ of the spectra could potentially be automatically classified ($\text{FPR} < 1\%$) without supervision with SNID. The deep-learning-based DASH only reaches a low-enough FPR to enable automatic classification when applied to SEDM spectra for 24% of spectra for any cuts on rlap and the softmax score. This is likely due to the fact that no spectra with similarly low resolution to those produced by SEDM were part of the DASH training data.¹⁸

The cuts on SNIAscore and $\sigma_{\text{SNIAscore}}$ mentioned above result in the highest TPR while keeping the FPR within the range we consider acceptable ($<1\%$). However, these cuts do result in some false positives, which we discuss in Section 6. More conservative cuts at $\text{SNIAscore} > 0.8$ and $\sigma_{\text{SNIAscore}} < 0.275$ result in zero false positives on the validation and training data, with F1-scores of 0.91 and 0.86 for the validation and testing data sets, respectively. Depending on the use case, either choice of these cuts may be suitable. For live reporting of classifications to the community via TNS, the stricter cuts seem

¹⁷ Details on how SNID and DASH were used to produce the corresponding ROC curves can be found in the appendix (Section B).

¹⁸ We have investigated similar convolutional neural networks as used in DASH (Muthukrishna et al. 2019b). When trained on SEDM data, similar performance to our RNN approach can be achieved when cutting at the standard $\text{SNIAscore} > 0.5$. However, we were unable to eliminate false positives in the SNIAscore range of 0.8–1 to the same extent as for our optimal RNN.

Table 2Performance Comparison of SNI_{ascore}, SNID, and DASH on the BTS18 Validation Data Set

Method	TPR	FPR	Cuts ^a
SNI _{ascore} TPR	0.90	0.006	score > 0.6, $\sigma < 0.3$
SNI _{ascore} FPR	0.86	0.000	score > 0.8, $\sigma < 0.275$
SNID	0.53	0.009	rlap > 11.1
DASH	0.24	0.009	rlap > 7.1, score > 0.5

Notes. For SNI_{ascore} TPR, SNID, and DASH we show the optimal TPR that can be had while maintaining FPR < 1%. For SNI_{ascore} FPR, we show the optimal TPR that can be had for FPR = 0.

^a score refers to the softmax score for both SNI_{ascore} and DASH. For SNI_{ascore}, σ refers to $\sigma_{\text{SNI}_{\text{ascore}}}$.

appropriate (Section 7). We summarize our recommended SNI_{ascore} cuts and the performance compared to SNID and DASH in Table 2. An investigation of how the classification performance of SNI_{ascore} varies depending on data quality can be found in Appendix C.

Finally, we note that a low SNI_{ascore} does not necessarily mean that the transient being observed cannot be an SN Ia. There are multiple reasons for low scores, for example, low S/N, strong galaxy light contamination, and significant contamination from cosmic rays that failed to be removed during data reduction. As such, a score that is lower than the cuts discussed above should generally be interpreted as that the spectrum cannot be used to make a confident SN Ia classification, and nothing further. Only when the SNI_{ascore} is extremely low (<0.01), is there significant statistical power to support a conclusion that the transient associated with the respective spectrum cannot be an SN Ia. We have made no attempt at optimizing the FNR for qlow scores, but based on the FNR curves for the validation and testing data sets (Figure 4), SNI_{ascore} < 0.01 corresponds to FNR \approx 1%.

5.2. Redshifts

Because only SN Ia spectra were used during training for redshift prediction (Section 4.2), we only perform predictions for spectra that pass the stricter cuts that optimize SNI_{ascore} for low FPR (Table 2). Any false positives will receive inaccurate redshift predictions, so they need to be avoided.

To evaluate the accuracy of the redshifts predicted by SNI_{ascore}, we compare predictions for the BTS18 validation and testing data sets primarily to redshifts derived from host-galaxy spectra, and secondarily to redshifts determined from broad SN features through manual template matching using SNID which were performed for the BTS18 SEDM data in Fremling et al. (2020). The performance of the SNI_{ascore} redshift regression network is illustrated in Figure 5.

