Title

Deep Generative Modeling of Periodic Variable Stars Using Physical Parameters

Journal

The Astronomical Journal, 164(6)

ISSN

0004-6256

Authors

Martínez-Palomera, Jorge

Bloom, Joshua S

Abrahams, Ellianna S

Publication Date

2022-12-01

DOI

10.3847/1538-3881/ac9b3f

The Astronomical Journal, 164:263 (14pp), 2022 December https://doi.org/10.3847/1538-3881/ac9b3f

Deep Generative Modeling of Periodic Variable Stars Using Physical Parameters

Jorge Martínez-Palomera[1], Joshua S. Bloom[1,2], and Ellianna S. Abrahams[1,3]
[1] Department of Astronomy, University of California, Berkeley, CA 94720-3411, USA; jorgemarpa@ug.uchile.cl
[2] Lawrence Berkeley National Laboratory, 1 Cyclotron Road, MS 50B-4206, Berkeley, CA 94720, USA
[3] Department of Statistics, University of California, Berkeley, CA 94720-3411, USA
Received 2020 May 14; revised 2022 September 27; accepted 2022 October 10; published 2022 November 30

Abstract

The ability to generate physically plausible ensembles of variable sources is critical to the optimization of time domain survey cadences and the training of classification models on data sets with few to no labels. Traditional data augmentation techniques expand training sets by reenvisioning observed exemplars, seeking to simulate observations of specific training sources under different (exogenous) conditions. Unlike fully theory-driven models, these approaches do not typically allow principled interpolation nor extrapolation. Moreover, the principal drawback of theory-driven models lies in the prohibitive computational cost of simulating source observables from ab initio parameters. In this work, we propose a computationally tractable machine learning approach to generate realistic light curves of periodic variables capable of integrating physical parameters and variability classes as inputs. Our deep generative model, inspired by the transparent latent space generative adversarial networks, uses a variational autoencoder (VAE) architecture with temporal convolutional network layers, trained using the OGLE-III optical light curves and physical characteristics (e.g., effective temperature and absolute magnitude) from Gaia DR2. A test using the temperature–shape relationship of RR Lyrae demonstrates the efficacy of our generative "physics-enhanced latent space VAE" (PELS-VAE) model. Such deep generative models, serving as nonlinear nonparametric emulators, present a novel tool for astronomers to create synthetic time series over arbitrary cadences.

Unified Astronomy Thesaurus concepts: Time domain astronomy (2109); Time series analysis (1916); Convolutional neural networks (1938); Periodic variable stars (1213)

1. Introduction

Robust and in-production automated image-based discovery on streaming survey data has matured significantly, from random forest-based methods (Bloom et al. 2012; Goldstein et al. 2015; Förster et al. 2016; Mahabal et al. 2019) to deep learning approaches (Sánchez et al. 2019). Nonetheless, to extract new knowledge in the time-domain era, the physical nature of the variability must be inferred. Retrospective classification (e.g., after each observing season or after survey completion) has shown great utility for the study of variable stars (e.g., Smolec 2005; Drake et al. 2013; Pietrukowicz et al. 2015). However, the scientific impact for ongoing time-domain surveys such as the Zwicky Transient Factory (ZTF, Bellm et al. 2018), the Vera Rubin Observatory (VRO-LSST, Ivezić et al. 2019), and the Wide Field Infrared Survey Telescope (WFIRST, Spergel et al. 2015) can only be maximized if additional follow-up resources are appropriately marshaled on scientifically relevant sources. Beyond its utility in broad demographic studies, once a source is classified, inference of the underlying physical state that dictates the observed variability (and any potential differences of that state from others in the same class) is often desirable.

Physical models of transient and variable stars provide, in principle, the most direct path to classification and the inference of the underlying physical state. As generative models—where the relevant initial conditions are fed forward through simulations to obtain the observables—these can be used to solve the inverse problem: the inference of the physical state from the observables. Physical models abound in certain time-domain subfields: e.g., gravitational-wave chirp signals from binary black hole mergers (Kumar et al. 2014); the Physics of Eclipsing Binaries (PHOEBE, Prša et al. 2016) for binary stars; SNANA (Kessler et al. 2009) software for supernova analysis; the Modular Open Source Fitter for Transients (MOSFiT, Guillochon et al. 2018) designed for transients interacting with circumstellar material such as tidal disruption events, kilonovae, Type II supernovae, and Type I superluminous supernova; and PyLIMA (Bachelet et al. 2017) for microlensing events. Wrapping physical models within a Bayesian inference framework, e.g., through Markov Chain Monte Carlo (MCMC) modeling, allows one to constrain the parameters of interest with the data.

Physical models, however, present several disadvantages. First, producing observables from ab initio parameters can be computationally expensive. A generative model that requires even a few seconds of wall-time computation can be prohibitively long when used as part of traditional MCMC inference. This challenge compounds when it is necessary to apply this approach to many sources. Second, current models do not include all physics (intrinsic and extrinsic), and the physical processes that are included are often approximated. As such, parameter inference with physical models is inherently imprecise. Last, physical models are often known to describe a subset of the transient and variable star dynamics. In the absence of a physical model, template fitting based on observed class exemplars may be used. For example, Sesar et al. (2010) produced templates of RR Lyrae light curves, spanning the range of the observed optical variability. Classification of a new suspected RR Lyrae source is then tantamount to a model selection process across the template bank of RR Lyrae subtypes.

As physical models and templates do not exist generally for the full diversity of the variable sky, classification requires a more data-driven approach. Here, classification is established as a supervised machine learning (ML) challenge, where existing data for a set of sources with known classes ("labels") are used to train an algorithm to predict class membership on new (unlabeled) sources. The efficiency of ML techniques had been largely demonstrated in providing robust classification of variable sources, either by using feature-based approaches (Richards et al. 2011, 2012; Pichara & Protopapas 2013; Lochner et al. 2016; Nun et al. 2016; Pichara et al. 2016; Martínez-Palomera et al. 2018) or by directly using the time-series data (Naul et al. 2018; Aguirre et al. 2019; Tsang & Schultz 2019; Jamal & Bloom 2020). These retrospective classification efforts[4] benefited from the use of highly curated training sets. One principal disadvantage of data-driven (as opposed to physics-driven) classification, however, is the need for a large set of training examples. As new surveys begin, no labeled real data exist with the depth and cadence of the survey.[5] Even after a survey has obtained data and sources are labeled, few, if any, of the minority subclasses may be observed and labeled, leading to a large class imbalance that alters the efficacy of classifiers in correctly identifying the (often more interesting) minority classes.

To expand the volume of examples in training sets, data augmentation is often employed (Dieleman et al. 2015; Cabrera-Vives et al. 2017; Martínez-Palomera et al. 2018; Boone 2019). This technique synthesizes new data by generating samples along observational axes believed to be extrinsic to the source itself. Through a series of simple transformations (e.g., rotation, translation, scaling, phase shifting) new instances are generated. Similarly, for observations with known noise properties, new data can be generated by bootstrap resampling the light curves from the training and/or test data sets (e.g., Naul et al. 2018). Although data augmentation provides a simple and fast path to increase training examples, the methodology expands upon only the known exemplars from the training data. Since the technique exploits a finite set of data, this data augmentation approach will not generally capture the full continuum of possible behavior within and between classes. That is, from a physical perspective, data augmentation does not afford a principled interpolation nor extrapolation in the way that physics-driven models can naturally accommodate.

Machine learning-based generative modeling, showing recent promise across different domains, provides a more natural framework for improving training set sizes that combines data augmentation techniques with the possibility of interpolation/extrapolation beyond the original training set. In the ML context, generative models refer to the approach of learning the joint distribution of low-dimensional (latent) random variables that describe the studied phenomena. Deep generative models (DGMs) refer to the use of deep neural network (NN) architectures for the learning and creation process. Multiple variants of DGMs are present in the literature—for a comprehensive review see Chapter 20 of Goodfellow et al. (2016)—such as variational autoencoders (VAEs, Kingma & Welling 2013) and generative adversarial networks (GANs, Goodfellow et al. 2014). Both have shown astonishing results in the image domain, where after training they are able to create realistic new images. Applications of DGMs in astronomy are numerous, and include the works by Tröster et al. (2019) that exploited both GAN and VAE models to map the large-scale gas distribution and temperature of *N*-body simulations; Ichinohe & Yamada (2019) trained a VAE for anomaly detection in X-ray spectroscopy data; Gabbard et al. (2019) implemented a conditional VAE to speed-up the Bayesian estimation of physical parameters of gravitational-wave progenitors; a GAN model was created for pulsar candidate classification (Guo et al. 2019); Mustafa et al. (2019) used a GAN that generates weak lensing convergence maps; and Yi et al. (2020) trained a VAE model to restore missing data in cosmic microwave background maps.

