The Astronomical Journal, 159:16 (14pp), 2020 January https://doi.org/10.3847/1538-3881/ab557d

An Information Theory Approach on Deciding Spectroscopic Follow-ups

Javiera Astudillo[1], Pavlos Protopapas[2], Karim Pichara[1,2,3], and Pablo Huijse[3,4]
[1] Computer Science Department, School of Engineering, Pontificia Universidad Católica de Chile, Chile; jfastudillo@uc.cl
[2] Institute for Applied Computational Science, Harvard University, Cambridge, MA, USA
[3] Millennium Institute of Astrophysics, Chile
[4] Informatics Institute, Universidad Austral de Chile, Valdivia, Chile
Received 2019 May 31; revised 2019 November 5; accepted 2019 November 6; published 2019 December 13

Abstract

Classification and characterization of variable phenomena and transient phenomena are critical for astrophysics and cosmology. These objects are commonly studied using photometric time series or spectroscopic data. Given that many ongoing and future surveys are conducted in a time domain, and given that adding spectra provides further insights but requires more observational resources, it would be valuable to know which objects we should prioritize to have a spectrum in addition to a time series. We propose a methodology in a probabilistic setting that determines a priori which objects are worth taking a spectrum of to obtain better insights, where we focus on the insight of the type of the object (classification). Objects for which we query their spectrum are reclassified using their full spectral information. We first train two classifiers, one that uses photometric data and another that uses photometric and spectroscopic data together. Then for each photometric object we estimate the probability of each possible spectrum outcome. We combine these models in various probabilistic frameworks (strategies), which are used to guide the selection of follow-up observations. The best strategy depends on the intended use, whether it is obtaining more confidence or accuracy. For a given number of candidate objects (127, equal to 5% of the data set) for taking spectra, we improve the class prediction accuracy by 37% as opposed to 20% of a non-naive (non-random) best-baseline strategy. Our approach provides a general framework for follow-up strategies and can be extended beyond classification to include other forms of follow-ups beyond spectroscopy.

Unified Astronomy Thesaurus concepts: Computational methods (1965); Astronomy data analysis (1858); Astrostatistics tools (1887); Variable stars (1761)

1. Introduction

Variable phenomena have been astronomical objects of great interest as they reveal important information about our universe (Groenewegen 2018). Some examples are RR Lyrae stars, which are used to trace distances within our galaxy, allowing us to improve our understanding of the Milky Way structure and evolution (Minniti et al. 2016); Mira variables, which are long-period red giants that correspond to a late-stage phase in the evolution of stars like our Sun (Perrin et al. 2004); Cepheids, which are used as distance indicators with well-studied physical properties (Tanvir 1999; Freedman et al. 2001); supernovae, which are explosive events, some of which act as standard candles in the cosmological distance scale and have been key in recent discoveries related to dark energy (Riess et al. 1998; Perlmutter et al. 1999; Hicken et al. 2009); and quasars, or active galactic nuclei (AGN), which are static transients that help us understand the nature of their host galaxies (Nolan et al. 2001) and early stages of galaxy formation (Eilers et al. 2018). The classification of transients and the identification of novel variable phenomena is critical for astrophysics and cosmology research. This is reflected in the scientific objectives of past, current, and near-future wide-field time-domain surveys such as the Supernova Legacy Survey (Astier et al. 2006; Perrett et al. 2010); ESSENCE (Miknaitis et al. 2007), the Large Synoptic Survey Telescope (LSST; Ivezić et al. 2008); the Sloan Digital Sky Survey-II (SDSS-II; Frieman et al. 2008; Sako et al. 2008); the Square Kilometre Array (Lazio 2009); the Catalina Real-Time Transient (CRTS; Djorgovski et al. 2011); the Dark Energy Survey (Bernstein et al. 2012; Abbott et al. 2018); the Palomar Transient Factory (Surace 2015), which transitioned to the Zwicky Transient Facility (ZTF; Smith et al. 2014); and the Panoramic Survey Telescope and Rapid Response System (Pan-STARRS1; Chambers et al. 2016; Scolnic et al. 2018). Of these, ongoing surveys such as the ZTF (Smith et al. 2014) and future survey like the LSST (Ivezić et al. 2008) will provide extensive data sets, with an estimated number of alerts per night equal to 0.1 million for the ZTF (Masci et al. 2019) and 10 million for the LSST (LSST 2014) from which variable phenomena have to be automatically detected.

Photometry from time-domain surveys allows us to detect time-variable phenomena such as explosions, accretion, pulsations, eclipses, and relativistic phenomena that cannot be detected by other means (Djorgovski et al. 2011; Kessler et al. 2015). The emerging of synoptic sky surveys that scan sky areas repeatedly have leveraged time-domain astronomy in past years (Djorgovski et al. 2012), so that recent and future research for variable phenomena discovery and classification of different real transients has focused on automatic methods that mostly rely on photometry (Debosscher et al. 2007; Richards et al. 2011; Bloom & Richards 2012; Pichara et al. 2012; Pichara & Protopapas 2013; Mackenzie et al. 2016; Castro et al. 2018; Martínez-Palomera et al. 2018).

In addition to photometry, spectroscopic data provide information such as physical properties: gravity, temperature, chemical compositions, and radial velocities, which are hard to obtain in any other way (Massey & Hanson 2013). Spectroscopic surveys usually target objects selected from photometric surveys, and often their main purpose is to obtain a redshift (Djorgovski et al. 2013). Examples are The Baryon Oscillation Spectroscopic Survey (BOSS; Dawson et al. 2013) and the Large Sky Area Multi-object Fiber Spectroscopic Telescope

(LAMOST; Luo et al. 2015). In addition to redshift estimation, spectra become helpful for object identification. An example would be supernovae (SNe) Ia, which play a role as distance indicator and are distinguishable from other type of supernovae mainly by the lack of hydrogen lines in their spectra and the presence of pronounced silicon features (Campbell et al. 2013). Another example are quasars or AGNs, which are among the brightest objects and help us understand the nature of their host galaxies (Nolan et al. 2001) and early stages of galaxy formation (Eilers et al. 2018). The most certain way to confirm these object types and redshift is through the analysis of their spectroscopy (Peters et al. 2015). Last, RR Lyraes may be distinguished from eclipsing binaries through their spectra jointly with their light curves (Kinman & Brown 2010). A spectrum with a resolution of 1 Å can provide enough information to distinguish between a pulsating variable and an eclipsing binary even at low amplitudes.

Although photometry usually is the first available resource for inspecting an object, the spectra ultimately let us understand their physical properties. It is consequently beneficial to have both to achieve better insights. Unfortunately, spectra use many more resources (observational time) and are therefore infrequently available. For instance, the SDSS (York et al. 2000), an ongoing photometric and spectroscopic survey, has imaged nearly one billion objects, while for spectra it has around four million objects.[5] Because of this reason, the spectroscopic follow-up strategy remains a challenge, primarily because of the increasing data stream from imaging surveys along with potential new interesting objects that need to be studied in depth. Assuming that we mostly have photometric data and that we can achieve better prediction of the object class by adding spectra at a specific cost, which objects should we prioritize?

In this paper, we propose a model that efficiently finds the celestial objects for which obtaining spectra could improve classification results, with regard to either accuracy or confidence. Each unique object in the data set is classified, using its time series, into one of the classes with an associated level of confidence. Classification of a given object may be poor either because the classification is incorrect (wrong class) or because the confidence in the classification is low. For each object, if its spectrum is queried and used alongside its time series to reclassify the object, either the class or the confidence, or both, may change. Note that we aim to use the full spectral information in addition to time-series information to make more informed classifications. Our model helps us to find the objects for which querying for their spectra helps to change to the true class or increase its confidence. Similarly, it avoids objects of which we either know, with high confidence, what their class is or of which there is no chance of improving our knowledge.

Our model assigns a priority to each object by measuring the information gain and indicator (of the classification change) outcome caused by the addition of a spectrum to already available time series. For this, we first extract features from objects (in both a training and a test set) for which both time series and spectrum are available. For spectrum features, we first learn automatic feature extraction using autoencoders (Section 3.2), while for time series, we use a set of already existing features. We then train classifiers on time series only and on time series and spectra combined. Then we leave the spectra aside and estimate the spectrum for each object given only its time series. For each object, we estimate the information gain and an indicator between the estimated spectra jointly with the time series with just the time series. We assign a priority to each object using the latter estimators, and a subset of the objects with the highest priority is selected for follow-ups. Objects in the subset are classified according to their time-series and real spectrum features, whereas non-selected objects are classified using only time-series features. Our approach is mainly focused on catalogs of variable phenomena, although it could be applied to any object that has available time series, and potentially available spectra. Note that for at least a subset of them we need both spectra and time series for the training phase.

Section 2 presents related work. Section 3 describes the general background theory. Section 4 defines how we assess the information gain and the indicator, and Section 5 is dedicated to the proposed method. In Section 6 we summarize the data we used for the experiments. Section 7 mentions the libraries and hardware we used to implement the different models. Section 8 presents the results from the experiments and further work. Finally, Section 9 presents overall conclusions and future work.

2. Related Work

In this section, we describe related works and compare them to ours. First, we present the work of Peters et al. (2015), which combines different types of data. We then describe the spectrum follow-up strategy of Ishida et al. (2019), which selects objects for labeling to efficiently train a classifier. Next, we show the design of the experiment approach mentioned in Yang et al. (2015), which deals with the optimal set of filters for estimating the true spectral energy distributions (SED). Last, we describe the Xia et al. (2016) method, which optimizes the decision-making process regarding which objects to observe given limited telescope time.

Peters et al. (2015) aimed to combine different types of data to enhance classification. They compare the objects using band color features only, time-domain features only, and both for the task of detecting quasars, which implies a classification into quasar or non-quasar. The color features correspond to those of the SDSS (York et al. 2000) adjacent colors ($u-g$, $g-r$, $r-i$, $i-z$), while the time-series features come from fitting a structure function (Schmidt et al. 2010) that characterizes the variability of a time series for each band and object. This is motivated by the fact that variability-based classification misses some quasars at high redshifts, while a color-based classification becomes highly confused at medium-redshift zones. Their results show that the highest completeness (the number of known quasars correctly classified as quasars divided by the number of known quasars) is reached using both types of features. This is consistent with our motivation for combining data to obtain a better insight into selected variable phenomena. Similar to our work, they seek to improve classification results by increasing the available information (more features) of the objects and do not improve classification models for the same information. Differently from us, they use features from broad color bands, while we use spectra. Spectra have much more detailed color information.