We find that the performance of our network optimized on the BTS18 validation data carries over very well to the testing data set; we find no evidence for systematic bias in the predictions from SNI_{ascore} for either data set. The absolute value of the difference between each prediction compared to the redshift from the host-galaxy spectrum is <0.005 for 77% of the spectra in the validation set and for 85% of the spectra in the testing data set. For 96% (validation data) and 100% (testing data), this difference is <0.01.

We find the mean of the redshift residuals, $\Delta z = -0.0008$, and the standard deviation, $\sigma_z = 0.0046$, for the BTS18

validation data when SNI_{ascore} is compared to redshifts from host-galaxy spectra. For the testing data we find $\Delta z = 0.0006$ and $\sigma_z = 0.0035$, when compared to redshifts from host-galaxy spectra. When SNI_{ascore} is compared to redshifts derived manually from broad SN features using SNID we find $\Delta z = -0.0002$ and $\sigma_z = 0.0049$ (only possible for the validation data). It has been previously found that $\sigma_z = 0.005$ when manual usage of SNID is compared to redshifts from host-galaxy spectra on the BTS18 sample (Fremling et al. 2020), which is consistent with what was found for higher-resolution spectra by Blondin & Tonry (2007). As such, we conclude that the performance of SNI_{ascore} is consistent with what can be done with SNID when each result and spectrum is manually inspected. A detailed investigation of the accuracy of automatic SNID SN Ia redshifts from SEDM is being performed for data up until 2021 (M. Rigault et al. 2021, in preparation).

6. Discussion

A few spectra in both our validation and testing data sets end up as false positives with the optimal TPR cut we suggest to be used with SNI_{ascore} (Table 2). Most of these spectra are of low quality or affected by strong host contamination (and uninteresting). However, there are also a few high-quality spectra. It is interesting to investigate these and the associated SNe in some detail.

In the testing set, ZTF20aatxryt (SN 2020eyj; Tonry et al. 2020) stands out. Our first SEDM spectrum, taken around maximum light of the SN on 2020 April 2, is of high quality and looks consistent with an SN Ia spectrum and gets an SNI_{ascore} of 0.7 ± 0.3 . However, it later turned out that the ejecta from this SN crashed into circumstellar material, resulting in a flattening of its lightcurve and changing its spectrum into that of an SN Ibn (E. Kool et al. 2021, in preparation). The best matches in SNID are also SNe Ia with rlap > 10. However, by manually going through all the matches produced by SNID at rlap > 5, some SN Ic matches can also be found. Given the lightcurve evolution and the fact that SN Ia lightcurve models do not match well even around maximum light before the interaction sets in, it may be more likely that this was a stripped-envelope SN. We note that the ZTF BTS officially classified this as an SN Ia as part of our routine operations based on the SNID match to the peak spectrum (Dahiwalé & Fremling 2020). Only later did we realize that the object was unusual.

In the validation set, ZTF18aaxmhvk (SN 2018cne; Tonry et al. 2018a) stands out (SNI_{ascore} = 0.71 ± 0.17). Our SEDM spectrum of this source, taken around maximum light on 2018 June 14, is of high quality and is also best matched to SNe Ia in SNID with very high rlap values. However, the lightcurve of ZTF18aaxmhvk is not consistent with that of a normal SN Ia. Based on this, and the fact that matches to SNe Ic can also be found in SNID, the BTS officially classified this as an SN Ic (Fremling & Sharma 2018). However, it could also have been a peculiar SN Ia. The SEDM spectrum is the only spectrum available for this source; we lack higher-resolution spectra and do not have any late-time spectra, which could have provided a conclusive answer.

In conclusion, the main contaminants that may produce high SNI_{ascore} values appear to be stripped-envelope SNe, and in particular events with SN Ic-like spectra. The BTS has been classifying roughly 30 SNe Ic per year (Perley et al. 2020).