A major drawback in standard ML generative modeling lies in the limitation of interpolation if unconstrained by physical consideration: generated samples from a learned model may be acceptable visually but are nonetheless unbound to the physics. This shortfall constitutes the starting point for this paper: is it possible to connect the learned latent representation of a generative model with the characteristic/physical attributes of the training data to produce realistic samples that connect to our physical understanding of these sources?

Connecting intrinsic attributes to the latent space has been attempted in the image domain. Lample et al. (2017) trained an adversarial encoder–decoder architecture on the CelebA data set[6] to disentangle the latent space and the value attributes. The latter allow a user of the model to continuously control the parameters of a generated headshot sample. A similar idea was explored by S. Guan in his Transparent Latent-space GAN (TL-GAN),[7] where he paired a pre-trained GAN model with a pre-trained feature extraction model (both trained with CelebA data set) to then use a linear regression model to connect the latent space with the predicted features, allowing a smooth exploration of different feature axes (e.g., gender, age, hair type).

Generative models of variable stars capable of reproducing realistic time series can also be an important tool to explore and plan different observation cadences for future time-domain surveys (e.g., VRO-LSST). To optimize cadence strategies, figures of merit must be intercompared with a broad diversity of simulated time-domain sources/events. The Photometric LSST Astronomical Time-Series Classification Challenge (PLAsTiCC, The PLAsTiCC team et al. 2018) generated about 3.5×10^6 light curves, simulated using current physical models and class templates (Kessler et al. 2019). Despite the known diversity of galactic variable sources (Gaia Collaboration et al. 2019), the data set of PLAsTiCC variable stars consisted of just five classes (RR Lyrae, eclipsing binaries, Miras, microlensing events, and M-dwarf flares). A rational explanation for this is that these are the classes for which reliable physical models and templates exist. The absence of other periodic variables, such as Cepheids and δ Scuti, represents the lack of precise physical models that can generate realistic time series. Therefore, this opens a window to

[4] In contrast, automated streaming machine-learning classification is relatively new: Muthukrishna et al. (2019); Carrasco-Davis et al. (2019); Zorich et al. (2020); and ALeRCE http://alerce.science/.

[5] In the ML context, transfer learning can help address this problem by learning a model using one data set and predicting in a different domain. See Zhang (2019) for an extensive review and Benavente et al. (2017) for time-domain astronomy applications.

[6] CelebFaces Attributes Dataset (CelebA) is a large-scale face attribute data set with more than 200,000 celebrity images, each having 40 attribute annotations such as male/female, hair color and length, presence of eyeglasses or hats, nose shapes, and smile. http://mmlab.ie.cuhk.edu.hk/projects/CelebA.html.

[7] https://blog.insightdatascience.com/generating-custom-photorealistic-faces-using-ai-d170b1b59255

Table 1
OGLE-III Light Curves and Gaia DR2 Stellar Parameters

Variable	Total Light Curves				$T_{\rm eff}$		R and L	
	Original[a]	Clean	Validated	Augmented	Validated	Augmented	Validated	Augmented
(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)
ACEP	83	72	71	5000	1 (1.4%)	70 (1.4%)	– (0%)	– (0%)
CEP	8052	7265	7121	10,045	4931 (69.3%)	6934 (69.0%)	5 (0.1%)	6 (0.1%)
DSCT	2596	44	42	5090	32 (76.2%)	3840 (75.4%)	10 (23.8%)	1166 (22.9%)
ECL	419,868	10,002	9505	10,000	8581 (90.3%)	9035 (90.4%)	1495 (15.7%)	1556 (15.6%)
ELL	25,217	2328	2269	10,365	1908 (84.1%)	8720 (84.1%)	135 (5.9%)	603 (5.8%)
LPV	343,596	4460	4349	10,044	3730 (85.8%)	8638 (86.0%)	6 (0.1%)	15 (0.1%)
RRLYR	44,031	10,028	9322	10,169	2814 (30.2%)	3062 (30.1%)	31 (0.3%)	31 (0.3%)
T2CEP	599	450	436	5047	322 (73.8%)	3746 (74.2%)	3 (0.7%)	32 (0.6%)
Total	844,042	58,049	33,114	65,760	22,319 (67.4%)	44,045 (70.0%)	1685 (5.1%)	3409 (5.2%)

Note. The original number of light curves available from OGLE-III database is shown in column (2). Columns (3) and (4) give the number of sources after the cleaning and cross-match validation (see the Appendix for details), respectively. Column (5) shows the number of sources after augmentation. Columns (6) and (7) show the number of sources with $T_{\rm eff}$ values (percentages) for the validated cross-matches and augmented data set, respectively. Similarly, columns (8) and (9) show the stellar radius and luminosity.
[a] Includes all variability subtypes.

the use of state-of-the-art deep learning algorithms to provide fast data-driven nonparametric generative models.

Inspired by such challenges faced in massive surveys and the need to expand the representation across variability classes, here we propose a DGM based in a VAE architecture to simulate new irregularly sampled light curves using physically relevant parameters as input variables. In particular, we train a conditional VAE (cVAE) using OGLE-III light curves for a total of eight different variability classes. The cVAE model is constructed in two parts: the encoder, which compresses the input time series and metadata into a low-dimensional latent vector enclosing the relevant information of each source; and the decoder, which uses the latent code and metadata (the conditional) to reconstruct the original time series. With this design, the model learns the underlying distribution behind the generative process and therefore is able to create new observations by sampling from the learned distributions. The conditional information provided to the model consists of the variability class, physical parameters, and the time-stamp of each observation. We explore the connection between the latent and physical space by means of different regression models. Thus, during the evaluation of the model, the user can specify a set of class label, physical parameters, and observation cadence as inputs to the decoder to generate a new realistic light curve.

This paper is structured as follows. Section 2 describes data selection. Section 3 presents the artificial neural network architecture and training procedures. Section 4 discusses the results of the selected generative models. Section 5 presents our conclusions and further prospects. We include an Appendix that provides a detailed overview of the extensive validation process in our cross-match results. Alongside this paper, the scripts, trained models, and validated training data set used for the analysis shown in this work are available online[8]; copies of these (Version 0.1.1) files were also deposited to Zenodo (Martínez-Palomera 2022).

2. Data

In order to train a DGM that can generate time series of variable sources using physical parameters of the sources as input, we require (a) the light curves of previously classified variable sources, and (b) a catalog of relevant physical parameters for these objects, e.g., stellar radius, metallicity, effective temperature. In this section, we describe both data sources, as well as data preprocessing.

2.1. Time Series

We construct our training and testing data sets from The Optical Gravitational Lensing Experiment (OGLE, Udalski et al. 1992) in its third phase (OGLE-III, Udalski et al. 2008).[9] The I-band observed light curves were collected from the Galactic Bulge, Galactic Disk, and the Large and Small Magellanic Clouds fields and describe eight variability types: anomalous Cepheids (ACEP), classical Cepheids (CEP), δ Scuti (DSCT), eclipsing binaries (ECL), ellipsoidal variables (ELL), long-period variables (LPV), RR Lyrae (RRLYR), and Type II Cepheids (T2CEP). Table 1 column (2) summarizes the total number of light curves available in the data set, as well as the number counts per variability class.

After a visual inspection of the available light curves, we decided to keep only time series with sufficient variability signal. This was assessed by calculating the signal-to-noise ratio (S/N) between the variability amplitude and the mean photometric uncertainty. We removed light curves with $S/N < 5$, and performed three iterations of 3σ clipping filter over the magnitude and uncertainty values in order to remove outlier observations. Some of the variability classes have subtypes with multiple pulsation modes and/or are semiregular pulsators, e.g., DSCT-MULTIMODE and LPV-SRV (where SRVs are semiregular variables); this results in amorphous shapes in the period-folded (phase) space. We opted to drop these subtypes in order to maintain only variability subtypes that define a regular shape in their phase-folded light curve. The subtypes that most impact the volume of our data set are LPVs, SRVs, OGLE small-amplitude red giants (OSARGs), and carbon-rich (C) and oxygen-rich (O) variables.