A relevant work is the spectrum follow-up strategy of Ishida et al. (2019) because, as proposed in this work, it selects objects for spectrum querying, although it was proposed for a different

[5] For further details regarding SDSS data statistics, see SDSS Scope page.

use case. The authors focus on improving type Ia versus non-Ia SNe photometric classification using active learning (AL; Cohn et al. 1996). AL refers to algorithms that select the best objects to label in order to improve classification prediction the most at each training step. It is thereby a way to improve a classification prediction at minimum labeling cost. In their work, labeling is obtained by means of spectrum confirmation because SNe Ia may be distinguished from non-Ia through some specific spectral features that are linked to a particular physical process (Campbell et al. 2013).

The classifiers obtained through their method can improve purity (reduce the incorrectly predicted SNe Ia) compared to two proposed baseline strategies: the passive-learning strategy, and the canonical strategy. In the passive-learning strategy, objects to be labeled are randomly selected from the unlabeled set, while in the canonical strategy, they are randomly selected from a sample that closely follows their initially labeled dataset distribution. Similar to our work, they propose a spectroscopic follow-up design strategy that uses AL strategies to select follow-up objects. The main difference with our work is that we propose to use spectral features to improve the reliability of the prediction, while they query for labels (through the spectra) to retrain the classifier to reduce the false-positive predictions. We automatically extract features that compress the full spectral information into a set of features so as to increase the available information of each object. This may potentially complement regular classification (by an expert) because the latter also entails some uncertainty. This allows taking full advantage of the spectra and thus obtaining more informed decisions when classifying.

The described follow-up strategy could belong to the much broader umbrella of the design of experiment area (Fisher 1935). Generally speaking, it concerns which data should be collect for an experiment and how this should be done to avoid wasting resources or missing important data. One example of this sort is the work of Yang et al. (2015), which determines how the optimal set of filters can be chosen to be used to estimate the true SED of an astronomical object. Similar to our work, the authors aim to make efficient use of telescope resources. Differently from us, they optimize the observations (i.e., which and how many filters) that are required to build the SED for a single object, while we decide which objects to observe in the first place (i.e., taking the spectra).

The work of Xia et al. (2016) presents a method for selecting objects to observe in a batch fashion optimizing use of telescope resources. It takes into account the field of view (FOV) of the telescope in use so that many objects may be observed at the same time. It also proposes updating and recommending objects to observe in batches, given that the infrastructure in a telescope is not able to update the observations schedule in real time, but it can do this in batches. The authors use AL to select objects and take advantage of the location of the candidates in the sky to reach feasible ways of scoring them. They use other AL approaches to evaluate the performance of their method. These include (1) randomly selecting the points, (2) selecting the most uncertain points for the current classifier, (3) selecting the most undersampled points according to the training set distribution, (4) selecting points that maximize the change in the predicted probabilities, and (5) AL and semi-supervised learning methods from other works. Their approach surpasses all of the rest regarding the observing time and number of queries required to reach a given accuracy. Similar to our work, they select objects to observe to improve the classifier prediction without wasting telescope resources. Differently from us, they do not mix different sourced data, but instead only use time-series features (FATS; Nun et al. 2015) to classify. Furthermore, their focus is not to add information to already observed objects and hence gain better insight, but rather to improve their classifiers through labeling.

3. Background Theory

3.1. Information Theory and Entropy

Information theory was conceptualized in Shannon (1948) to solve the problem of optimal information transmission over a noisy channel. In his seminal work, Shannon proposed information entropy as a measure of uncertainty. Initially conceived as the average rate of information produced by a stochastic process, it has been broadly used for quantifying information, choice, and uncertainty. Shannon's entropy is defined as

$$H = -\sum_{i=1}^{n} p_i \log p_i, \tag{1}$$

where $p_1 ... p_n$ are the probabilities associated with a set of possible outcomes correspondingly. For this particular work, Equation (1) will always refer to the entropy of one single object. It can be seen from the definition that the less probable an event is, the higher its contribution to entropy, which translates into more information needed to describe those events.

Two of the main properties that make it a suitable measure regarding the amount of information are (1) $H = 0$ when there is only one certain event and therefore no amount of information is needed to describe it. (2) H is maximum when all p_i are equal (i.e., $1/n$), and consequently, the uncertainty on the outcome is maximum so that more information is needed to describe it.

3.2. Autoencoders

Autoencoders (Olshausen & Field 1996; Lee et al. 2006; Vincent et al. 2008) are algorithms that learn data representations automatically. These representations can be used as features for subsequent tasks of classification or clustering. The most basic autoencoder is composed of a deterministic encoding function that maps the input x to a hidden or latent space z and a deterministic decoding function that returns $\hat{x}$ from z. The model is trained by minimizing the error between $\hat{x}$ and x. By imposing that z dimension is much lower than x, we force the model to learn the most relevant features of the data. Different techniques exist to enhance the encoding, such as sparse autoencoders (Olshausen & Field 1996; Lee et al. 2006), which through regularization force sparsity on z, and denoising autoencoders (Vincent et al. 2008), which reconstruct x from a corrupted version of it in order to make the model robust to noise.

Autoencoders have been combined with recurrent neural networks (Hochreiter & Schmidhuber 1997) so as to learn low dimensional representations for time series (Srivastava et al. 2015; Witten et al. 2016). These so-called sequence-to-sequence autoencoders have been developed for dense and

regularly sampled time series. Important work related to our work is Naul et al. (2018), where an autoencoder for astronomical time-series classification was proposed. A shortcoming of this work is that it performs poorly on non-periodic or unfolded time series.

On the other hand, the variational autoencoder (VAE; Kingma & Welling 2013), or alternatively, the deep latent Gaussian model (Jimenez Rezende et al. 2014), is a deep generative model that resembles an encoder and a decoder structure, but instead of looking for a deterministic encoding z or decoding $\hat{x}$, it seeks to estimate their distributions. The authors provide an amortized model for variational inference (VI; Blei et al. 2016), using the same model to estimate the variational parameters of different data points, avoiding costly loops per data point. VI is a family of techniques to approximate computationally intractable posterior distributions via solving an optimization problem. In VAE we approximate the posterior of z given x using a factorized Gaussian distribution. The encodings follow a regular geometry (usually a Gaussian distribution) and are more meaningful than the encodings obtained through a regular autoencoder. This model may be used for artificial data generation, data representation, and inference tasks. Some examples are collaborative filtering (Liang et al. 2018) and image analysis (Wang et al. 2017).

Extensions to VAE for fitting sequential data have been made, such as Bowman et al. (2015) for sentence generation, Fabius & van Amersfoort (2014) for simple video game song data sets, and Chung et al. (2015) for speech and handwriting data sets. Sequence VAE were developed for regularly sampled time series and are therefore not appropriate for astronomical data. It is worth noting that training a sequence VAE is more difficult than conventional VAE. The optimization challenges of sequence VAE are described by Bowman et al. (2015) in their Section 3.1 and byDieng et al. (2018) in their Section 4.2.

4. Problem Description and Notation

Our goal is to select objects for which spectra improve the class prediction most when they are added to its time series, with regard to either accuracy or confidence. We define two metrics to assess classification improvement. The first is information gain and addresses the confidence. It uses the entropy (Shannon 1948, Section 3.1) over some estimator $\hat{y}$ of the label y (the real class of a given object),

$$H(\hat{y}|x_*) = -\sum_{c\in C} P(\hat{y} = c|x_*)\log P(\hat{y} = c|x_*). \quad (2)$$

Here $\hat{y}$ is an arbitrary class predictor given certain data x_* and P is the probability of predicting a certain class c (i.e., $\hat{y} = c$), where c is any class from the set of possible classes to predict C. For example, $\hat{y}$ could be the outcome of any classification method that outputs class probabilities, so that the prediction for a given object with data x_* is the class with the highest probability P. Entropy H (Shannon 1948, Equation (2)) lets us measure the confusion of a probability density function (PDF), such as the outcome of a classification task. To calculate entropy, we need to check the value of the PDF at each value in the domain, which for our case is each of the possible classes to predict. The higher the value of the entropy, the more confused the outcome, while the lower the value, the less confused the outcome. A confused outcome is whenever probabilities are more even between them, and hence we are uncertain of the class. A non-confused outcome is whenever the probabilities are concentrated in one or a few classes, so that we are more certain of the class. Note that our definition of H works for any feature x_*, not only for photometric or spectroscopically features (x_t and x_s, correspondingly). This means that our method could include the addition of any differently sourced information as long as features may be extracted from it.

We now define the information gain for a given object x as the reduction of entropy in the classification outcome caused by the addition of spectrum features x_s to the initially available time-series features x_t:

$$IG(x_t, x_s) = H_t(\hat{y}|x_t) - H_{ts}(\hat{y}|x_t, x_s), \quad (3)$$

where

$$H_t(\hat{y}|x_t) = -\sum_{c\in C} P_t(\hat{y} = c|x_t)\log P_t(\hat{y} = c|x_t)$$
$$H_{ts}(\hat{y}|x_t, x_s) = -\sum_{c\in C} P_{ts}(\hat{y} = c|x_t, x_s)\log P_{ts}(\hat{y} = c|x_t, x_s). \quad (4)$$

We have used H_{ts}, H_t to denote the difference between the distributions of their corresponding class predictors P_{ts}, P_t depending on the given data. Note that we subtract the resulting entropy from using spectrum features to the initial entropy because we wish to measure the entropy reduction, or equivalently, the confusion reduction.

In addition to the information gain metric, we develop a second indicator metric that addresses the classification accuracy. For a given object, it indicates if the predicted class with time-series information is different from the prediction with the spectrum information added. This facilitates the detection of objects that are incorrectly classified (*false positives*) with only time-series features x_t, but are correctly classified when spectrum features x_s are added. The indicator function is as follows:

$$\Delta\hat{y}(x_t, x_s) = \begin{cases} 1, & l_t(x_t) \neq l_{ts}(x_t, x_s) \\ 0, & l_t(x_t) = l_{ts}(x_t, x_s) \end{cases} \quad (5)$$

$$l_t(x_t) = \arg\max_c P_t(\hat{y} = c|x_t) \quad (6)$$

$$l_{ts}(x_t, x_s) = \arg\max_c P_{ts}(\hat{y} = c|x_t, x_s) \quad (7)$$

,where l_t is the class or label predicted given time-series data, x_t, calculated as the class with the highest probability assigned by the predictor $\hat{y}$. Similarly, l_{ts} is the class predicted given time-series and spectrum data, x_t and x_s.