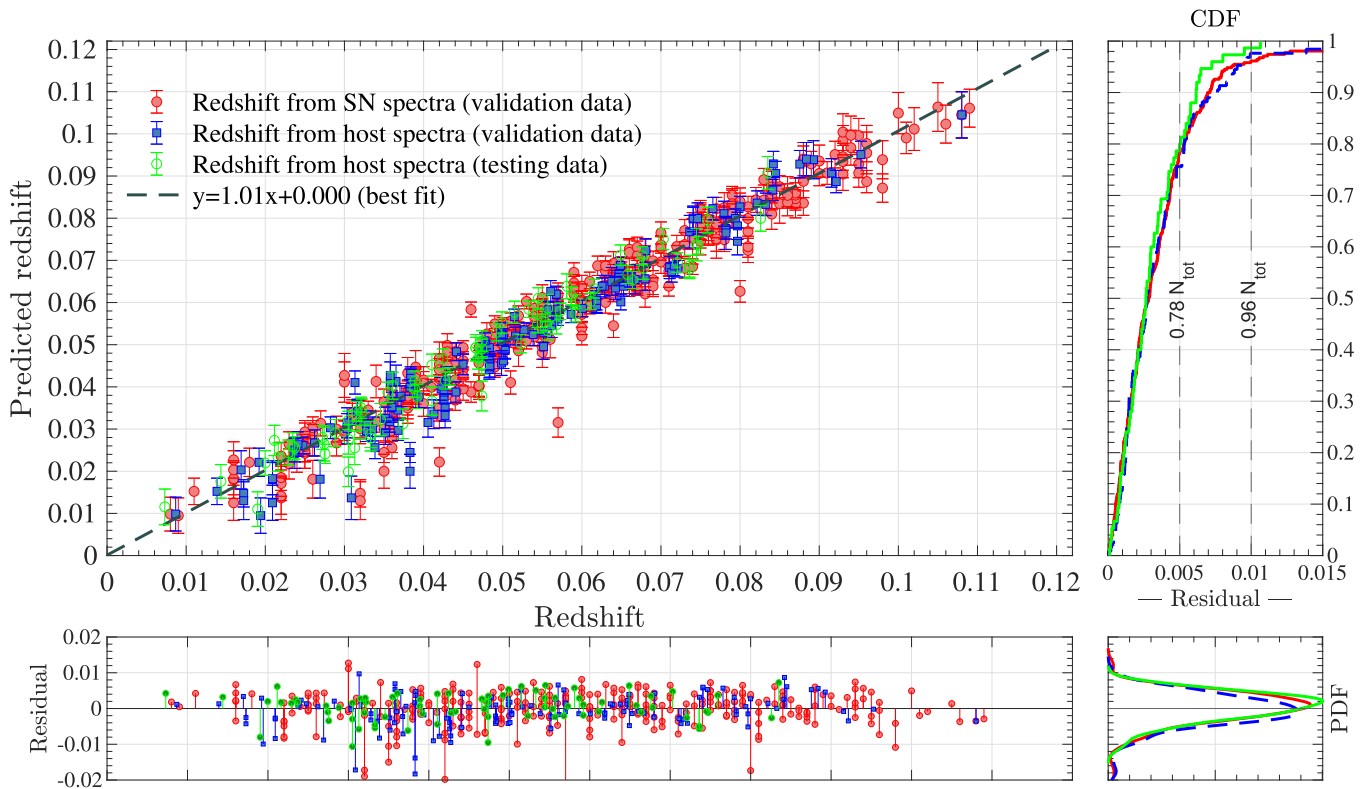


Figure 5. Redshift regression performance for `SNIAscore` on the BTS18 validation sample (red and blue markers and lines) and on the testing data set (green markers and lines). We find no evidence for systematic bias in the redshift predictions of `SNIAscore`. The best fit to the validation data is shown as a dashed black line in the main panel. The fit is consistent with $y = x$ within the uncertainties (the standard errors are $\pm 0.01x$ and ± 0.001), and we do not find significant evidence for systematic bias in the redshift residuals (bottom-left panel) of either the validation or testing data sets. Probability density functions for the residuals are shown in the bottom right panel. The typical uncertainty of a redshift estimate is < 0.005 , based on the cumulative distribution functions for the absolute value of the residuals of both the validation and testing data sets (top-right panel).