In order to train our proposed NN (see Section 3), we required all light curves to have the same number of observations. After analyzing the distributions of light-curve lengths of OGLE-III data, we decided to set the sequence

[8] https://github.com/jorgemarpa/PELS-VAE

[9] http://ogle.astrouw.edu.pl/main/collections.html

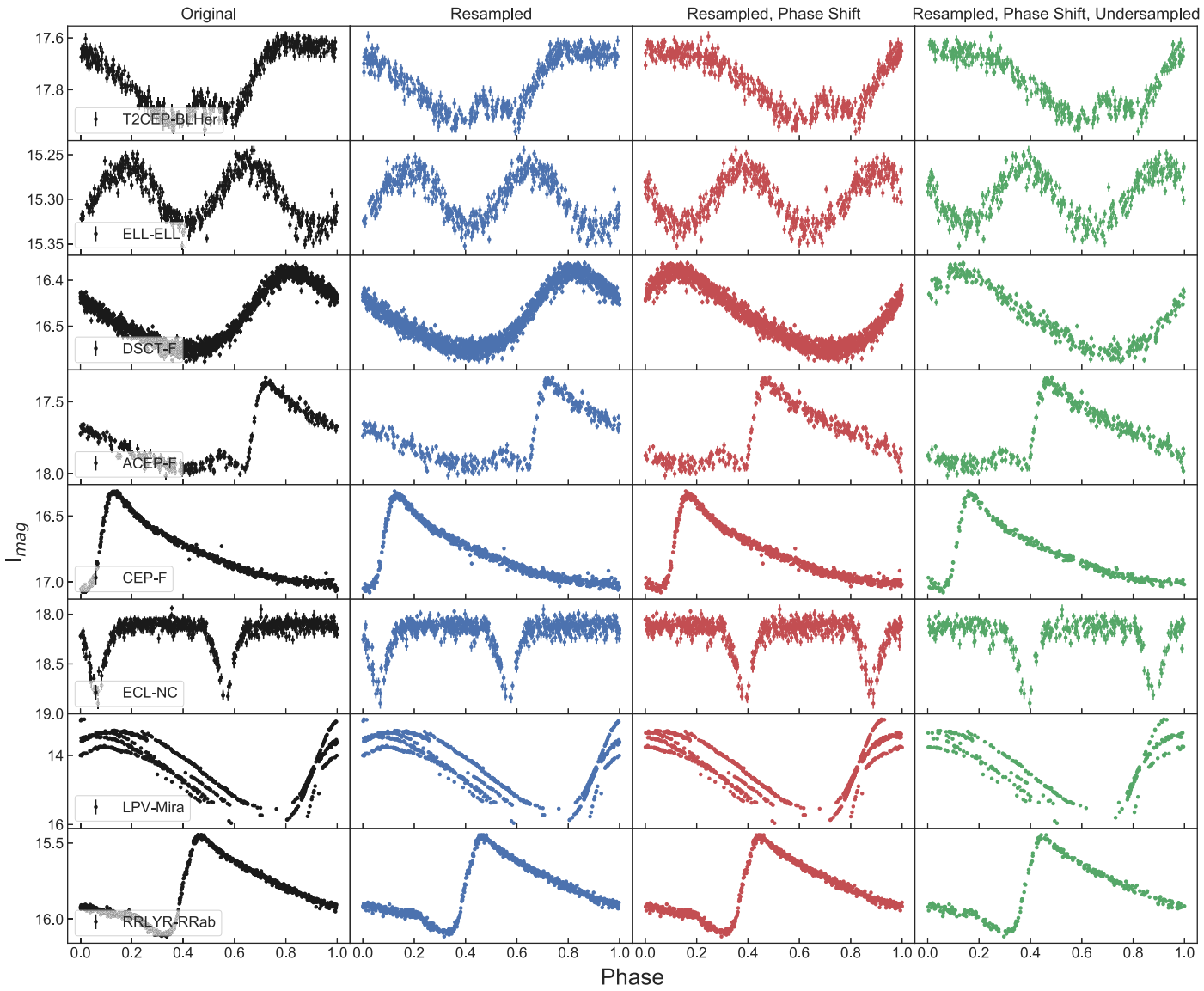

Figure 1. Examples of light curves from the OGLE-III survey. Eight different variability classes are shown in each row. In column 1 the original phase-folded light curves are shown; column 2 shows the resampled light curves, column 3 shows the phase-shifted light curve, and column 4 depicts the undersampling to $t_{\rm len} = 300$ observations.

length to $t_{\rm len} = 300$ observations. We phase-folded every time series using the periods reported by OGLE-III studies and then we randomly undersample to $t_{\rm len}$ data points per light curve. All light curves were normalized using a min–max scaler; this puts all time series in the range [0, 1] and removes the interclass brightness dependence, allowing our network to focus on learning the light-curve features that are relevant to each variability class. The total number of counts after the preprocessing step is presented in Table 1 column (3); examples of light curves are shown in the first column of Figure 1.

2.2. Data Augmentation

As seen in Table 1 (column (3)), the distributions of variability classes reflect a clear imbalance, with a conspicuous lack of sources in the DECT, ACEP, and T2CEP classes. In order to compensate for this, we artificially augmented the data set. For a given phase-folded untrimmed light curve we resampled the photometric measurements following a Gaussian distribution with mean and variance corresponding to the photometric magnitude value and its associated uncertainty, respectively. Then we applied a phase shift sampled from a [0, 1] uniform distribution. Finally, we randomly undersampled the new light curve to $t_{\rm len}$ observations. We performed these steps for a random selection of sources in each variability class to reach a uniform count of data per class. The total number of time series per class after data augmentation is shown in column (5) of Table 1. For classes with more than 2000 examples, we augmented the number counts up to ∼10,000, for the rest to ∼5000. In the case of ECL, where the initial number of sources is noticeably larger than for other types, we reduced the data set to ∼10,000 examples prioritizing sources with physical parameters (see Section 2.3). Figure 1 shows light curves for different variability classes, as well as the result of the three steps followed in our augmentation procedure. This demonstrates that the important characteristics of each light curve such as shape, amplitude, and photometric statistics are reasonably preserved.

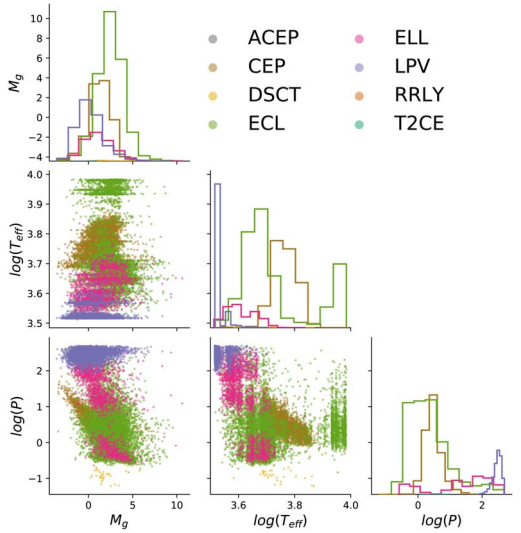

Figure 2. Joint distributions of physical parameters—period P, effective temperature $T_{\rm eff}$, and absolute g-band magnitude M_g—used during training of the generative model (see Section 4.2) color-coded by variability class.

2.3. Physical Parameters

To incorporate stellar parameters in our model we cross-matched the OGLE-III variability catalog with the Gaia Data Release 2 (Gaia DR2, Gaia Collaboration et al. 2016, 2018) using a 2″ radius search. We followed a rigorous set of steps in order to validate the cross-matched sources (more details in the Appendix), which include compensating for proper motion, comparison of Gaia and OGLE variability classification, and positioning of variables in the color–magnitude diagram. This provides stellar parameters for a fraction of the data set of light curves, such as effective temperature $T_{\rm eff}$, stellar radius R, and luminosity L. Table 1 shows the number of sources with stellar parameters of all eight variability classes. We would like to highlight that a sizeable fraction (38%) of our data set is associated with measurements of effective temperatures, whereas few measurements of stellar radii and luminosities from Gaia are made available. The Gaia pipeline only provides R and L for less than half of the sources with temperature estimates due to post-processing filtering (Andrae et al. 2018). Due to the lack of sources with stellar luminosity and radius, we choose to exclude these physical parameters in our training process. The Gaia catalog also provides color values based in the blue and red passbands ($G_{\rm BP} - G_{\rm RP}$) and parallaxes (Gaia Collaboration et al. 2019); absolute g-band magnitudes $M_{\rm G}$ were calculated using distances derived by Bailer-Jones et al. (2018). Figure 2 shows the joint distribution of the three physical parameters (period, $M_{\rm G}$, and $T_{\rm eff}$) used during model training; see Section 4.2 for details.