In summary, we have time-series data, x_t, for all objects and spectrum data, x_s, for only some of them. We also have a subset of objects that have $\{x_t, x_s, y\}$, which we use to estimate classifiers P_t and P_{ts}, with their corresponding entropy functions, H_t and H_{ts}, respectively. This is depicted in Figure 1. We wish to select objects that do not have x_s and have not yet been labeled (unknown y) to query for their spectra, so as to improve their classification results the most. These objects are those that either improve their classification confidence or are reclassified into their real class when their spectra are queried, embodied by functions $IG(x_t, x_s)$ (Equation (3)) and $\Delta\hat{y}(x_t, x_s)$ (Equation (5)), respectively. Both these equations assume that we know x_s and require it in their calculations; nevertheless, we evaluate them for objects whose spectra have not been observed yet (unobserved x_s). Our method focuses on how to approximate x_s. We propose to replace the spectra (x_s) with an average over the most probable

Figure 1. Problem description. Time-series data (x_t) are available for all objects in the data set, but spectroscopic data (x_s) for only some of them. A subset of objects has $\{x_t, x_s, y\}$, which are used to estimate P_t, P_{ts}, H_t, and H_{ts}. We wish to select objects from the subset spectra query candidates to query for their spectra so as to improve their classification results the most.

x_s conditioned over x_t. Thus we need to model the conditional distribution of x_s given x_t.

5. Method Description

This section describes the methodology for developing the selection criterion. Section 5.1 describes how we deal with the fact that x_s are not observed. Section 5.2 describes the classification models, P_t and P_{ts}, corresponding to H_t and H_{ts}, respectively. Section 5.3 describes the feature extraction methods for x_t and x_s. Section 5.4 describes the proposed and comparison strategies. Finally, Section 5.5 presents an overview of the proposed method.

5.1. *Unobserved* x_s

Because we do not observe the spectra prior to selecting the candidates, evaluating $H_{ts}(\hat{y}|x_t, x_s)$ from Equation (3) directly is not feasible. Instead, we propose to replace it with an average over the most probable x_s conditioned over x_t, which in turn is represented by a learned distribution $Q(x_s|x_t)$ as follows:

$$\begin{aligned}\overline{H}_{ts}(\hat{y}|x_t) &= \int H_{ts}(\hat{y}|x_t, x_s)\, Q(x_s|x_t) dx_s \\ &= \mathbb{E}_{x_s \sim Q(x_s|x_t)}[H_{ts}(\hat{y}|x_t, x_s)] \\ &\approx \frac{1}{N}\sum_{i=1}^{N} H_{ts}(\hat{y} = y|x_t, \hat{x}_{s,i}), \qquad (8)\end{aligned}$$

$$\hat{x}_{s,i} \sim Q(x_s|x_t). \qquad (9)$$

We use a Gaussian mixture model (GMM) to estimate the joint distribution $Q(x_t, x_s)$ from which we derive the conditional distribution $Q(x_t|x_s)$ using Bayes' rule. We use GMM due to its flexibility to approximate any distribution. We approximate $\overline{H}_{ts}(\hat{y}|x_t)$ through a Markov chain Monte Carlo process, were $\hat{x}_{s,i}$ is the ith spectrum sampled from the GMM conditioned on x_t.

When we replace $H_{ts}(\hat{y}|x_t, x_s)$ for $\overline{H}_{ts}(\hat{y}|x_t)$ in Equation (3), the information gain function of an object is

$$IG(x_t) \approx H_t(\hat{y}|x_t) - \overline{H}_{ts}(\hat{y}|x_t). \qquad (10)$$

Similarly, for $l_{ts}(x_t, x_s)$ defined in Equation (7), the predicted class of $P_{ts}(x_t, x_s)$ is replaced with

$$\begin{aligned}\bar{l}_{ts}(x_t) &= \text{mode}_{x_s \sim Q(x_s|x_t)}[l_{ts}(x_t, x_s)] \\ &\approx \text{mode}_{\hat{x}_{s,i}}[l_{ts}(x_t, \hat{x}_{s,i})], \qquad (11)\end{aligned}$$

$$\hat{x}_{s,i} \sim Q(x_s|x_t). \qquad (12)$$

This means that Equation (5), the indicator function, is replaced with

$$\Delta\hat{y}(x_t) = \begin{cases} 1, & l_t(x_t) \neq \bar{l}_{ts}(x_t) \\ 0, & l_t(x_t) = \bar{l}_{ts}(x_t). \end{cases} \qquad (13)$$

The calculation of our updated information gain $IG(x_t)$ and indicator $\Delta\hat{y}(x_t)$ requires that the classifiers P_t and P_{ts} are trained with $x_t \in \mathbb{R}^t$ and with $\{x_t, x_s\} \in \mathbb{R}^{t+s}$ correspondingly.

5.2. P_t *and* P_{ts}

As mentioned in Section 4, P_t and P_{ts} may be any type of classifier that outputs probabilities rather than just hard predictions. The models in this work are based on random forest (RF) classifiers (Breiman et al. 1984). RF classifiers are shown to be a good compromise between performance, efficiency, and easy training when features are already extracted. P_{ts} is trained over the joint space $\{x_t, x_s\} \in \mathbb{R}^{t+s}$ and P_t over $x_t \in \mathbb{R}^t$ space. The proportion of trees voting for each class is taken as the output probability.

5.3. x_t *and* x_s

In general, the performance of the GMM is inversely proportional to the dimensionality of the features. This is explained by the curse of dimensionality (Bellman et al. 1961), a term that refers to the issues that arise when working in high dimensional spaces. As explained in Bishop (2006), a Gaussian distribution probability mass spreads on the tails as the dimensionality increases, so that most mass becomes concentrated in a thin shell, thereby losing its characteristic shape and becoming unsuitable for some tasks. For this reason, it is more suitable that x_t and x_s are of low dimension.

x_t is a subset of expert features (FATS; Nun et al. 2015), while x_s is built from learned features ($\mu(z)$), where *expert* refers to well-known studied features and *learned* are those that are automatically extracted, for instance, with an autoencoder (Section 3.2). $\mu(z)$ is the mean of the latent space z of a VAE trained to encode and decode spectra. Note that we use a VAE only for spectrum feature extraction and that time series will use a set of already existing features. Spectra need to be preprocessed to have a common input shape for the VAE, as described below in Section 6.4 and depicted in Appendix A.1. As mentioned in Section 3.2, the latent space of a VAE follows a regular geometry, which simplifies the GMM modeling. Because we do not know the most suitable dimensionality of the latent space z for certain, for our framework we try different dimensionalities d and select the dimensionality that best reconstructs the spectra over a test set. The test set contains multiple spectra that were not used for training to provide an unbiased evaluation of the model being tested. We train multiple VAE for $d \in [1, 15]$, $d \in \mathbb{N}$, $z \in \mathbb{R}^d$. The selected d is the one with the least test R^2 (coefficient of determination) between the original spectra and the reconstructed spectra, which is the output of the VAE. As mentioned in Section 3.2, current extensions of VAE for sequences are both difficult to

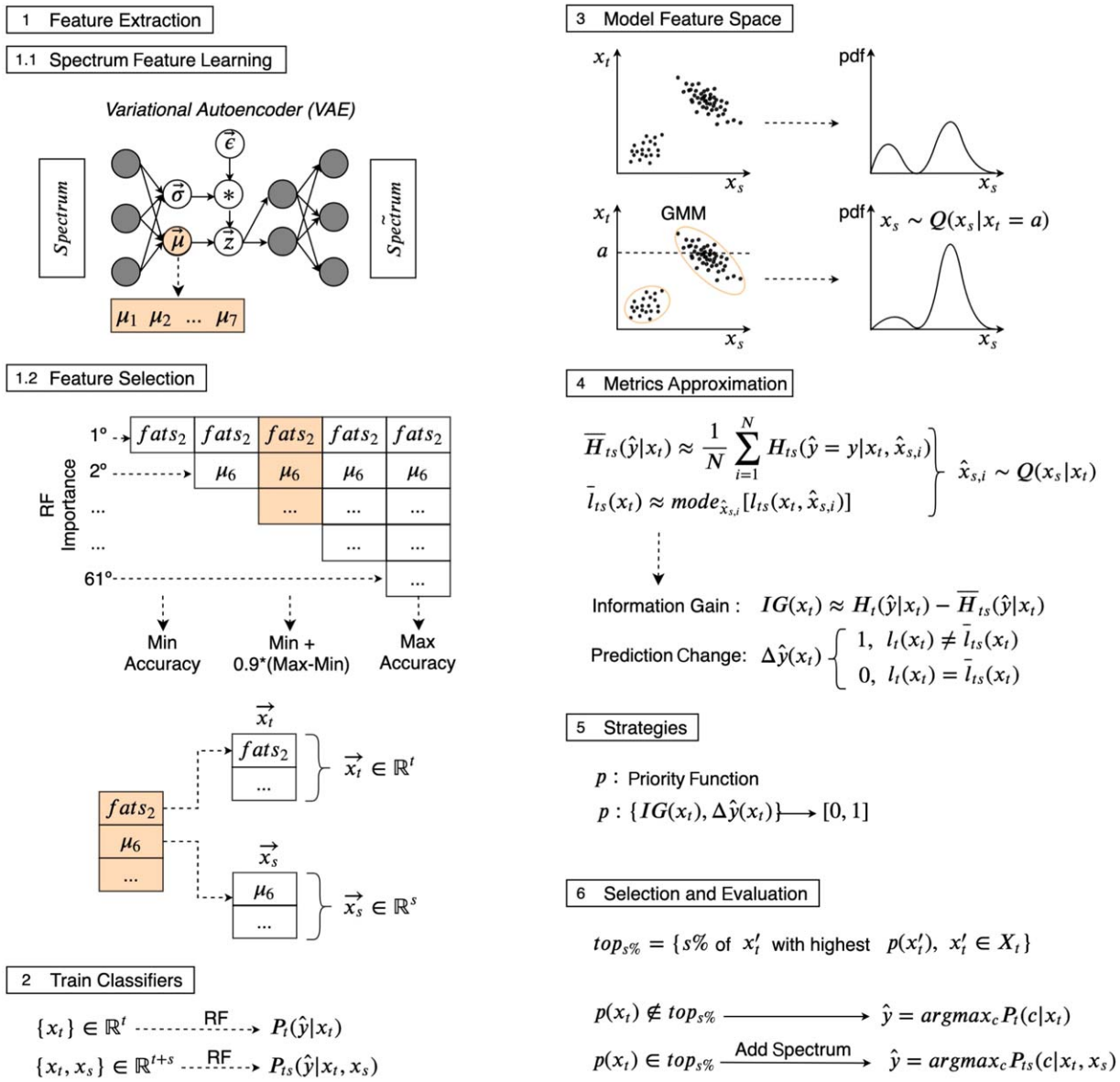

Figure 2. Proposed method overview. (1) We first extract time-series and spectrum features for all objects in our data set. (2) We then train classifiers with these features. (3) After this, we model the joint space of time-series and spectrum features so that (4) the spectrum is estimated from its time-series features to calculate information gain and indicator function. (5) Different strategies are proposed to assign a spectrum query priority to each object. (6) Finally, objects with highest priority are selected and classified using their time-series and spectrum features, while non-selected objects are classified with just time-series features.

train and not capable of dealing with unevenly sampled and non-periodic time series. For time-series features we opt to use a subset of FATS (Nun et al. 2015).