Thus, as there is only one SN Ic spectrum that gives a high `SNIAscore` value in each of the validation and testing data sets, it appears that only in rare cases do SNe Ic cause confusion for the BTS (which probes $z < 0.1$ for core-collapse SNe). Furthermore, if the lightcurves of these SNe are taken into account it appears relatively straightforward to remove such objects from any sample of SNe Ia that is required to be devoid of both core-collapse SNe and highly peculiar SNe Ia.¹⁹

For future versions of `SNIAscore`, we plan to add the option to include lightcurve information as input. Looking at objects that change `SNIAscore` significantly when the lightcurve is included or excluded may turn out to be a way to identify rare events that would warrant further follow-up from larger facilities. However, the intended use case of `SNIAscore` is primarily to provide live classifications when SEDM is observing. As such, a potential limitation is the fact that we do not necessarily have much lightcurve information available at the time of the spectral observation by SEDM. A significant fraction of our spectra are observed before maximum light (Figure 1), and full lightcurves cannot be leveraged.

Some work on machine-learning methods for early photometric classification has been done (e.g., RAPID; Muthukrishna et al. 2019a). However, we have not been able to reproduce the expected performance in Muthukrishna et al. (2019a) on live ZTF data. Work on lightcurve classifiers

trained on ZTF data is in progress, which we plan to eventually combine with `SNIAscore` and the redshift predictions that we already produce. Work on a deep-learning classifier capable of distinguishing between more SN subtypes while maintaining a comparably low FPR is also ongoing (Y. Sharma et al. 2021, in preparation).

7. Summary and Implementation

With `SNIAscore` we have shown that a simple RNN architecture (Section 3) can be used to create a method that is capable of fully automatic classifications of low-resolution SN Ia spectra obtained by the SEDM instrument. Most importantly, `SNIAscore` achieves a consistently low FPR of $< 1\%$ when classifying up to 90% of the SN Ia spectra that SEDM observes during routine operation (Section 5.1). For securely classified SN Ia spectra, `SNIAscore` also achieves excellent redshift regression performance, comparable to determining redshifts by carefully running each spectrum manually through SNID. Typically, the redshift predictions made by `SNIAscore` differ by less than $z \pm 0.005$ when compared to host-galaxy redshifts (Section 5.2).

Starting from 2021 April 15 we have fully deployed `SNIAscore` as part of the automated ZTF SEDM pipeline (Fremling et al. 2021). This includes automatic reporting of confident SN Ia classifications and redshift estimates to TNS. With this, we have achieved automation all the way from (spectroscopic) observation to data reduction and spectroscopic classification for the first time.

¹⁹ We note that with the low resolution of SEDM, it is not possible to spectroscopically separate between the various SN Ia subtypes, except in the most clear-cut cases (see Fremling et al. 2020).

The first fully automated SEDM SN Ia classification was sent for SN 2021ijb (SNIAscore 2021). This report was sent within roughly 10 minutes after the finished SEDM exposure. With automated ZTF candidate vetting and automatic triggering of SEDM, which is currently possible through AMPEL (Nordin et al. 2019) and the Fritz marshal (Duev et al. 2019; van der Walt et al. 2019), a fully automated imaging (ZTF) and spectroscopic (SEDM) survey is now becoming realistic.

Initially for our TNS reports, we are using a very conservative SNIAscore threshold of 0.9. Based on the performance evaluation in Section 5.1, we plan to eventually relax this to a cut on the SNIAscore of >0.8 combined with a cut on $\sigma_{\text{SNIAscore}}$ of <0.275 . This will eliminate the need for human interaction in the classification process for the majority of BTS SNe Ia ($\approx 80\%$ based on our testing data or $\approx 90\%$ based on our validation data), which translates to $>50\%$ of BTS SNe.

We do not find significant variations in the FPR as a function of time or overall data quality (Appendix C), based on the data currently available to us. However, we note that the corresponding TPR (e.g., Table 2) and the numbers above are subject to changing observing conditions and the condition of the telescope and instrument (lower S/N results in lower SNIAscore). During 2020 and the first half of 2021, it has not been possible to realuminize the primary mirror of the Palomar 60 inch telescope. Due to this, our testing data set is affected by decreased overall throughput (which we currently attempt to compensate for by increased exposure times). It is possible that an increased number of SN Ia spectra can be classified when the mirror has been realuminized. Work on improvements to the SEDM data reduction pipeline is also ongoing (e.g., more effective cosmic-ray rejection and host-galaxy background subtraction).