3. Neural Network Model and Training

3.1. Network Architecture

We used the VAE (Kingma & Welling 2013) architecture as our deep generative model of choice. A VAE provides a probabilistic approach for calculating a compressed representation of a set of observations. A VAE is described by two components. First, an encoder stage transforms training data into a low-dimensional representation in the so-called latent space. Then, a decoder processes the latent representation and expands it in order to reconstruct the original data. In a VAE, the encoder output describes a probability distribution for each latent dimension, instead of a deterministic representation as in the case of classic autoencoders (Hinton & Salakhutdinov 2006). The dimensionality of the latent space is a hyperparameter of the model to tune. This probability is assumed to be normally distributed and the encoder predicts its mean and variance. Later, a latent vector z is sampled from the learned distribution and fed into the decoder using the reparameterization trick:

$$z = \mu + \sigma \odot \epsilon \tag{1}$$

where μ and σ describe the probability distribution returned by the encoder, and ϵ is sampled from a unit Gaussian distribution. This allows backpropagation to be performed during the training phase.

Our encoder–decoder architecture consists of two types of layers for each module—a temporal layer processing the sequential nature of the data, and fully connected layers for outputs. Figure 3 shows an overview of the VAE network architecture. The temporal component can be implemented as either a temporal convolutional network (TCN, Bai et al. 2018) or a recurrent neural network (RNN).

TCNs refers to a family of 1D convolutional architectures designed for efficiently handling sequential data. The main features of TCNs are (1) causal convolutions, meaning that there is no information leakage from future to past; (2) dilated convolutions, the equivalent of adding a step between each pair of adjacent filters to allow for a large receptive field (Yu & Koltun 2015) and an extensive lookback time; and (3) residual connections, where the output of each residual block is constructed by adding the input data and the transformed data layers (see Figure 2 in He et al. 2016). TCNs are described by the following hyperparameters: the convolution kernel size ($k_{\rm size}$), the number of hidden units ($h_{\rm size}$) in the convolutional layers, the dilation of convolution (d), and the number of temporal blocks ($n_{\rm blocks}$).

Alternatively, RNNs are recursive architectures that combine operations per cell (time step) in order to calculate a cell state and output. Such cell states are carried into the next cell (time step) and contain the relevant historic information learned. RNNs suffer from several well-known issues such as short-term memory and vanishing gradient problem during training (Bengio et al. 1994; Pascanu et al. 2012). There are variants of RNN architectures designed to prevent such limitations. The most widely used are long–short term memory (LSTM, Hochreiter & Schmidhuber 1997) and gated recurrent units (GRUs, Cho et al. 2014), which proved to be able to prevent the gradient-vanishing and explosion typically noted in traditional RNNs by including an internal mechanism called gates to regulate the flow of information. The network size is controlled by the number of hidden units per cell ($h_{\rm size}$) and the number of stacked RNNs ($n_{\rm layers}$), where the output of each cell is fed into the next cell in the same RNN layer but also to the corresponding cell in the next RNN layer.

After the sequential layer(s) in the encoder we include stacks of fully connected layers followed by an ReLU[10] activation and dropout layer.[11] The network then connects to two independent

[10] A rectified linear unit (ReLU) function is defined as $f(x) = \max(0, x)$.

[11] Dropout is a regularization technique and refers to the process of randomly deactivating neurons during training in order to avoid overfitting. The number of dropped neurons per layer is defined by a probability that is a hyperparameter of the model.

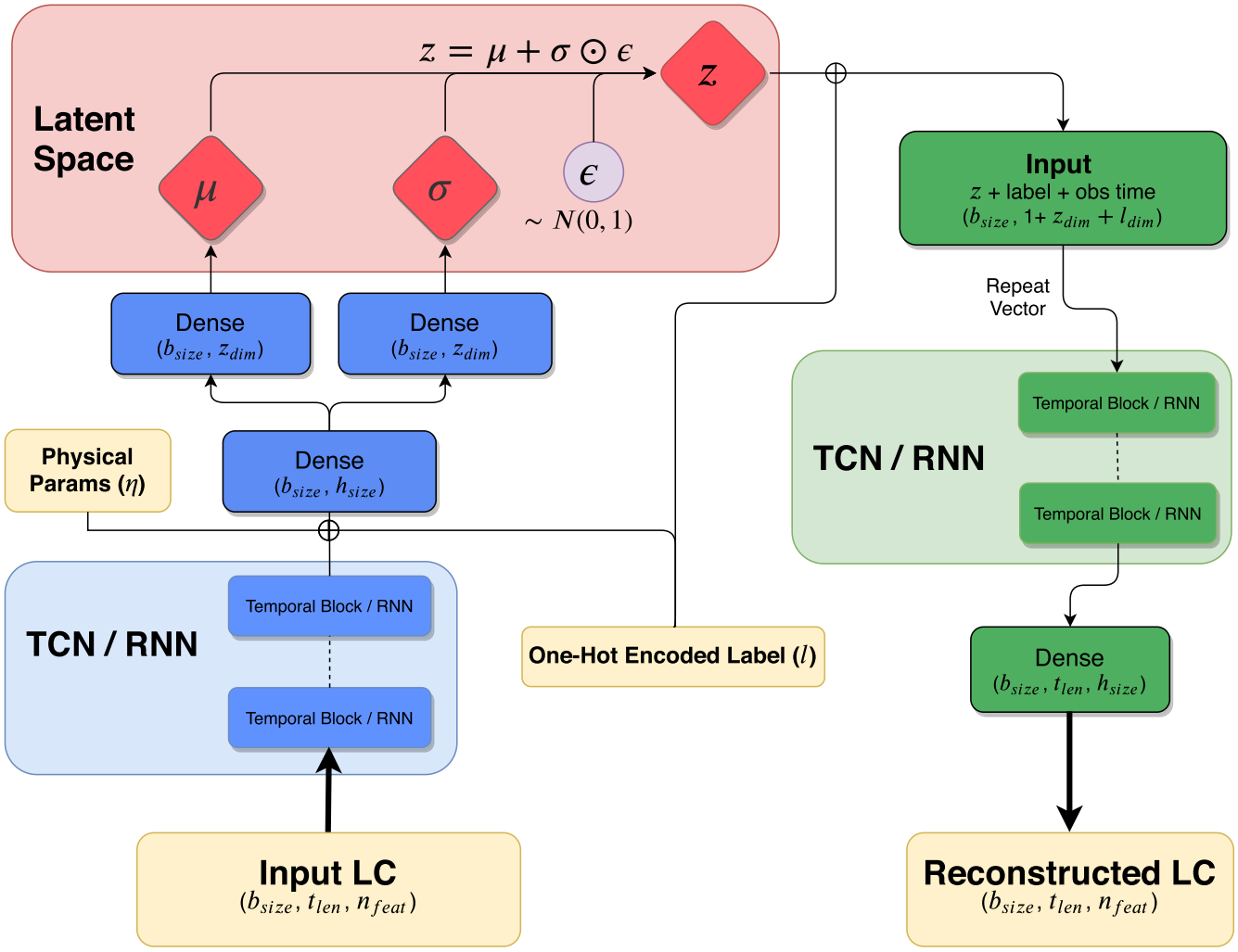

Figure 3. Neural network architecture of a conditional variational autoencoder. The left-hand side of the diagram represents the encoder (blue boxes), while the right-hand side represents the decoder (green boxes). Red boxes show the latent space. Each side, encoder and decoder, uses a sequential architecture (TCN or RNN) and a fully connected dense layer to map the outputs. Yellow boxes represent input data, such as light curves (LC), variability labels (l), and the corresponding physical parameters (η).

fully connected layers, one to predict the mean and the other the log-variance of the n-dimensional Gaussian distributions of the latent space.

For the decoder, a sequential network (TCN or RNN) receives as input a repeated vector of the latent code and metadata reshaped according to the number of time steps $t_{\rm len}$. Each time step is tagged with the corresponding observed difference in time $\Delta t_i = t_i - t_{i-1}$. This sequential network uses the same architecture and hyperparameters as the encoder. After that, a fully connected layer followed by a sigmoid[12] activation function returns the reconstructed scaled light curve.