We select $x_t \subseteq$ FATS and $x_s \subseteq \mu(z)$ jointly to avoid repeated information. We build candidate subsets $C_{d'} \subseteq \{FATS, \mu(z)\}$, $C_{d'} \in \mathbb{R}^{d'}$ with $1 \leqslant d' \leqslant |FATS| + |\mu(z)|$. The latter notation refers to the cardinality of the set so that $C_{d'}$ may be a subset of size 1 at the minimum and a subset with a size equal to the number of FATS together with $\mu(z)$ at the maximum. Subset $C_{d'}$ corresponds to the d' most important features (Gini importance) according to an RF classification task as described in Breiman et al. (1984). Note that x_t is not necessarily of the same dimensionality as x_s. The selected features are the smallest $C_{d'}$ that improves the possible improvement of the accuracy by more than 90% between the subset with the worst accuracy and the subset with the best accuracy. The selection is depicted in Figure 2.

5.4. Strategies

To assign a priority to query for spectrum for each object in the data set, we develop multiple strategies and compare them with two baseline strategies and one ideal scenario strategy. Each strategy assigns a priority to each object and selects a subset of objects to whose available information spectrum features x_s are added. For any arbitrary subset size (s), the s objects with the highest priority are selected for each strategy. Selected objects are classified according to their x_t and x_s features. Non-selected objects are classified using only x_t features. We consider two baseline strategies to compare with ours:

1. *Random*: randomly selects the objects from the data set.
2. H_t: selects the objects with the highest entropy on the time-series classification outcome $H_t(\hat{y}|x_t)$. This strategy

is similar to common AL strategies that query for the label, such as Ishida et al. (2019).

We develop three strategies that assign priority to each object according to the estimated information gain $IG(x_t)$ (Equation (10)) and indicator $\Delta\hat{y}(x_t)$ (Equation (13)):

1. $IG(x_t)$: selects the objects with highest approximate information gain $IG(x_t)$ (Equation (10)).
2. $IG(x_t) + \Delta\hat{y}(x_t)$: first selects the objects with $\Delta\hat{y}(x_t) = 1$ (Equation (13)) and then according to the highest $IG(x_t)$. This strategy prioritizes the objects whose predicted class using the sampled spectra (Equation (13)) is different from objects whose class only uses the time series (Equation (6)). The focus is to detect the objects that are incorrectly predicted with time-series features x_t only ($l_t(x_t) \neq y$), but which are correctly predicted when spectrum features x_s are added ($l_{ts}(x_t, x_s) = y$).
3. $H_t + \Delta\hat{y}(x_t)$: first selects the objects with $\Delta\hat{y}(x_t) = 1$ (Equation (13)) and then according to the highest entropy on the time-series classification outcome $H_t(\hat{y}|x_t)$. Similar to the previous strategy, it focuses on detecting incorrectly predicted objects by combining the baseline strategy H_t and our proposed indicator (Equation (13)). It avoids choosing objects with certain classification (low $H_t(\hat{y}|x_t)$) and lifts objects that might have a change in prediction when spectrum information is added.

Finally, one ideal scenario for $IG(x_t)$ is included:

1. *Ideal scenario*: selects the objects according to the current value for $IG(x_t, x_s)$ (Equation (3)) using $H_{ts}(\hat{y}|x_t, x_s)$ instead of its approximation $IG(x_t)$ (Equation (10)), which uses $\overline{H}_{ts}(\hat{y}|x_t)$. Note that this is not a feasible strategy for selection (because it uses x_s), but a reference of how $IG(x_t)$ would perform if the estimation of the spectra for each object were the real spectrum.

5.5. Overview of the Method

We give a brief description of our method below:

1. *Feature extraction*. We extract low dimensional features of time series ($x_t \in \mathbb{R}^t$) and spectra ($x_s \in \mathbb{R}^s$) for all objects that have both available simultaneously. x_t is a subset of expert features (FATS; Nun et al. 2015), while x_s is built from learned features ($\mu(z)$). $\mu(z)$ is the mean of the latent space z of a VAE trained to encode and decode spectra. x_t and x_s are jointly chosen as the smallest subset of the most important features according to an RF classification task, which achieves over 90% of the maximum improvement between the subsets with the lowest and highest accuracy.
2. *Train classifiers.* Two RF classifiers are trained over $x_t \in \mathbb{R}^t$ space and over the joint space $\{x_t, x_s\} \in \mathbb{R}^{t+s}$. The output classifiers are $P_t(\hat{y}|x_t)$ and $P_{ts}(\hat{y}|x_t, x_s)$.
3. *Model feature space.* An empirical GMM is fit over the joint space $\{x_t, x_s\} \in \mathbb{R}^{t+s}$ so that $Q(x_t, x_s)$ is the distribution over the joint space. The conditional distribution $Q(x_s|x_t)$ is obtained from the GMM properties.
4. *Metrics approximation.* For each object we evaluate $H_t(\hat{y}|x_t)$, $\overline{H}_{ts}(\hat{y}|x_t)$, $l_t(x_t)$, $\bar{l}_{ts}(x_t)$, the entropy, and predicted classes of the classifiers trained over the x_t and the $\{x_t, x_s\}$ space, respectively. With these we assess $IG(x_t)$ and $\Delta\hat{y}(x_t)$, the approximate information gain and indicator function.
5. *Strategies.* We assign a priority to each object with calculated metrics through different strategies.
6. *Selection and evaluation.* For each strategy, we select objects with highest priority according to a threshold or to a fixed number of candidates. For selected objects we query for their observed spectrum features and classify them according to $P_{ts}(\hat{y}|x_t, x_s)$. Note that in the latter case there is no need to approximate the spectrum features x_s because it is given information. Non-selected objects are classified with $P_t(\hat{y}|x_t)$.

The above description is depicted in Figure 2.

6. Data

We perform a cross-match over two catalogs to find objects with both time-series and spectra data. The reached surveys are the Catalina Sky Surveys (CSS, Larson et al. 2003) for time series and the SDSS (York et al. 2000) for spectra. Both surveys are better detailed in Sections 6.1 and 6.2, respectively. The built merged data set is better detailed in Section 6.3.

In addition to the previous data set, we gather more spectra to train the VAE for spectrum feature extraction from the SDSS. This is due to the insufficient spectra in the cross-match for training this kind of network. The spectra-only data set is better detailed in Section 6.4.

6.1. CSS Survey

The CSS[6] (Larson et al. 2003) is a NASA-funded project that searches for Near-Earth Objects and covers between decl. $\delta = -75$ and $+65$ degrees (Drake et al. 2014). It started in 2004 and has three telescopes: the CSS and the Mount Lemmon Survey in Tucson, Arizona, and the Siding Spring Survey in Siding Spring, Australia. They set fields that tile their observed sky. Photometry is obtained using the aperture-photometry program SExtractor. In this work we use data from their first Data Release (CSDR1; Drake et al. 2014), which follows 198 million discrete sources monitored between 2005 April and 2011 June with an average of 250 observations per field and an exposure time of 30 seconds. More specifically, we use 47,000 objects from the Catalina Surveys Periodic Variable Star Catalog Drake et al. (2014) that were found to be periodic variables. An inspection was held by a single person for the labeling of 112,000 periodic candidates. It consisted of the examination and comparison of the phased time-series morphology with known types of periodic variables. The class distribution is shown in column "CSDR1" in Table 1.

6.2. SDSS Survey

The SDSS[7] (York et al. 2000) is an ongoing project that started its operation in 2000 and consists of three main stages: SDSS-I/II, SDSS-III, and SDSS-IV, each of them composed of multiple surveys. It is headquartered at Apache Point Observatory in southeast New Mexico and Las Campanas Observatory in northern Chile. Two main spectographs are

[6] https://catalina.lpl.arizona.edu
[7] https://www.sdss.org/

Table 1
Data Set Label Distribution

	CSDR1	Cross-match
EW	30,743	749 (764)
EA	4,683	148 (149)
beta Lyrae	279	⋯(7)
RRab	16,797	343 (346)
RRc	5,469	1193 (1219)
RRd	502	78 (81)
Blazkho	223	⋯(5)
RS CVn	1,522	43 (43)
ACEP	64	⋯(1)
Cep-II	124	⋯(3)
HADS	242	⋯(28)
LADS	7	⋯(⋯)
LPV	512	⋯(9)
ELL	143	⋯(4)
Hump	25	⋯(⋯)
PCEB	85	⋯(23)
EA_UP	155	⋯(1)
Total	61,575	2554 (2683)

Note. The second column shows time series from CSDR1. The third column shows unique objects in the cross-match between CSDR1 with SDSS DR14. The original number of objects is shown in parentheses, while the objects kept to build the data set used in our experiments are shown without parentheses. Labels that represented less than 1% are removed, together with objects that do not present spectrum features x_s. Labels are retrieved from Drake et al. (2014).

used in its surveys (Smee et al. 2013): the SDSS spectrograph[8] and the BOSS spectograph.[9] The SDSS spectrograph contains 640 fibers of 3″ of diameter per plate, it covers along 3800–9200 Å and has a resolution of 1500 at 3800 Å and 2500 at 9000 Å . On the other hand, the BOSS spectrograph contains 1000 fibers of of 2″ of diameter per plate, covers 3600–10,400 Å, and has a resolution of 1560–2270 in the blue channel and 1850–2650 in the red channel.

The data used in this work are taken from Data Release 14 (DR14), part of the fourth phase of SDSS (SDSS-IV; Blanton et al. 2017). All observations used the 2.5 m Sloan Foundation Telescope (Gunn et al. 2006). All of these spectra share the same wavelength grid spacing, but differ in the starting or ending point.[10]

6.3. Time Series and Spectrum Data Set

There are 2683 unique objects in the original cross-match of SDSS DR14 spectra and CSDR1 photometry shrunk to 2554 after initial data preprocessing. All objects have one time series but may have two spectra. Their labels are retrieved from Drake et al. (2014). The class distribution is shown in column "Cross-match" in Table 1. The original number per class is shown in parentheses beside the resulting number after discarding elements. Elements may be discarded either because their class represents less than 1% of the data set or because they do not have spectrum features x_s. A spectrum may not have spectrum features x_s if its wavelength coverage is shorter than required by the VAE input. The latter is better detailed in the following Section 6.4. The resulting cross-match data set used in our framework has objects from six different classes: equivalen width (EW; contact binary, 29.3%), EA (semidetached binary, 5.8%), RRab (fundamental mode RR Lyrae, 13.4%), RRc (first-overtone RR Lyrae, 46.7%), RRd (double-mode RR Lyrae, 3.1%), and RS CVn (RS Canum Venaticorum, 1.7%).