SED Machine is based upon work supported by the National Science Foundation under grant No. 1106171. Based on observations obtained with the Samuel Oschin Telescope 48 inch and the 60 inch Telescope at the Palomar Observatory as part of the Zwicky Transient Facility project. ZTF is supported by the National Science Foundation under grant No. AST-1440341 and a collaboration including Caltech, IPAC, the Weizmann Institute for Science, the Oskar Klein Center at Stockholm University, the University of Maryland, the University of Washington, Deutsches Elektronen-Synchrotron and Humboldt University, Los Alamos National Laboratories, the TANGO Consortium of Taiwan, the University of Wisconsin at Milwaukee, and Lawrence Berkeley National Laboratories. Operations are conducted by COO, IPAC, and UW. This work was supported by the GROWTH project funded by the National Science Foundation under PIRE grant No. 1545949. The Oskar Klein Centre is funded by the Swedish Research Council. C.F. gratefully acknowledges support of his research by the Heising-Simons Foundation (#2018-0907). M.R. has received funding from the European Research Council (ERC) under the European Union’s Horizon 2020 research and innovation program (grant agreement No. 759194—USNAC). M.W.C. acknowledges support from the National Science Foundation with grant No. PHY-2010970. Based in part upon work supported by the LSST Corporation (LSSTC), through Enabling Science Grants #2019-UG01 and #2020-01.

Facilities: P48, P60(SEDM).

Software: MATLAB (MATLAB 2020), SNID (Blondin & Tonry 2007), DASH (Muthukrishna et al. 2019b), the GROWTH Marshal (Kasliwal et al. 2019), Fritz (<https://github.com/fritz-marshall/fritz>).

Appendix A Loss Function Definitions

The MSE is the mean of the squared distances between the target variable (y_i) and predicted values (y_i^p):

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (y_i - y_i^p)^2. \quad (\text{A1})$$

The MBE is the mean of the distances between y_i and y_i^p :

$$\text{MBE} = \frac{1}{n} \sum_{i=1}^n (y_i - y_i^p). \quad (\text{A2})$$

The MPE is the mean of the distances between y_i and y_i^p divided by y_i :

$$\text{MPE} = \frac{1}{n} \sum_{i=1}^n \frac{(y_i - y_i^p)}{y_i}. \quad (\text{A3})$$

The mean absolute error (MAE) is the mean of the absolute distances between y_i and y_i^p :

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |y_i - y_i^p|. \quad (\text{A4})$$

Appendix B SNID and DASH ROC Curves

To evaluate the performance of SNID (v5.0), we use the standard template bank included with the software²⁰; the SNe Ia and non-SN templates from the Berkeley SN Ia program (BSNIP; Silverman et al. 2012); the SN Ib and Ic templates from Modjaz et al. (2014, 2016), Liu et al. (2016), and Williamson et al. (2019); and the SN IIP templates from Gutiérrez et al. (2017). The ROC curve shown in Figure 4 was computed by varying the minimal threshold for the `rlap` parameter between 0 and 25. For each automatic classification we take the best SNID match with the highest `rlap` value for that particular spectrum, as we have found that this performs better than considering multiple matches (e.g., counting how many SN Ia matches are among the top 10 best matches). The redshift range was restricted to a reasonable range for the SNe expected to be found by the BTS ($z < 0.2$). We place no restriction on the phase of the templates, and we have not attempted to restrict the template set used when running SNID. For future work it may be of interest to investigate if there is an optimal set of templates to use for SN Ia binary classification.