The flow of data through the VAE network (see Figure 3 for reference, where the arrows represent the flow of data) is as follows: scaled light curves are first fed into the encoder, leading to the extraction of representative features. Then the last time-step state is concatenated with a one-hot encoding of the label value and the physical parameters. Next, the stack of fully connected layers is branched into two dense layers that predict the mean and log-variance of the latent space distributions. Later, a new latent vector is sampled and concatenated with the observation times of the light curve (as in Naul et al. 2018) and with the encoded labels; this vector is repeated $t_{\rm len}$ times and presented to the sequential component of the decoder; finally a fully connected layer processes the sequential output and returns the magnitude and error of the reconstructed light curve. Our VAE model accepts nonuniformly sampled time series and is time-conditioned, and therefore only reconstructs the photometric measurements. Moreover, due to the inclusion of side information into the network, the latent variables not only encode the relevant features extracted from the light curves, but also embed the provided metadata, which enforces a correlation between the latent space and the physical parameters that can be exploited after training. We call this architecture the "physics-enhanced latent space VAE" (PELS-VAE) model.

3.2. Training

Let x be the observed training data points, z the latent vector, $q_\theta(z|x)$ the encoder network with θ model parameters, and $p_\phi(x|z)$ the decoder network with ϕ model parameters, then the

[12] A sigmoid function is defined as $f(x) = (1 + e^{-x})^{-1}$ and constrains output values to the range [0, 1].

Table 2
Notation for Neural Network Hyperparameters and Grid Search

Parameter	Description	Grid Search[a]
$b_{\rm size}$	Batch size	[32, 64, **128**]
lr	Learning rate	$\sim U(0.00005, 0.1)$, **0.001**
$lr_{\rm sch}$	Learning rate scheduler	[none, exponential, **cosine**, plateau]
β	KL divergence weight	$\sim U(0, 1)$, **0.75**
$z_{\rm dim}$	Latent space dimension	[**4**, 6, 8, 12]
$p_{\rm drop}$	Dropout probability	$\sim U(0, 0.5)$, **0.2**
$n_{\rm blocks}$	Number of temporal blocks in TCN	[5, **7**, 9]
$k_{\rm size}$	TCN kernel size	[3, 5, **7**, 9]
d	Dilation in TCN	**2**
seq_arch	Sequential architecture	**TCN**, GRU, LSTM
$h_{\rm size}$	Number of hidden units in TCN/RNN	[16, 32, **48**, 64]
$n_{\rm layers}$	Number of RNN layers	[1, **2**, 3]

Note.
[a] Hyperparameter grid. In bold are highlighted the values associated with the best-performing model.

classical VAE objective function is

$$\mathcal{L} = \mathbb{E}_{z\sim q_\theta(z|x)}[\log p_\phi(x|z)] - D_{\rm KL}(q_\theta(z|x)||p(z)) \quad (2)$$

where the first term is the reconstruction likelihood of the decoder network given a latent vector; the expectation value is taken with respect to the encoder's distribution over the representations. The second term is a regularization, the Kullback–Leibler divergence (KL, Kullback & Leibler 1951) between the learned latent distribution and its prior, which is assumed to be the unit Gaussian $p(z) \equiv \mathcal{N}(0, \boldsymbol{I})$, with $\boldsymbol{I}$ the identity matrix with dimensions corresponding to the size of the latent space. This loss function equally treats the reconstruction error and the similarity of the latent representation with a unit Gaussian. The latter intends to capture the underlying data generative factors, enforce that similar data points have a similar latent representation, and aim for a disentangled representation, meaning that every single latent direction controls a single aspect of the generative factor. One way to enforce disentanglement in the latent space would be to introduce an additional hyperparameter (β) that weights the importance of the second term in Equation (2) as follows:

$$\mathcal{L} = \mathbb{E}_{q(z|x)}[\log p(x|z)] - \beta D_{\rm KL}(q(z|x)||p(z)). \quad (3)$$

Introduced by Burgess et al. (2018), the hyperparameter β plays a role in disentangling the latent representation. Higher values of β enforce orthogonality between latent directions due to the assumption of a diagonal covariance matrix in its prior distribution. With $\beta = 0$, the traditional autoencoder loss is recovered. We used a slightly modified version of the empirical expression for Equation (3) when $p(z) \equiv \mathcal{N}(0, \boldsymbol{I})$:

$$\begin{aligned}\mathcal{L} = &\frac{1}{t_{\rm len} N}\sum_{i=0}^{N}\sum_{j=0}^{t_{\rm len}}\left(\frac{x_i^j - \hat{x}_i^j}{\sigma_i^j}\right)^2 \\ &- \beta\sum_{i=0}^{N}(\sigma_i^2 + \mu_i^2 - \log(\sigma_i) - 1) \\ &+ D_{\rm KL}(\hat{\sigma}_i^j||\sigma_i^j) \end{aligned} \quad (4)$$

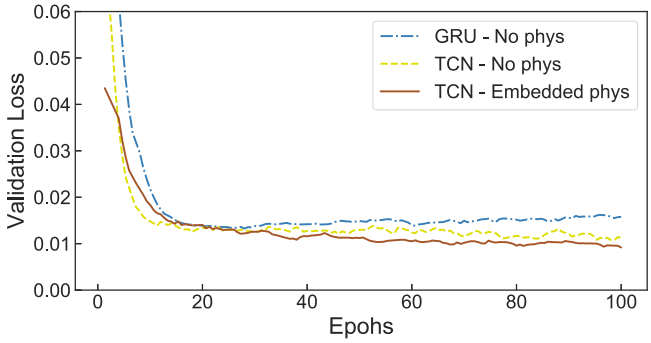

Figure 4. Validation loss during training epochs for VAE models without physical parameters and GRU (blue) and TCN (yellow) architectures, while the red line shows a model with TCN layers that include physical parameters during training.

where N is the total number of light curves, x_i^j, $\hat{x}_i^j$, σ_i^j, and $\hat{\sigma}_i^j$ are the jth measurements, reconstruction values, measurement error, and reconstructed errors of the ith light curve, respectively; μ_i and σ_i refer to the mean and dispersion the latent distributions for the ith light curve. The first term corresponds to the weighted mean squared error that represents the reconstruction error, and the second term refers to the KL divergence. We added a third regularization term that enforces the proper reconstruction of predicted measurement errors by calculating the KL divergence between the true and predicted values. This last term regularizes that the probability distribution of reconstructed errors $\hat{\sigma}_i^j$ follows the true distribution of σ_i^j.

We partition our data set into three subsamples, the training (60%), validation (20%), and test (20%) sets. We followed a stratified split strategy to ensure that class proportions are preserved for each partition. The test set only contains real sources that were not used during data augmentation. To search for the set of hyperparameters of the best-performing model, we run a hyperparameter sweep and optimization using the Weight & Biases[13] framework. We used a Bayesian optimization search strategy provided by this framework that employs a Gaussian process to model the hyperparameter function and then chooses parameters that improve the probability of minimizing a specific metric, which in our case was the loss function for the validation set. The hyperparameter search covered different combinations and is summarized in Table 2.

After the sweep search, we found the set of optimized hyperparameters highlighted in Table 2. We treated the latent space dimension as a model's hyperparameter and optimized it for best loss performance, which makes it dependent on the model and data set. The best latent dimensions are 4 and 6, with insignificant differences in their loss performance, but with higher correlation coefficients between embeddings for the latter. Therefore, a four-dimensional latent space is sufficient to encapsulate the necessary information to then fully reconstruct the original time series, while still keeping a low-dimensional space that can be correlated to a low-dimensional physical parameter space a posteriori. We did not find a significant difference between the best configuration of GRU and TCN in terms of reconstructed light curves and latent space properties but found a reduced convergence time in training for TCNs, which were at least three times faster than GRUs, even though