Table 2
Data Set Overview

	Spectra	Time Series	Cross-match
Survey	SDSS	CSS	CSS and SDSS
Data Release	All	DR1	DR1 and All
Source Type	Spectra	Photometry	Photometry and Spectra
Labeled	No	Yes	Yes
N^o of Time Series	⋯	2683	2683
N^o of Spectra	20,602 (20,949)	⋯	3296

6.4. Spectrum Data Set

For training the VAE for spectrum feature extraction, 20,949 spectra from SDSS DR14 are retrieved, as shown in Table 2. A VAE implemented with a fully connected neural network has a fixed input size, nevertheless, not all spectra have the same wavelength grid, as explained in Section 6.2. Hence we have to preprocess the spectra so that they all have a common wavelength grid. For this, we establish a starting and ending wavelength. Each is chosen following the value on the 99th percentile of the sorted starting and ending points from all spectra correspondingly. Any spectrum that has a higher starting point or a lower ending point is dismissed. We keep the same wavelength spacing because it is the same for all entries. The resulting starting and ending wavelength are 3830 Å and 9174 Å, correspondingly, with a grid of size 3794. The processed data set has 20,602 spectra, which is 1.7% less than the original one.

7. Implementation

The libraries used to implement our method are Tensorflow[11] and scikit-learn (Pedregosa et al. 2011). To train the VAE, we use a GPU GeForce GTX1080Ti, 11 GB, and the selected model takes 746 s. (12.43 minutes) to train. The RF with tenfold cross validation and GMM are trained with a 2.3 GHz dual-core seventh-generation Intel Core i5 processor and take 31.97 s. and 2.27 s, respectively. The sampling (with $N = 200$) and calculation of $\overline{H}_{ts}(\hat{y}|x_t)$ from Equation (8) and $\bar{l}_{ts}(x_t)$ from Equation (13) take 437 s. (7.17 minutes) using the latter hardware. All code is provided here.[12] Data can be found here.[13]

8. Results

This section describes the necessary components x_s, x_t, and $Q(x_t, x_s)$ to test our methodology and presents the results for the different strategies described in Section 5.4. Section 8.1 details

[8] http://classic.sdss.org/dr7/instruments/spectrographs/index.html
[9] https://www.sdss.org/instruments/boss_spectrograph/
[10] http://www.sdss3.org/dr9/spectro/spectro_basics.php
[11] https://www.tensorflow.org/
[12] https://github.com/jfastudillo/An-Information-Theory-Approach-On-deciding-Spectroscopic-follow-ups.git
[13] https://drive.google.com/drive/folders/1AVentdOhgknlfCAz8aWOOm_fLU3BuosS

the selected x_t and x_s, as described in 5.3. Section 8.2 describes the fitted joint distribution $Q(x_t, x_s)$. Finally, Section 8.3 presents a comparative study of the performance for multiple metrics for the different strategies.

8.1. Features

For the x_s features we train 15 VAE models, each with a different latent space dimensionality $d \in [1, 15]$, $z \in \mathbb{R}^d$, $x_s \subseteq \mu(z)$, as described in Section 5.3. The training set is the spectrum data set described in Section 6.4, composed of 20,602 spectra, as indicated in Table 2. The models are trained over 100 epochs with annealing[14] (Kirkpatrick et al. 1983) over the loss function, as done in Bowman et al. (2015). We preprocess spectra so that they have a common wavelength grid between 3830 and 9174 Å and normalize the flux to the [0, 1] range for each spectrum. We select the model with $\mu(z) \in \mathbb{R}^7$ that reports $R^2 = 0.96$ (coefficient of determination) over the test set between the original spectra and the reconstructed spectra, as explained in Section 5.3. The selected model consists of four encoding fully connected layers with ReLU (Glorot et al. 2011) activations with 2847, 1,900, 953, and 7 units, respectively. Symmetrically, the decoder has four layers with an output layer size of 3794 units and a sigmoid activation at the output.[15] The final model is depicted in Appendix A.1.

x_t is a subset of expert features (FATS; Nun et al. 2015), while x_s is built from learned features ($\mu(z)$), $x_t \subseteq$ FATS, and $x_s \subseteq \mu(z)$. They are jointly chosen as the smallest subset of the most important features according to an RF classification task, which achieves over 90% of the maximum improvement between the subsets with the lowest and highest accuracy. Further details are described in Section 5.3. To refer to each dimension of $\mu(z)$, we use the notation μ_i, $i = 0, \ldots, 6$. The data set used to select the x_t and x_s features is the cross-match data set (Table 2) detailed in Section 6.3. The selected features are $x_t = \{$PeriodLS, Freq1_harmonics_amplitude_0, MedianAbsDev, Q31, FluxPercentileRatioMid35, FluxPercentileRatioMid50, Freq1_harmonics_amplitude_1$\}$ and $x_s = \{\mu_2, \mu_5\}$. The cumulative relative importances (Gini impurity) of the selected features x_s, x_t, and $\{x_t, x_s\}$ according to an RF are 0.08, 0.39, and 0.46, respectively. The average accuracy of the RF trained over 10 stratified data folds using the selected $\{x_t, x_s\}$ features is 0.87, which equals to 93% of the maximum improvement over the accuracy between the worst selection of x_t and x_s (with the lowest accuracy equal to 0.73) and the best selection of x_t and x_s (with the highest accuracy equal to 0.88).

8.2. Joint Distribution

A joint distribution of the features x_s and x_t over the cross-match data Set (Table 2) is estimated using a GMM. As mentioned in Section 5.3, x_s by construction of the VAE follows a regular geometry (in this case, a single-mode Gaussian), but FATS need more modes to fully describe the distribution. Because of this, the number of clusters is set equal to the number of classes as an initial guess and the number of components adapts according to the data with a variational Bayesian method, as explained in Bishop (2006) and provided in the scikit-learn package (Pedregosa et al. 2011). The resulting GMM has the same number of components as the initial guess, which is six for this case.

8.3. Candidate Selection

To assign a query priority to the spectrum of each object in the data set, we propose three strategies ($IG(x_t)$; $IG(x_t) + \Delta\hat{y}(x_t)$; $H_t + \Delta\hat{y}(x_t)$) and compare them with two baseline strategies (*Random* and H_t) and the ideal scenario strategy. Each strategy assigns a priority to each of the objects and selects a subset of them for which to add spectrum features x_s to their available information. For any arbitrary subset size (s), the s first objects with the highest priority are selected for each strategy. Selected objects (in the subset) are classified according to their x_t and x_s features ($l_{ts}(x_t, x_s)$). Non-selected objects are classified using only x_t features ($l_t(x_t)$).

All strategies are detailed in Section 5.4. Our developed strategies are based on $IG(x_t)$ (Equation (10)), which prioritizes the objects with the highest approximate information gain and $\Delta\hat{y}(x_t)$ (Equation (13)), which prioritizes the objects that are most likely to change their classification if the spectrum features were added. The H_t baseline strategy prioritizes the objects with highest entropy on the time-series classification outcome, while *Random* gives random priorities to the objects. Ideal scenario selects the objects according to their actual information gain instead of their approximation $IG(x_t)$. Note that the latter is not a feasible strategy for selection, but it is included to study how the $IG(x_t)$ strategy would perform if the real spectra were used instead of the estimated spectra.

A comparative performance of the strategies is shown in Figure 3. We evaluate our method using three metrics and different selected subset sizes (s) (x-axis). The left plot shows the subset average improvement of the ground truth probability (GTP), i.e., the average probability assigned to the label y (the real class) for each object. Our $IG(x_t) + \Delta\hat{y}(x_t)$ strategy yields the highest GTP average improvement, especially for smaller s.[16] The next best strategy is $IG(x_t)$, which surpasses $H_t(x_t) + \Delta\hat{y}(x_t)$ and H_t especially within the first s ($\approx s < S_{510}$), and last, the *Random* strategy is the worst, with a constant average GTP improvement of 0.026. In this scenario, $\Delta\hat{y}(x_t)$ detects 113 objects that are most probable to have a change of classification prediction (with $\Delta\hat{y}(x_t) = 1$). Strategies that use the indicator $\Delta\hat{y}(x_t)$ lift up all objects with $\Delta\hat{y}(x_t) = 1$ to have the highest priority so that they are the first to be selected for spectra querying. In this way, the performance of the latter strategies in any regard is affected by this indicator only up to $s = 113$.

We note that it is more important to have a good performance at low s values rather than at high s values because we aim to have the largest improvement in classification prediction with the least querying for spectra. A close-up of the left plot in Figure 3 for the lower values of s is shown in the left plot of Figure 4. Here it is shown that $IG(x_t) + \Delta\hat{y}(x_t)$ has higher values for GTP mean improvement and also a higher gap with the baseline strategies for low s values compared to high s values. Take for example $s = 127$, which represents 5% of the data set. If the objects of this subset are selected with $IG(x_t) + \Delta\hat{y}(x_t)$, H_t, and *Random* strategies, it reaches a GTP

[14] When annealing a VAE, a variable weight is added to the term that pushes the encodings to follow a prior distribution (KL divergence) in the cost function at training time, which starts at 0 and progressively increases to 1 through training epochs. It is used so that the autoencoder first learns how to encode and decode correctly and then to better shape the distributions of the encodings.

[15] It is suitable to use a sigmoid for this case because inputs are normalized to the [0, 1] range.

[16] The ideal scenario does not compete because it is only a reference strategy and not a feasible one.