In order to evaluate DASH (v1.0 with Models_v06), we used the default template bank included with the software.²¹ To create the ROC curve we have investigated the effect of cuts on both the `softmax` scores and the `rlap` values that DASH produces. We investigate the ranges `softmax` > 0.2 –1 and `rlap` > 0 –25. To create the ROC curve shown in Figure 4 we consider all possible cuts in the range `softmax` > 0.5 –0.9 and `rlap` > 0 –25, because for this range we observe the most

²⁰ <https://people.lam.fr/blondin.stephane/software/snid/#Download>

²¹ <https://github.com/daniel-muthukrishna/astrodash>

stable behavior in DASH. More extreme cuts on $\text{softmax} > 0.92\text{--}0.99$ can offer some improvement on the BTS18 data set, but only for one specific and narrow range of FPR (FPR = 0.02), which then gives TPR = 0.62. The rest of the ROC curve becomes noisy and with no real improvement over the one produced for $\text{softmax} > 0.5\text{--}0.9$. As such, we consider this behavior to be unstable and do not recommend extreme cuts on the softmax score for SEDM data.

Appendix C

Classification Performance Depending on Data Quality

It is of interest to investigate how SNIascore performs as a function of data quality. For SN classification, the S/N of a spectrum on its own is not an ideal tracer, because S/N cannot separate between SN-dominated and host-galaxy-dominated spectra. Instead, we use the rlap values produced by SNID, which correlate positively with S/N and negatively with host-galaxy contamination. To create a high-S/N sample, we use our testing data set (Table 1) and select spectra with $\text{rlap} \geq 0.8$. To create a low-S/N sample, we select spectra with $0 < \text{rlap} < 8$.

The performance of SNIascore on these high- and low-S/N samples is shown in Figure 6. There is a clear

correlation between S/N and SNIascore . For real SN Ia spectra, the scores for low-S/N spectra peak at 0.95. The scores for high-S/N SN Ia spectra peak at 1, with the majority of high-S/N SN Ia spectra receiving scores higher than 0.95. The number of false positives at high SNIascore does not decrease in the high-S/N sample compared to the low-S/N sample. Selecting only high-S/N spectra is thus unlikely to eliminate the most likely false positives discussed in Section 6.

Most importantly, we find that very low FPR can be achieved regardless of the overall S/N of the sample and that the same cut to minimize FPR ($\text{SNIascore} > 0.8$), found in Section 5.1 remains appropriate. SNIascore has successfully learned to consistently give low scores to bad-quality data. This finding is important because ground-based observations are always subject to bad observing conditions.

It is unnecessary to apply any external cuts based on S/N before running SNIascore on the data SEDM is producing that pass the quality checks of the data reduction pipeline (Blagorodnova et al. 2018; Rigault et al. 2019). The main impact of conditions leading to data of bad quality is that fewer spectra out of the total number of spectra that are observed can be successfully classified.