[13] https://www.wandb.com/

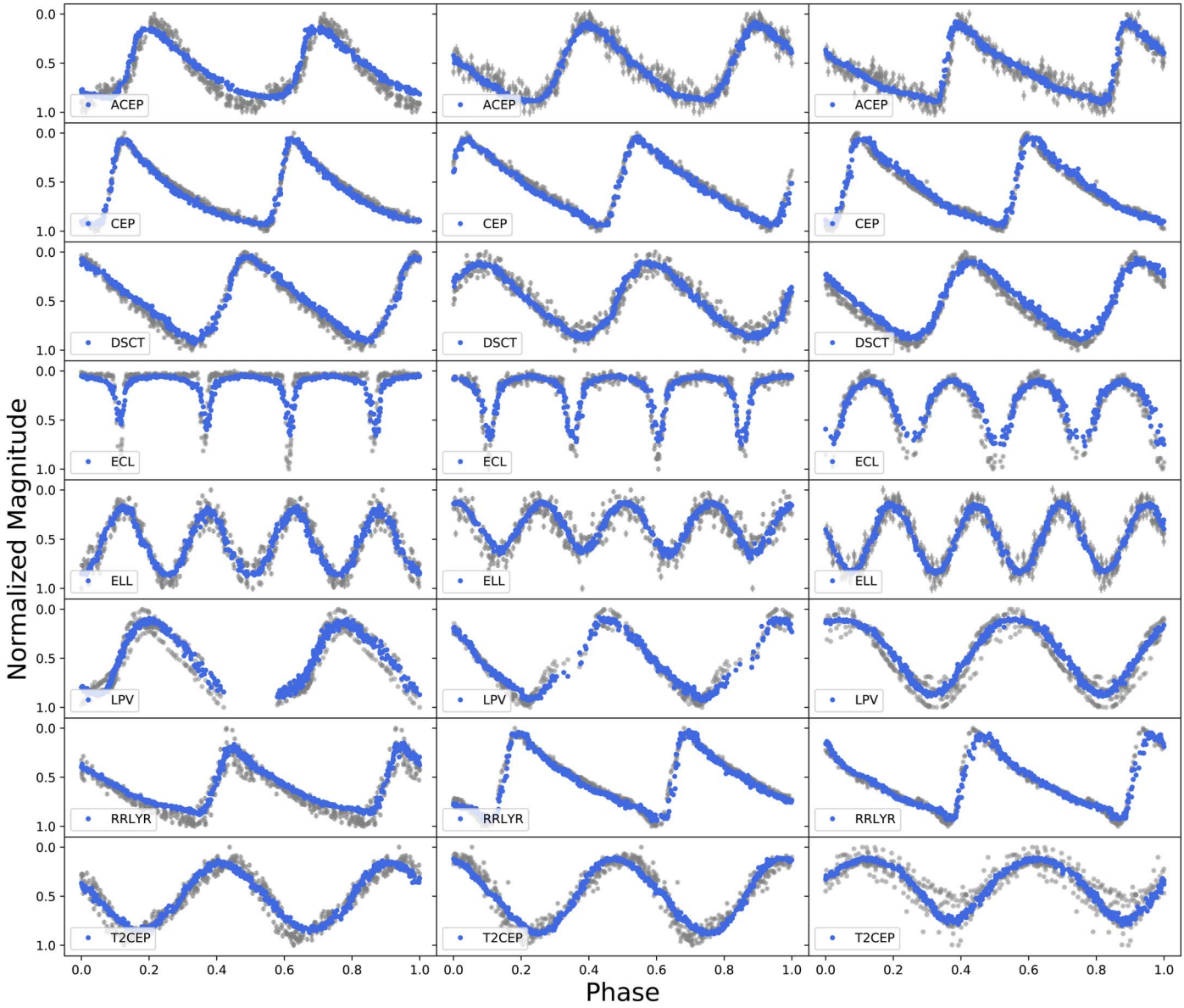

Figure 5. Displays of reconstructed phase-folded light curves obtained at the decoder level by the best-performing model trained only with time series (cVAE). Three examples per variability class are shown. Gray markers denote the observed photometric OGLE-III *I*-band light curve, while in blue are the decoder reconstructions.

the TCN network capacity was six times larger. This is consistent with the recent findings in Jamal & Bloom (2020).

We used the ADAM optimizer (Kingma & Ba 2014) during training for over 100 epochs. Training and testing loss values are shown in Figure 4, where convergence is shown to be achieved. Our models were implemented using `Pytorch 1.3` (Paszke et al. 2019).

4. Results

4.1. Light-curve Reconstruction

First, we explore the basic form of our cVAE model, trained with light-curve data and conditioned only with variability labels, excluding physical parameters, in order to explore the capabilities of the model to capture the necessary information to reconstruct light curves of periodic variable sources. Figure 5 shows reconstructed light curves from the selected best model, with three examples for each variability class. The overall shape and small details characteristic of each variability class are recovered by the generative model. Due to the variational nature of the model, stochasticity introduced when sampling from the latent variables, the reconstructed light curves are not completely equal to their original counterparts. As expected, the model is optimized to learn a smooth latent space that facilitates the generative process rather than the reconstruction.

Figure 6 shows the joint distributions of all four latent dimensions, particularly the predicted mean values (μ) that describe the Gaussian distribution of the latent space. Due to the regularization term added to the objective function, the KL divergence term in Equation (2), the learned latent space resembles a normal distribution in each dimension. The clustering of different variability classes is not strong, due to this regularization, which drives toward a smooth and dense latent space. The latter is particularly useful when generating new instances, especially when interpolating between different loci of the latent space that were not explored during training.

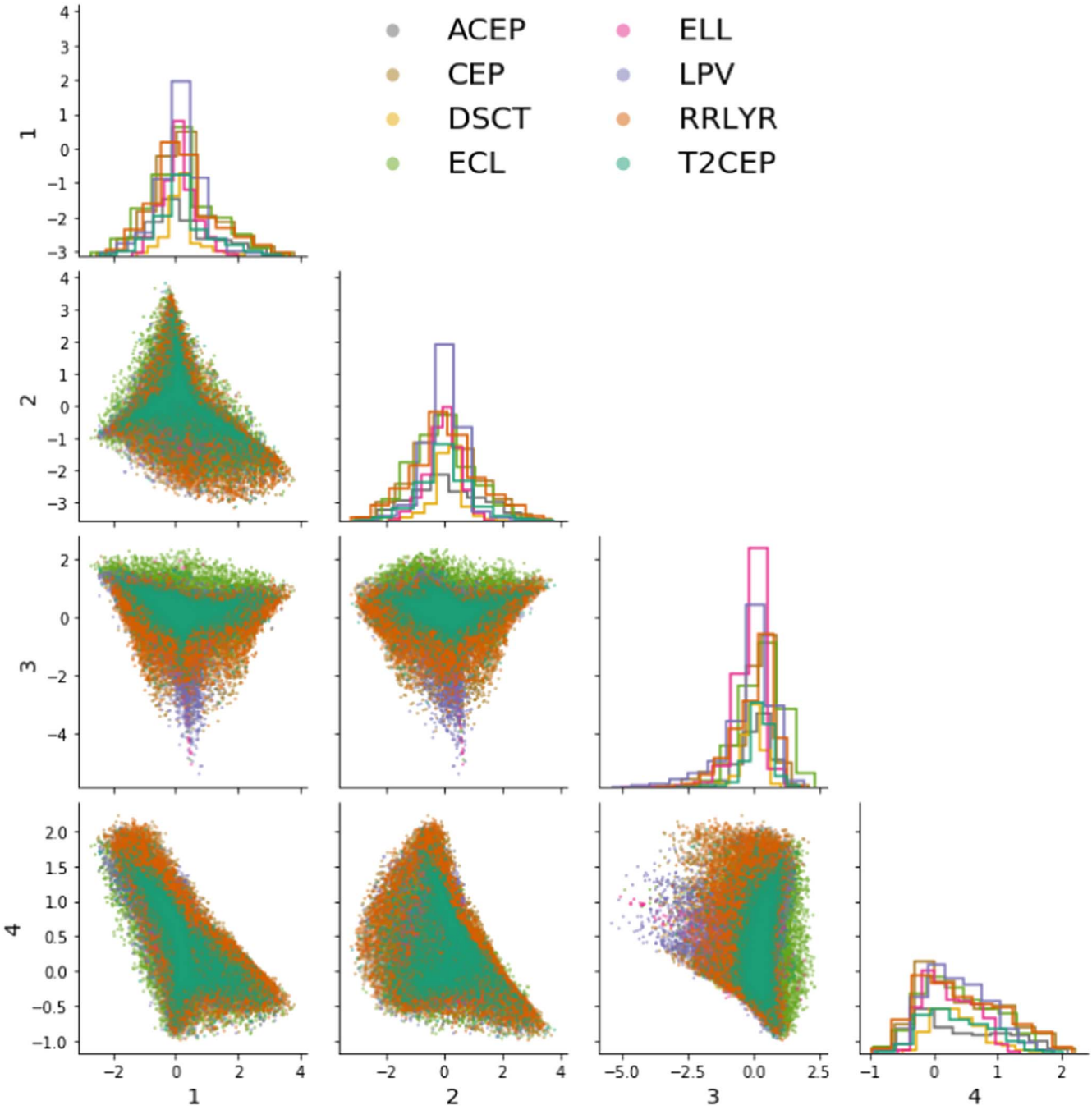

Figure 6. Joint distributions of all μ (encoded features) values for the four dimensions of the latent space obtained by the best-performing model trained only with time-series data (cVAE). Color coding corresponds to the eight variability types.