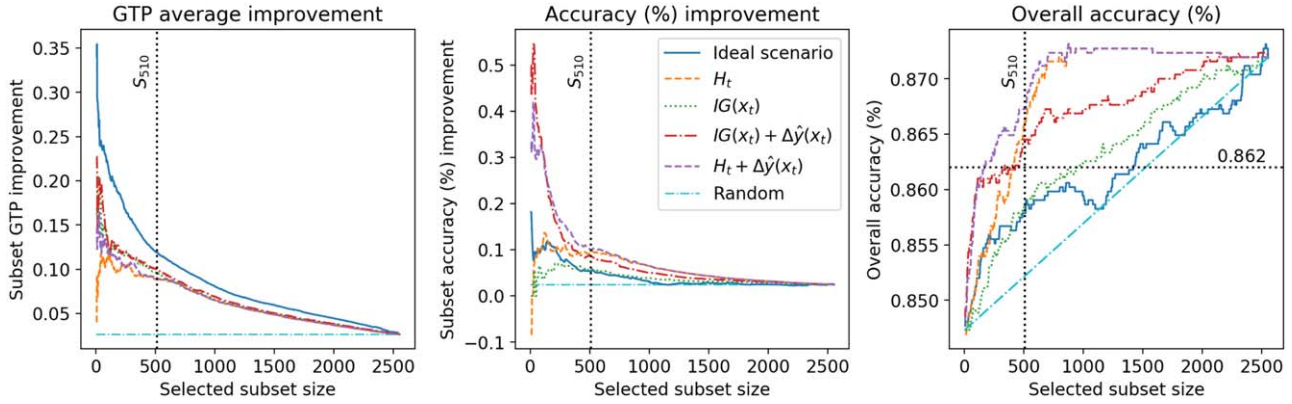

Figure 3. Comparative performance of our selection strategies ($IG(x_t)$; $IG(x_t) + \Delta\hat{y}(x_t)$; $H_t + \Delta\hat{y}(x_t)$) with baseline strategies (*Random*; H_t) and the ideal scenario strategy with respect to (left) the GTP average improvement on selected subset, (middle) accuracy (%) improvement on the selected subset, and (right) overall accuracy. S_{510} represents the subset with 510 objects (equal to 20% of the data set).

mean improvement of 0.13, 0.11, and 0.03, respectively. This means that $IG(x_t) + \Delta\hat{y}(x_t)$ improves the GTP 1.18 times as much as the H_t strategy and 4.33 times as much as the *Random* strategy. The same example but with $s = 510$ (20% of the data set) gives that $IG(x_t) + \Delta\hat{y}(x_t)$ improves the GTP of each selected object on average 1.11 times as much as the H_t strategy and 3.33 times as much as the *Random* strategy.

These results show that using estimated spectra for object selection leads to better results compared to not using them. Strategies that use $IG(x_t)$ could be further improved to be as good as the ideal scenario if we improve $Q(x_s|x_t)$ so that the estimation of the spectra is closer to the real spectra. Objects for which we can achieve good estimations of their spectra may not be worth spectrum querying, and hence we can save observational resources by not observing them. This is depicted in Figure 5, which shows the GTP improvement versus spectrum uncertainty for each object. To calculate the spectrum uncertainty for a given object with x_t features, we first sample the most likely spectra from the conditional distribution $Q(x_s|x_t)$, which is a GMM, as described in Section 8.2. Each sampled spectrum $\hat{x}_s$ has an assigned probability proportional to $Q(x_s = \hat{x}_s|x_t)$. Uncertainty then is assessed as the entropy of the sampled spectra $\hat{x}_{s,i} \sim Q(x_s|x_t)$, $i \in [1...N]$ for each object in the data set. We focus on objects with positive approximate information gain (Equation (10)), i.e., objects that have a higher priority assigned with the $IG(x_t)$ strategy. If the uncertainty is high, the GTP improvement has a wide range of positive and negative values, while if the uncertainty is low, the most likely GTP improvement is near 0 (no gain). Whenever we are fairly certain about an object spectrum, it is less likely to gain improvement in classification results and thus it is not worth querying for it. On the other hand, if we are widely uncertain of an object spectrum, we may gain a significant improvement in classification results if we query for its spectrum. In this way, it is worth improving our estimation of the spectra ($Q(x_s|x_t)$) so we can further save observational resources by not choosing objects of whose spectrum outcome we are already certain and instead choose objects of whose spectrum outcome we are unsure.

Complementary to the GTP average improvement over the selected subset, the right plot of Figure 4 shows the proportion of objects in the selected subset that improve their GTP in any positive amount, disregarding the average improvement. The best results are reached with $IG(x_t)$ and $IG(x_t) + \Delta\hat{y}(x_t)$. After these, $H_t + \Delta\hat{y}(x_t)$ and H_t stay competitive, and finally, *Random* presents the worst performance.

The middle plot of Figure 3 shows the accuracy (%) improvement of the classification of the selected subset (y-axis). If an arbitrary s is taken and it has an accuracy improvement of 0.3 with some strategy, it means that it has 30% more objects in the selected subset that are correctly classified using spectrum features ($l_{ts}(x_t, x_s) = y$) compared to not using them ($l_t(x_t) = y$), for this strategy selection method. As mentioned earlier in this subsection, $\Delta\hat{y}(x_t)$ signals 113 samples to prioritize first in the selection, and thus the performance of strategies that use this indicator is affected by this only up to $s = 113$. $H_t + \Delta\hat{y}(x_t)$ and $IG(x_t) + \Delta\hat{y}(x_t)$ are the best strategies for improving accuracy of the selected subset on low s values ($s < 510$). This is to be expected because $\Delta\hat{y}(x_t)$ causes first a selection of objects that are most likely to change their classification and hence improve accuracy. Next follows H_t. For higher values of s, $\Delta\hat{y}(x_t)$ no longer affects the selection, and any strategy that uses H_t is the best strategy. $IG(x_t)$ performs worse than the previous strategies for all s, but is still better than the *Random* strategy. The latter is to be expected because $IG(x_t)$ selects objects that will improve their classification confidence regardless of the accuracy improvement.

We note that the ideal scenario is worse than most strategies regarding accuracy improvement, even though it uses the real spectra instead of an estimation. To improve accuracy, we must select objects that are incorrectly predicted—false positives (FP)—with time-series features but are correctly predicted if spectrum features added, regardless of the amount of improvement of their GTP. The ideal scenario uses only the current value of information gain $IG(x_t)$, which selects objects that most likely increase their GTP but will not necessarily change their classification when spectrum information is added. This is the reason why the ideal scenario is worse at selecting objects that will improve accuracy compared to strategies that use $\Delta\hat{y}(x_t)$.

As mentioned before, $\Delta\hat{y}(x_t)$ returns the 113 (4% of the data set) objects that are most likely to have a change of classification prediction if x_s were queried. Within these objects, 46 change to their true class from a former incorrectly predicted class. The number of available incorrectly predicted —FP—objects in the data set is equal to 390, of which 101 can correct their prediction if spectrum features are added. $\Delta\hat{y}(x_t)$ detects most of the FP that can be corrected (46%) with less

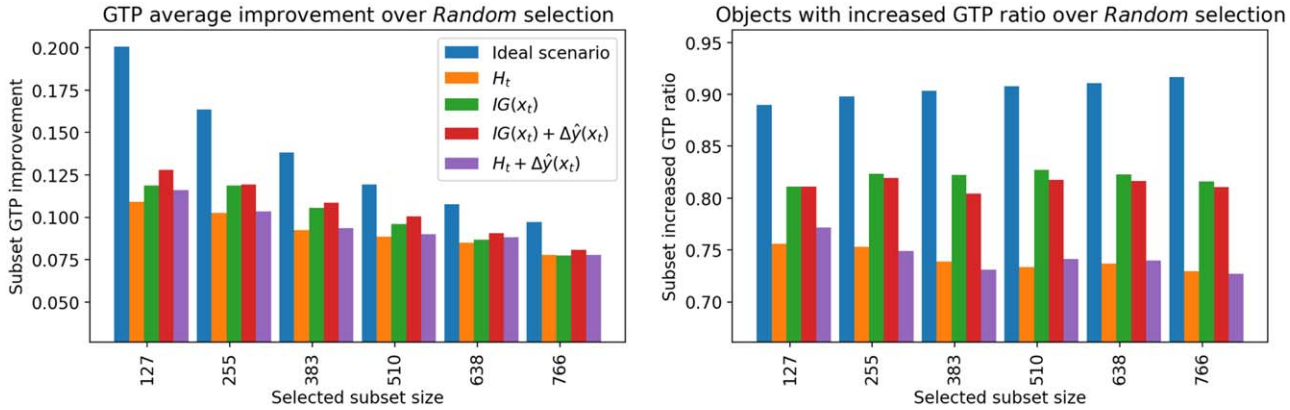

Figure 4. (left) Selected subset GTP average improvement and (right) ratio of objects in the selected subset with increased GTP for different strategies and subset sizes, over the *Random* selection strategy.

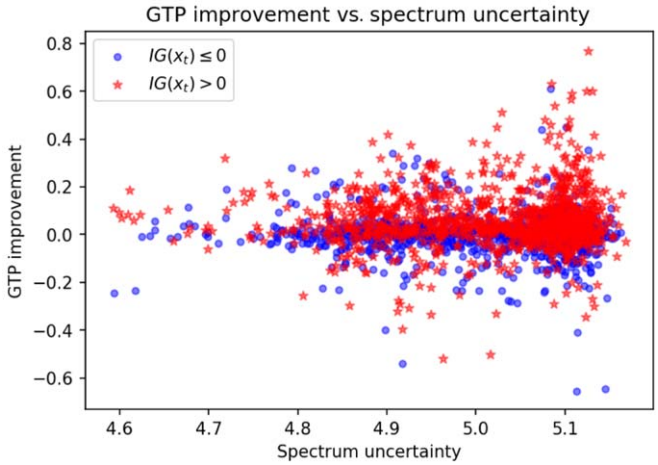

Figure 5. Ground truth probability (GTP) improvement vs. spectrum uncertainty. Uncertainty is assessed as the entropy of the sampled spectra for each object in the data set. Objects with positive approximate information gain (Equation (10)) are plotted as red stars, and uncertain objects are plotted as blue dots.

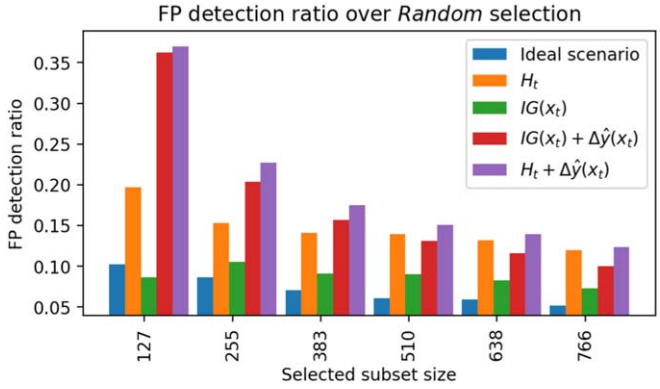

Figure 6. Comparison of the selected subset ratio that changes from a former incorrectly predicted—FP—class ($l_t(x_t) \neq y$) to their label y ($l_{ts}(x_t, x_s) = y$) for different strategies and subset sizes, over the *Random* selection.

than 4% of the data set. This means that $\Delta\hat{y}(x_t)$ is a good detector of incorrectly classified objects so that strategies that use it can quickly detect and correct FP objects (at least up until the number of objects with positive $\Delta\hat{y}(x_t)$).