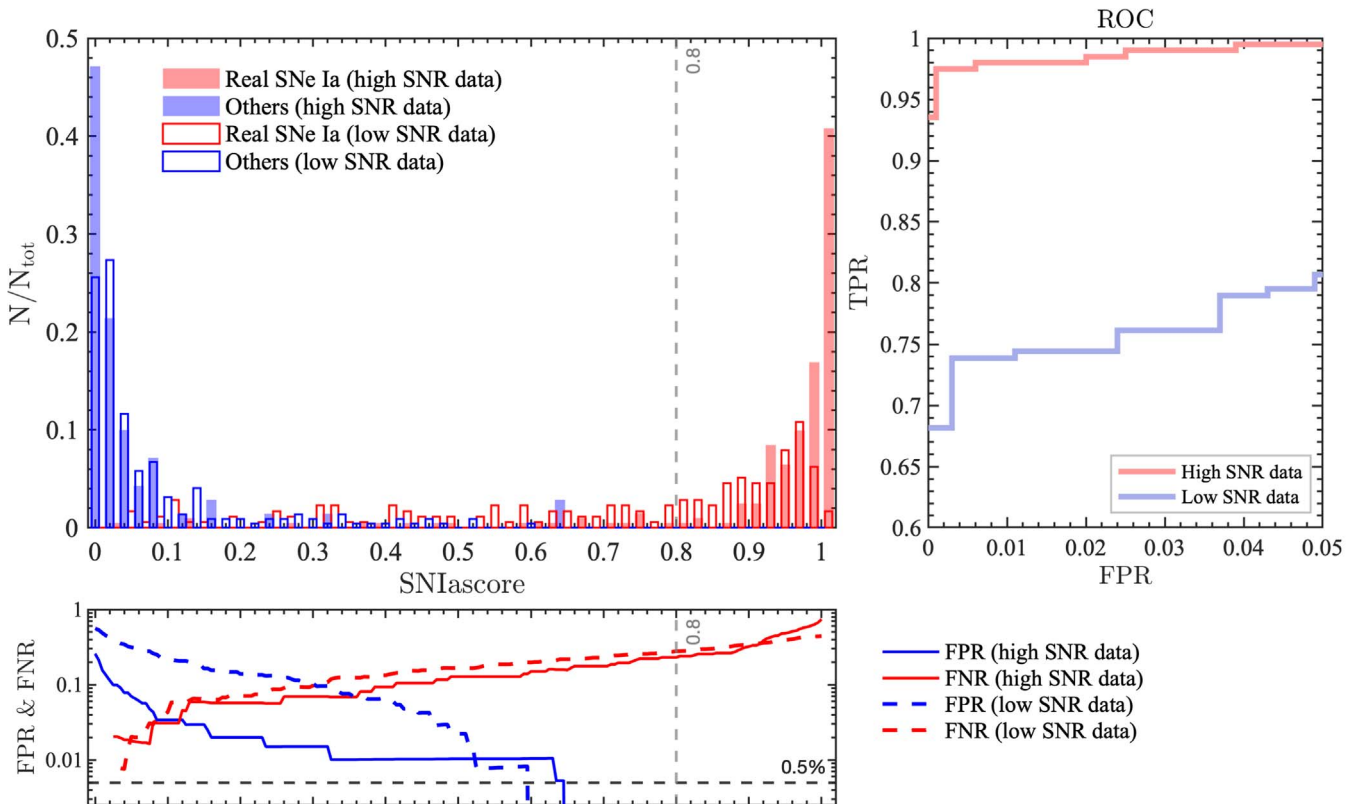


















Figure 6. Classification performance of SNIascore on high-S/N ($\text{rlap} \geq 0.8$) and low-S/N ($0 < \text{rlap} < 8$) subsamples selected from the testing data set. Top-left panel: the SNIascore distributions for both “Real SNe Ia” spectra (red) and spectra of “Other” transients (blue). The high-S/N data set is shown as solid bars, and the low-S/N data set is shown as open bars. N_{tot} is the total for each class separately (i.e., the sum of the “Real SNe Ia” bins add up to unity). Bottom panel: FPR and FNR as a function of SNIascore for the high-S/N (solid red and blue lines) and for the low-S/N (dashed red and blue lines) data sets. A cut at $\text{SNIascore} > 0.8$ results in an FPR of zero on both the high- and low-S/N data sets. Right panel: ROC curve for SNIascore on the high-S/N data set (red line) and the low-S/N data set (blue line).

ORCID iDs

Christoffer Fremling  <https://orcid.org/0000-0002-4223-103X>
 Michael W. Coughlin  <https://orcid.org/0000-0002-8262-2924>
 Dmitry A. Duev  <https://orcid.org/0000-0001-5060-8733>
 Matthew J. Graham  <https://orcid.org/0000-0002-3168-0139>
 Mansi M. Kasliwal  <https://orcid.org/0000-0002-5619-4938>
 Erik C. Kool  <https://orcid.org/0000-0002-7252-3877>
 Ashish A. Mahabal  <https://orcid.org/0000-0003-2242-0244>
 Adam A. Miller  <https://orcid.org/0000-0001-9515-478X>
 Daniel A. Perley  <https://orcid.org/0000-0001-8472-1996>
 Mickael Rigault  <https://orcid.org/0000-0002-8121-2560>
 Ben Rusholme  <https://orcid.org/0000-0001-7648-4142>
 Yashvi Sharma  <https://orcid.org/0000-0003-4531-1745>
 Kyung Min Shin  <https://orcid.org/0000-0002-1486-3582>
 David L. Shupe  <https://orcid.org/0000-0003-4401-0430>
 Jesper Sollerman  <https://orcid.org/0000-0003-1546-6615>
 S. R. Kulkarni  <https://orcid.org/0000-0001-5390-8563>