4.2. Embedding Physical Parameters

Using the same network architecture described in Figure 3, we trained a second model (cVAE-P) that includes three physical parameters: effective temperature, absolute magnitude, and period. Figure 7 shows the reconstructed light curves.

In order to compare the capability of our model to learn and encode physical parameters in the latent space, we compared two instances of the same model: (1) a cVAE network trained with light curves and labels, but without physical parameters as discussed in the previous section; (2) a similar network architecture (cVAE-P) that also includes physical parameters during training, concatenated to the label vector and provided to the encoder component of the network. After both models were trained, we establish a relation between the latent space and the physical parameters by fitting a multivariate regression between them. This allows us to select a given set of $T_{\rm eff}$, period, and absolute magnitude that are mapped to the latent space and later fed to the decoder in order to generate new light curves.

We evaluate three regression models:[14] linear, random forest (RF regressor), and a basic multilayer perceptron (MLP). All three regressors are fitted using the same training set, and the rms errors (RMSE) of the validation set (20% of the total data set) for each method are presented in Table 3. Both the linear

[14] We used the `scikit-learn` (Pedregosa et al. 2011) implementation for all three regression models.

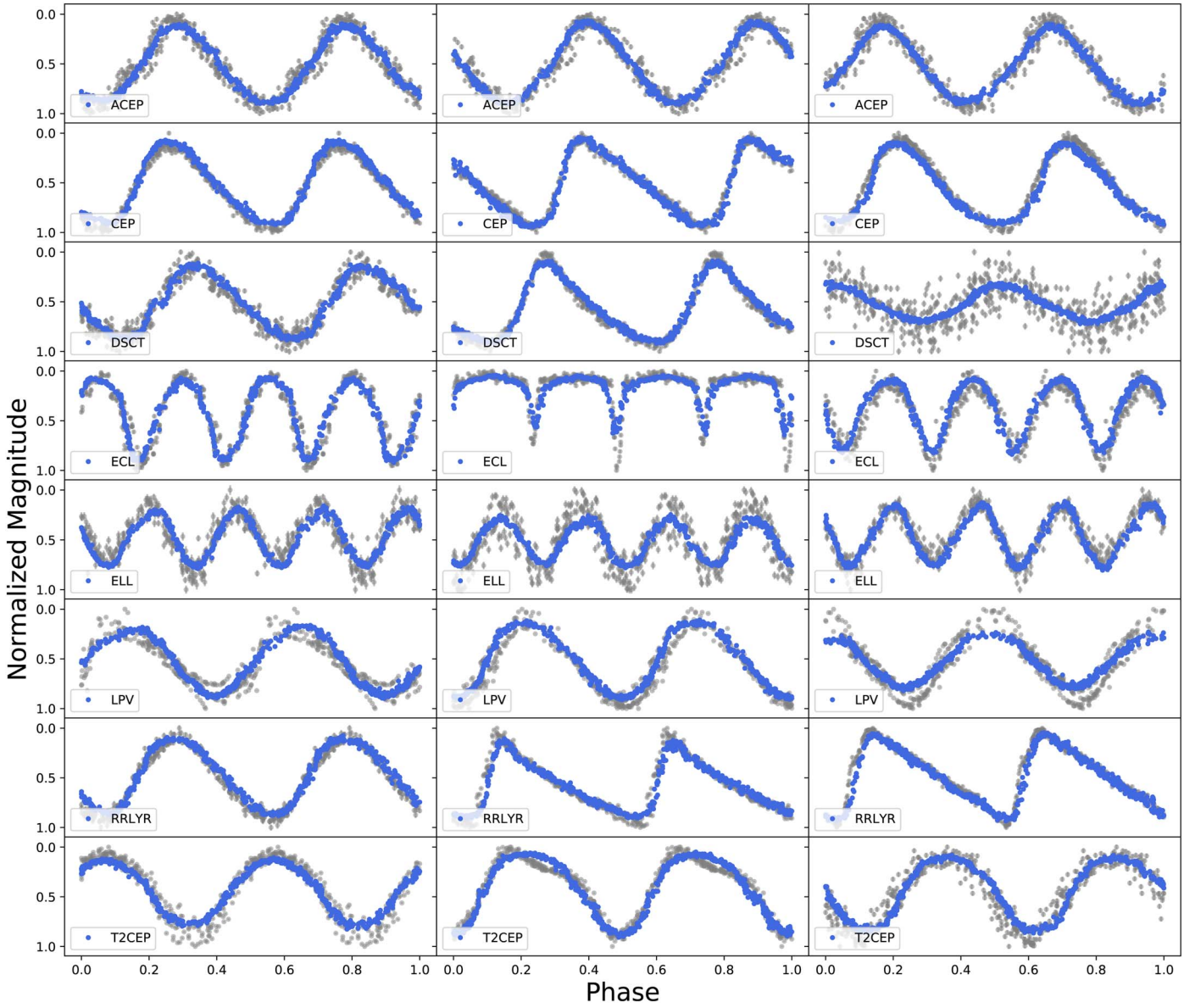

Figure 7. Reconstructed light curves obtained at the decoder level by the best-performing network (cVAE-P) exploiting physical parameters as auxiliary inputs. Displays of three objects per variability class are shown. Gray markers denote the observed photometric OGLE-III *I*-band light curve, and in blue are the decoder reconstructions.

and MLP regressions achieve similar RMSE, as expected, while the random forest regressor outperforms the others. Though the RF achieves a lower RMSE, tree-based regressors are restricted to predictions within the training set range. When comparing the two generative models, with and without seeing physical parameters during training, RMSE values are not substantially different but are consistently better for the model that includes physical parameters (cVAE-P).

In order to keep the variational power of our generative model and avoid obtaining the exact copy of the light curve when selecting a fixed vector of physical parameters, we added an extra "dummy" dimension to the physical space. Afterwards, the regression model is fitted with a collection of 100 repeated physical vectors per instance in the data set that only differs in the value of the extra dimension, which is sampled from a uniform distribution. In the latent space, thanks to the variational architecture that encodes the parameters μ and σ of the latent distributions, each latent vector is sampled 10 times from $\sim N(\mu, \sigma)$ for each instance of the data set. This allows the model to generate slightly different time series for the same set of physical vectors while keeping them consistent.

Table 3
Latent–Physical Space Regression

Generative Model	cVAE	cVAE-P
Linear	0.863	0.794
Random forest	**0.299**	**0.289**
Multilayer perceptron	0.863	0.798

Note. Values correspond to the rms error for a validation set. Values in bold are the best achieved for each generative model.

4.3. Generating New Light Curves

The process of generating a new light curve is described as follows: a vector of physical parameters is constructed given a

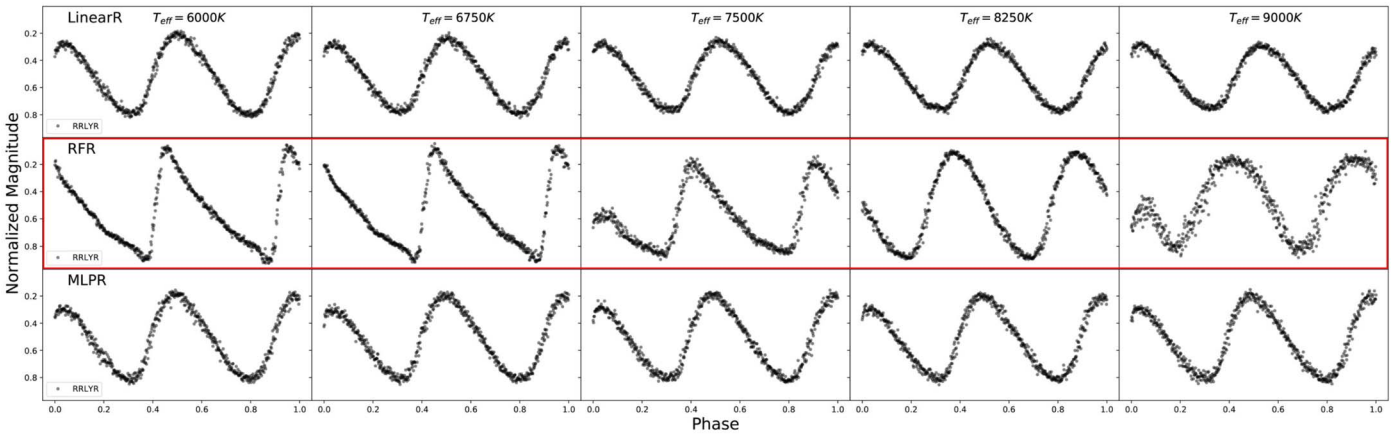

Figure 8. Generated light curves of RR Lyrae stars as effective temperature $T_{\rm eff}$ increases (column direction). The three rows show the results from the three regression models: linear, random forest, and multilayer perceptron. The RF regressor, second row, shows a better representation along the temperature series for RR Lyrae variables, which agrees with its comparative RMSE performance.