The described FP object detection is depicted in Figure 6, which shows the proportion of objects from the selected subset that change from a former incorrectly predicted class ($l_t(x_t) \neq y$) to their label y ($l_{ts}(x_t, x_s) = y$). All strategies shows their results over *Random* selection, which has a constant FP detection ratio of 4%. Our method $H_t + \Delta\hat{y}(x_t)$ has higher values for FP detection ratio and also a wider gap with the other strategies for low s values compared to high s values. For $s = 127$ (5% of the data set), if the objects of this subset are selected with $H_t + \Delta\hat{y}(x_t)$, H_t, and *Random* strategies, it corrects the classification of 37%, 20%, and 4% of the selected objects, respectively. This means that $H_t + \Delta\hat{y}(x_t)$ corrects the classification of 17% and 33% more objects than H_t and *Random* strategies, correspondingly. For $s = 510$ (20% of the data set) we see that $H_t + \Delta\hat{y}(x_t)$ corrects the classification of 1% (20) and 11% more objects than H_t and *Random* strategies, respectively.

From Figures 3 and 6 we note that for high s values, strategies that use H_t detect FP objects better than strategies that do not. This means that after the $\Delta\hat{y}(x_t)$ indicator, which covers low s values, the most important selection criterion is the entropy of the time-series classification outcome $H_t(\hat{y}|x_t)$. This is more clearly depicted in the right plot of Figure 3, which shows the overall accuracy (%) of the classification of the data set for different subset sizes and strategies. For any arbitrary strategy and subset size, the objects that are selected are classified using time-series and spectrum features ($l_{ts}(x_t, x_s)$), while the non-selected objects are classified just with time-series features ($l_t(x_t)$). Our $IG(x_t) + \Delta\hat{y}(x_t)$ and $H_t + \Delta\hat{y}(x_t)$ strategies reach the same accuracies as the H_t and *Random* strategies with smaller selected subsets for all accuracies below 0.862. To reach an accuracy of 0.86, the $H_t + \Delta\hat{y}(x_t)$, H_t, and *Random* strategies need to query for the spectra of 101, 374, and 1315 objects, correspondingly. $H_t + \Delta\hat{y}(x_t)$ needs 27% and 8% the number of objects that the baselines strategies need to reach the same accuracy.

The presented results suggest that the best strategy to use depends on the task that it is meant for. If the aim is to improve the prediction probability (confidence), then $IG(x_t) + \Delta\hat{y}(x_t)$ is the best choice, while if it is to detect FP, then any method that uses $\Delta\hat{y}(x_t)$ is the best choice. In both cases our methodologies surpass baseline strategies (H_t and *Random*), especially within small subset sizes.

9. Conclusions

In this paper, we develop a general method for selecting astronomical objects for which taking their spectrum would improve our knowledge regarding their classification. Adding spectrum information provides further insights to time-series information, but requires more observational resources. Given the current and future development of wide-field surveys such as the LSST (Ivezić et al. 2008), it is valuable to know which objects we should prioritize to have a spectrum in addition to time series given the few spectroscopic facilities at hand. Differently from other works, such as Ishida et al. (2019), we make use of the full spectral information through automatic spectrum feature extraction instead of querying for labels. Additionally, we perform a multiclass classification of objects as opposed to related works, such as Peters et al. (2015) and Ishida et al. (2019), who performed a binary classification. As a byproduct of our work, we develop a model for the estimation of the spectrum of an object from its time series, which may be used in other applications. To validate our method, we perform extensive tests using a cross-match between spectra from SDSS DR14 (Blanton et al. 2017) and CSS DR1 (Drake et al. 2014). The cross-matched catalog is provided here.

We propose multiple selection strategies based on two metrics. The first metric is $IG(x_t)$ (information gain), which gives higher selection priority to the objects that are likely to improve their classification confidence if their spectra are queried. The second metric is $\Delta\hat{y}(x_t)$, which prioritizes the objects that are likely to change their classification if their spectra are queried. This metric ($\Delta\hat{y}(x_t)$) uses spectrum estimations to indicate the objects that will most likely change their classification. If the estimate is close enough to the real spectrum, then it is reasonable that if the object changes its classification with the spectrum estimates, it will most probably change with the real spectrum as well. We compare our strategies mainly with the H_t strategy, which gives high priority to the most uncertain objects on the time-series classification outcome, similar to common AL strategies such as reported by Ishida et al. (2019). Last, we also build strategies that mix H_t and our metrics for object selection.

From the results, $\Delta\hat{y}(x_t)$ detects most of the incorrectly predicted—FP—that can be corrected (46%) with less than 4% of the data set. This means that $\Delta\hat{y}(x_t)$ is a good detector of incorrectly classified objects so that strategies that use it can quickly detect and correct FP objects. Subsets of candidates selected using $\Delta\hat{y}(x_t)$ have a higher improvement on classification accuracy than all other strategies, especially when a small number of objects are selected for spectrum follow-up. If more objects are selected, then $\Delta\hat{y}(x_t)$ no longer affects the selection, and any strategy that uses H_t is the best selection strategy. Subsets of candidates selected using $IG(x_t)$ have a higher improvement on the GTP (probability assigned to the real class) than baseline strategies. This suggests that spectra querying may be used in addition to labeling to improve the classification confidence of selected objects and more broadly, the knowledge of these objects. Our developed information gain $IG(x_t)$ metric leads us to select objects that are likely to improve their GTP and avoids objects for which we are fairly certain of their spectrum outcome and are not likely to gain improvement in classification results. This metric can be further enhanced if the estimation of the spectrum from the time series becomes closer to the real spectrum. In this way, we could further save observational resources by not choosing objects for which we are already certain of their spectra outcome.

As future work, improvement of x_t features with unsupervised time-series feature learning could be included. Our method could be adapted for online selection so that the $Q(x_s|x_t)$ and P_{ts} may be trained alongside with the choice of newly selected objects. The joint space of time-series features and spectrum features can be modeled in a different way so that the estimation of the spectrum is closer to the real spectrum.

Additionally, automatic determination of the optimum number of objects to be queried could be developed so that when no significant improvement over any metric (accuracy and GTP for this case) is achieved, then no more spectra queries are made. Further metrics could be included to evaluate the improvement of our knowledge of the objects.

Different alternatives within the design of experiment may be explored. For example, finding the minimum required spectrum resolution that still adds information to the time series. Alternatively, multiple options for enhancing the available information of an object could be included (simultaneously). Some examples of this are adding more points to the time series, changing the exposure time of observations, or adding more color bands.

Finally, our methodology may be tested with other cross-matched catalogs such as the *Global Astrometric Interferometer for Astrophysics* (*GAIA*, Gaia Collaboration et al. 2018) and the SDSS (York et al. 2000). Additionally, we could apply our work to the recent Photometric LSST Astronomical Time Series Classification Challenge (PLAsTiCC, Kessler et al. 2019) to rank objects for spectrum follow-up and compare with current works related to it.

We acknowledge the support from CONICYT-Chile, through the FONDECYT Regular projects 1180054 and 1170305 and from the Chilean Ministry of Economy, Development, and Tourism's Millennium Science Initiative through grant IC12009, awarded to The Millennium Institute of Astrophysics. Also, this research is supported by the Computer Science Department at PUC Chile, through the Fond-DCC project.

Appendix
Appendix Material

A.1. VAN

Figure 7 depicts the selected VAE for spectrum feature extraction, implemented with a neural network architecture. The preprocessing includes the normalization of the flux of each spectrum independently of the [0, 1] range and the pruning of wavelengths in the tails so as to set a common wavelength grid of size equal to 3794. The selected model consists of four encoding fully connected layers with ReLU (Glorot et al. 2011) activations with 2847, 1900, 953, and 7 units, respectively. Symmetrically, the decoder has four layers with an output layer size of 3794 units and a sigmoid activation at the output.[17]

ORCID iDs

Javiera Astudillo https://orcid.org/0000-0002-6099-7071

[17] It is suitable to use a sigmoid for this case because inputs are normalized to the [0, 1] range.

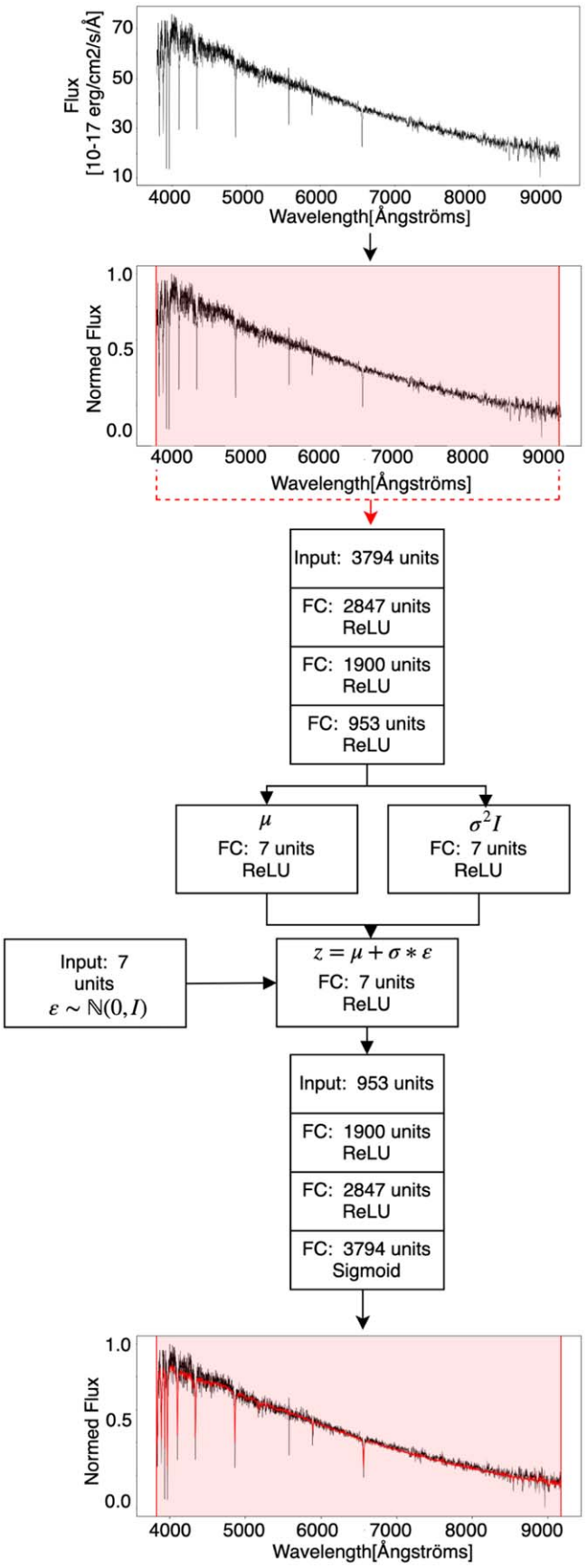

Figure 7. Selected variational autoencoder network diagram ($z \in \mathbb{R}^7$). At the output, the red curve corresponds to the output of the VAE (decoded spectrum), while the black curve corresponds to the original spectrum.