References

- Bellm, E. C., Kulkarni, S. R., Barlow, T., et al. 2019a, *PASP*, **131**, 068003
 Bellm, E. C., Kulkarni, S. R., Graham, M. J., et al. 2019b, *PASP*, **131**, 018002
 Blagorodnova, N., Neill, J. D., Walters, R., et al. 2018, *PASP*, **130**, 035003
 Blondin, S., & Tonry, J. L. 2007, in AIP Conf. Ser. 924, The Multicolored Landscape of Compact Objects and Their Explosive Origins, ed. T. di Salvo et al. (Melville, NY: AIP), 312
 Dahiwal, A., & Fremling, C. 2020, *TNSCR*, **2020-947**, 1
 Dekany, R., Smith, R. M., Riddle, R., et al. 2020, *PASP*, **132**, 038001
 Duev, D. A., Mahabal, A., Masci, F. J., et al. 2019, *MNRAS*, **489**, 3582
 Fremling, C., Dahiwal, A., Mahabal, A., et al. 2021, *TNSAN*, **2021-122**, 1
 Fremling, C., Miller, A. A., Sharma, Y., et al. 2020, *ApJ*, **895**, 32
 Fremling, C., & Sharma, Y. 2018, *TNSCR*, **2018-886**, 1
 Gal, Y., & Ghahramani, Z. 2015, arXiv:1506.02142
 Graham, M. J., Kulkarni, S. R., Bellm, E. C., et al. 2019, *PASP*, **131**, 078001
 Gutiérrez, C. P., Anderson, J. P., Hamuy, M., et al. 2017, *ApJ*, **850**, 89
 Kasliwal, M. M., Cannella, C., Bagdasaryan, A., et al. 2019, *PASP*, **131**, 038003
 Kingma, D. P., & Ba, J. 2014, arXiv:1412.6980
 Kulkarni, S. R. 2020, arXiv:2004.03511
 Liu, Y.-Q., Modjaz, M., Bianco, F. B., & Graur, O. 2016, *ApJ*, **827**, 90
 Masci, F. J., Laher, R. R., Rusholme, B., et al. 2019, *PASP*, **131**, 018003
 MATLAB 2020, MATLAB, 2020, version 9.8.0.1359463 (R2020a), The MathWorks, Inc., Natick, MA
 Modjaz, M., Blondin, S., Kirshner, R. P., et al. 2014, *AJ*, **147**, 99
 Modjaz, M., Liu, Y. Q., Bianco, F. B., & Graur, O. 2016, *ApJ*, **832**, 108
 Muthukrishna, D., Narayan, G., Mandel, K. S., Biswas, R., & Hložek, R. 2019a, *PASP*, **131**, 118002
 Muthukrishna, D., Parkinson, D., & Tucker, B. E. 2019b, *ApJ*, **885**, 85
 Nordin, J., Brinnel, V., van Santen, J., et al. 2019, *A&A*, **631**, A147
 Perley, D. A., Fremling, C., Sollerman, J., et al. 2020, *ApJ*, **904**, 35
 Rigault, M., Neill, J. D., Blagorodnova, N., et al. 2019, *A&A*, **627**, A115
 Shappee, B., Prieto, J., Stanek, K. Z., et al. 2014, AAS Meeting, **223**, 236.03
 Sherstinsky, A. 2020, *PhyD*, **404**, 132306
 Silverman, J. M., Foley, R. J., Filippenko, A. V., et al. 2012, *MNRAS*, **425**, 1789
 SNIascore 2021, *TNSCR*, **2021-1225**, 1
 Tonry, J., Denneau, L., Heinze, A., et al. 2020, *TNSTR*, **2020-863**, 1
 Tonry, J., Stalder, B., Denneau, L., et al. 2018a, *TNSTR*, **2018-818**, 1
 Tonry, J. L., Denneau, L., Heinze, A. N., et al. 2018b, *PASP*, **130**, 064505
 van der Walt, S. J., Crellin-Quick, A., & Bloom, J. S. 2019, *JOSS*, **4**, 1247
 Williamson, M., Modjaz, M., & Bianco, F. B. 2019, *ApJL*, **880**, L22