Pavlos Protopapas https://orcid.org/0000-0002-8178-8463
Karim Pichara https://orcid.org/0000-0002-9372-5574
Pablo Huijse https://orcid.org/0000-0003-3541-1697

References

Abbott, T. M. C., Abdalla, F. B., Allam, S., et al. 2018, ApJS, 239, 18
Astier, P., Guy, J., Regnault, N., et al. 2006, A&A, 447, 31
Bellman, R., Bellman, R., & Collection, K. M. R. 1961, Adaptive Control Processes: A Guided Tour (Princeton, NJ: Princeton Univ. Press)
Bernstein, J. P., Kessler, R., Kuhlmann, S., et al. 2012, ApJ, 753, 152
Bishop, C. M. 2006, Pattern Recognition and Machine Learning (Berlin: Springer)
Blanton, M. R., Bershady, M. A., Abolfathi, B., et al. 2017, AJ, 154, 28
Blei, D. M., Kucukelbir, A., & McAuliffe, J. D. 2016, J. Am. Stat. Assoc., 112, 859
Bloom, J. S., & Richards, J. W. 2012, in Advances in Machine Learning and Data Mining for Astronomy, ed. M. J. Way et al. (Boca Raton, FL: CRC Press), 89
Bowman, S. R., Vilnis, L., Vinyals, O., et al. 2015, Proc. 20th SIGNLL Conference on Computational Natural Language Learning, ed. S. Riezler & Y. Goldberg, (Berlin: Association for Computational Linguistics), 10
Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. 1984, Classification and Regression Trees (Monterey, CA: Wadsworth and Brooks)
Campbell, H., D'Andrea, C. B., Nichol, R. C., et al. 2013, ApJ, 763, 88
Castro, N., Protopapas, P., & Pichara, K. 2018, AJ, 155, 16
Chambers, K. C., Magnier, E. A., Metcalfe, N., et al. 2016, arXiv:1612.05560
Chung, J., Kastner, K., Dinh, L., et al. 2015, arXiv:1506.02216
Cohn, D. A., Ghahramani, Z., & Jordan, M. I. 1996, arXiv:cs/9603104
Dawson, K. S., Schlegel, D. J., Ahn, C. P., et al. 2013, AJ, 145, 10
Debosscher, J., Sarro, L. M., Aerts, C., et al. 2007, A&A, 475, 1159
Dieng, A. B., Kim, Y., Rush, A. M., & Blei, D. M. 2018, arXiv:1807.04863
Djorgovski, S. G., Drake, A. J., Mahabal, A. A., et al. 2011, arXiv:1102.5004
Djorgovski, S. G., Mahabal, A. A., Drake, A., Graham, M., & Donalek, C. 2013, in Planets, Stars and Stellar Systems, ed. T. D. Oswalt & H. E. Bond (Dordrecht: Springer), 223
Djorgovski, S. G., Mahabal, A. A., Drake, A. J., et al. 2012, in IAU Symp. 285, New Horizons in Time Domain Astronomy, ed. E. Griffin, R. Hanisch, & R. Seaman (Cambridge: Cambridge Univ. Press), 141
Drake, A. J., Graham, M. J., Djorgovski, S. G., et al. 2014, ApJS, 213, 9
Eilers, A.-C., Hennawi, J. F., & Davies, F. B. 2018, ApJ, 867, 30
Fabius, O., & van Amersfoort, J. R. 2014, arXiv:1412.6581
Fisher, R. 1935, The Design of Experiments (Edinburgh: Oliver and Boyd)
Freedman, W. L., Madore, B. F., Gibson, B. K., et al. 2001, ApJ, 553, 47
Frieman, J. A., Bassett, B., Becker, A., et al. 2008, AJ, 135, 338
Glorot, X., Bordes, A., & Bengio, Y. 2011, Proc. Fourteenth International Conf. on Artificial Intelligence and Statistics, ed. G. Gordon, D. Dunson, & M. Dudík, (Fort Lauderdale, FL: PMLR), 315
Gaia Collaboration, Brown, A. G. A., Vallenari, A., et al. 2018, A&A, 616, A1
Groenewegen, M. A. T. 2018, in IAU Symp. 330, Astrometry and Astrophysics in the Gaia Sky, ed. A. Recio-Blanco (Cambridge: Cambridge Univ. Press), 287
Gunn, J. E., Siegmund, W. A., Mannery, E. J., et al. 2006, AJ, 131, 2332
Hicken, M., Wood-Vasey, W. M., Blondin, S., et al. 2009, ApJ, 700, 1097
Hochreiter, S., & Schmidhuber, J. 1997, Neural Comput., 9, 1735
Ishida, E. E. O., Beck, R., González-Gaitán, S., et al. 2019, MNRAS, 483, 2
Ivezić, v., Tyson, J. A., Acosta, E., et al. 2008, arXiv:0805.2366
Jimenez Rezende, D., Mohamed, S., & Wierstra, D. 2014, arXiv:1401.4082
Kessler, R., Marriner, J., Childress, M., et al. 2015, AJ, 150, 172
Kessler, R., Narayan, G., Avelino, A., et al. 2019, PASP, 131, 094501
Kingma, D. P., & Welling, M. 2013, arXiv:1312.6114
Kinman, T., & Brown, W. 2010, AJ, 139, 2014
Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. 1983, Sci, 220, 671
Larson, S., Beshore, E., Hill, R., et al. 2003, BAAS, 35, 36.04
Lazio, J. 2009, in Proc. Panoramic Radio Astronomy: Wide-field 1–2GHz Research on Galaxy Evolution, ed. G. Heald & P. Serra (Trieste: Sissa Medialab), 58
Lee, H., Battle, A., Raina, R., & Ng, A. Y. 2006, in Proc. XIX Int. Conf. on Neural Information Processing Systems, NIPS'06, ed. B. Schölkopf, J. C. Platt, & T. Hoffman (Cambridge, MA: MIT Press), 801
Liang, D., Krishnan, R. G., Hoffman, M. D., & Jebara, T. 2018, arXiv:1802.05814
LSST, S. E. 2014, LSST Key System Parameters Summary https://ls.st/Document-16168
Luo, A. L., Zhao, Y. H., Zhao, G., et al. 2015, arXiv:1505.01570

Mackenzie, C., Pichara, K., & Protopapas, P. 2016, ApJ, 820, 138
Martínez-Palomera, J., Förster, F., Protopapas, P., et al. 2018, AJ, 156, 186
Masci, F. J., Laher, R. R., Rusholme, B., et al. 2019, PASP, 131, 018003
Massey, P., & Hanson, M. M. 2013, in Planets, Stars and Stellar Systems, ed. T. D. Oswalt & H. E. Bond (Dordrecht: Springer), 35
Miknaitis, G., Pignata, G., Rest, A., et al. 2007, ApJ, 666, 674
Minniti, D., Ramos, R. C., Zoccali, M., et al. 2016, ApJL, 830, L14
Naul, B., Bloom, J. S., Pérez, F., & van der Walt, S. 2018, NatAs, 2, 151
Nolan, L. A., Dunlop, J. S., Kukula, M. J., et al. 2001, MNRAS, 323, 308
Nun, I., Protopapas, P., Sim, B., et al. 2015, arXiv:1506.00010
Olshausen, B. A., & Field, D. J. 1996, Natur, 381, 6007
Pedregosa, F., Varoquaux, G., Gramfort, A., et al. 2011, Journal of Machine Learning Research, 12, 2825
Perlmutter, S., Aldering, G., Goldhaber, G., et al. 1999, ApJ, 517, 565
Perrett, K., Balam, D., Sullivan, M., et al. 2010, AJ, 140, 518
Perrin, G., Ridgway, S., Mennesson, B., et al. 2004, A&A, 426, 279
Peters, C. M., Richards, G. T., Myers, A. D., et al. 2015, ApJ, 811, 95
Pichara, K., & Protopapas, P. 2013, ApJ, 777, 83
Pichara, K., Protopapas, P., Kim, D. W., Marquette, J. B., & Tisserand, P. 2012, MNRAS, 427, 1284
Richards, J. W., Starr, D. L., Butler, N. R., et al. 2011, ApJ, 733, 10
Riess, A. G., Filippenko, A. V., Challis, P., et al. 1998, AJ, 116, 1009
Sako, M., Bassett, B., Becker, A., et al. 2008, AJ, 135, 348
Schmidt, K. B., Marshall, P. J., Rix, H.-W., et al. 2010, ApJ, 714, 1194
Scolnic, D. M., Jones, D. O., Rest, A., et al. 2018, ApJ, 859, 101
Shannon, C. E. 1948, The Bell System Technical Journal, 27, 379
Smee, S. A., Gunn, J. E., Uomoto, A., et al. 2013, AJ, 146, 32
Smith, R. M., Dekany, R. G., Bebek, C., et al. 2014, Proc. SPIE, 9147, 914779
Srivastava, N., Mansimov, E., & Salakhudinov, R. 2015, arXiv:1502.04681
Surace, J. A. 2015, IAUGA, 22, 2256381
Tanvir, N. R. 1999, in ASP Conf. Ser. 167, Harmonizing Cosmic Distance Scales in a Post-HIPPARCOS Era, ed. D. Egret & A. Heck (San Francisco, CA: ASP), 84
Vincent, P., Larochelle, H., Bengio, Y., & Manzagol, P.-A. 2008, in Proc. XXV Int. Conf. on Machine Learning, ICML '08, ed. W. Cohen, A. McCallum, & S. Roweis (New York: ACM), 1096
Wang, L., Schwing, A. G., & Lazebnik, S. 2017, arXiv:1711.07068
Witten, I. H., Frank, E., Hall, M. A., & Pal, C. J. 2016, Data Mining: Practical Machine Learning Tools and Techniques (San Mateo, CA: Morgan Kaufmann Publishers)
Xia, X., Protopapas, P., & Doshi-Velez, F. 2016, Proc. 2016 SIAM International Conference on Data Mining, ed. S. C. Venkatasubramanian & W. Meira, (Philadelphia, PA: SIAM), 477
Yang, J. J., Wang, X., Protopapas, P., & Bornn, L. 2015, arXiv:1501.02467
York, D., Adelman, J., Anderson, J., et al. 2000, AJ, 120, 1579