set of values for effective temperature, absolute magnitude, and period; an extra dimension is added by sampling from $\sim U(0, 1)$; this vector is projected into the latent space by means of the regression function, which returns a latent vector; this latent vector is tagged with the user-defined observed time-stamps calculated from the period and a zero-time value; this extended latent code is fed into the decoder network, which returns a new phase-folded light curve, where phase values can be converted to time using the previously selected period and zero-time. With a model conditioned to the observed phases, it is possible to change the effective observational cadence of the light curve. This provides an opportunity to explicitly explore different observing cadences and how such cadences might impact the discovery and characterization potential of different variability classes.

Figure 8 presents a sequence of generated RR Lyrae light curves for different values of effective temperature as they increase in value. RR Lyrae light curves morph in shape with increasing temperature, transitioning from a sawtooth shape characteristic of Bailey type *ab* to a more sinusoidal shape typically found in hotter Bailey type *c*. This change in the light-curve shape is clearly shown when using the RF regressor (middle row), but there is minimal change in the shape for the other two regressors.

4.4. Limitations and Future Explorations

Incorporating a fixed number of physical parameters in learning the generative model limits its capacity to use other stellar parameters without retraining. On the other hand, the model trained only on light curves provides for the possibility of including post facto additional (physical) variables that were not explored in this study. This can done by fitting a new regression model to connect the latent space with the new space of physical parameters, avoiding the retraining of the VAE model. For instance, adding metallicity, stellar mass, and surface gravity of the stars will provide a more complete generative model. The challenge here is to obtain a comprehensive training catalog of variable stars with respective stellar parameters, across a variety of variability classes. Our VAE models and physical-to-latent space mapping do not explicitly include the Gaia measurement uncertainties, particularly for $T_{\rm eff}$; this introduces a form of label noise that will require further exploration.

A future extension of this generative model would consist of finding a more complex and accurate connection between the latent space and the space of physical variables. Recent work has used flow-based models coupled to VAE models: Böhm et al. (2019) found, in an image-based domain, that by training a normalizing flow (NF, Jimenez Rezende & Mohamed 2015) to the encoded data distribution the sample quality improved when generating new images; both models provided competitive results with very little hyperparameter tuning. Moreover, the NF model also allows sampling from a true normal distribution and then mapping to the latent distribution, which, while regularized to be Gaussian, in practice does not strictly follow a Gaussian distribution. An important characteristic of flow-based models is that the transformations are invertible (bijectors), by construction, allowing one to solve the inverse problem of inferring physical parameters from the latent code. Even more, a flow-based model could capture better the covariance between physical parameters to find an even more physically constrained mapping to the latent space. The latter is specifically important for our purposes as each type of variable star tends to occupy a specific locus in the low-dimensional manifold of physical space. For instance, RR Lyrae are located in the intersection of the horizontal branch and the instability strip of the H-R diagram; this bounds the physically allowed values of effective temperature and luminosity to the [6000–7250] K range and $\sim 10^2\ L_{\odot}$, respectively.

5. Summary

To date, the most prominent uses of deep generative models have been in the image and spatial domains, with models that can tractably generate realistic landscapes, faces (Karras et al. 2019), galaxies (Dia et al. 2019), and dark matter distributions (Mustafa et al. 2019). Sequential data, primarily for natural language (Rajeswar et al. 2017) and music (Engel et al. 2019), have also been modeled with deep generative networks. However, previous to this work, we are not aware of the prior use of deep generative models in the astronomical time domain.

In this work, we presented a deep generative model based on a variational autoencoder architecture that, after being trained with irregularly sampled and noisy light-curve data, is able to reproduce and generate realistic periodic variable sources, such as RR Lyrae, eclipsing binaries, and Cepheids. This model

includes an encoder module to extract relevant information from the light curves and auxiliary metadata (i.e., physical parameters) and condense it into a low-level representation in latent space, and a decoder network that expands the latent code to the reconstruction of the original time series. Both networks make use of temporal convolutional network layers followed by fully connected (dense) layers (Figure 3).

We trained this model with OGLE-III light curves and stellar parameters from the Gaia DR2 catalog. Our trained models are capable of recovering the distinctive characteristics of the light-curve shapes for eight different types of periodic variables. We present a preliminary version of the model trained only with light curves and a second model that includes physical parameters as ancillary inputs. For the first approach, the latent space only encodes the information on light-curve shape, while for the second the latent space includes the information from physical parameters such as effective temperature, brightness (absolute magnitude), and period, highlighting the correlation between the latent and physical space by means of multioutput regression. In that regard, we explored tree-based, linear, and multilayer perceptron regression. Despite the limitations of tree-based aggregate learners to predict near the extrema of the target output variables, when using an rms error loss, the random forest regressor showed a better result than a simple linear model or a one-hidden-layer perceptron model.

With PELS-VAE, we introduce the methodology of generating new light curves by first selecting a vector of physical parameters that is projected into the latent space by means of a regression function. Afterwards, the latent vector is tagged with the desired observing time-stamps and fed into the decoder network, which creates a new light curve. The complete process of generating a batch of 100 new light curves on a modern CPU takes $\sim$1.3 s, independent of the regression method, without parallelization. The two generative models each present distinct advantages. The first model trained solely on the information from the phase-folded light curves is adjustable to include ancillary metadata (i.e., physical parameters) at a later stage without the need to retrain the model anew. The second model processes jointly the photometric observables and the metadata, leading to a better mapping between the latent space and the physical space. The exploration of highly-sophisticated models, such as autoregressive flows, which connect the space of physical parameters to the latent space, constitutes future work.

This research used the Exalearn computational cluster resource provided by the IT Division at the Lawrence Berkeley National Laboratory (supported by the Director, Office of Science, Office of Basic Energy Sciences, of the U.S. Department of Energy under Contract No. DE–AC02–05CH11231). J.M.P. and J.S.B. were partially supported by a Gordon and Betty Moore Foundation Data-Driven Discovery grant. E.S.A. was supported by a National Science Foundation (NSF) Graduate Research Fellowship, under grant DGE 1752814, and a Two Sigma Ph.D. Fellowship.

Software: `numpy` (van der Walt et al. 2011), `matplotlib` (Hunter 2007), `jupyter` (Kluyver et al. 2016), `pytorch` (Paszke et al. 2019), Weight & Biases,[15] `scikit-learn` (Pedregosa et al. 2011), and `pandas` (Wes McKinney 2010).

[15] https://www.wandb.com/

Appendix
Gaia DR2 Cross-match Validation with OGLE-III

The cross-match between the OGLE-III and Gaia DR2 catalogs uses a 2″ search radius. This radius encloses all sources with proper motion (PM) up to 130 mas yr^{-1} given the $\delta t = 15.5$ yr difference between the effective epochs of the catalogs. The resulting cross-matched sources are constrained to the range [−34, 40] mas yr^{-1}. (We found that a modest increase in this radius did not result in the addition of more matches.) This supports the selection of a 2″ search radius without the need for compensation due to high-PM sources.

To validate the resulting cross-matched sources, we first look for possible contaminating sources within the 2″ search radius. A total of 19,545 out of 34,653 sources do not have other neighbors within 2″, and each corresponding Gaia source has a calculated offset over δt that is within the angular distance of the search. There are 15,108 stars that have multiple sources within 2″ of the OGLE source. Of these, 1113 objects do not have neighbors within the angular distance of the OGLE source and the PM radius of the nearest Gaia source, and we accept these as valid matches. We acknowledge that this assumes that the nearest-neighbor match is correct in order to rely on Gaia PM values, which can be supported by the analysis presented above. Only six of 15,108 objects have a cross-match angular distance larger than their own PM radius value. We visually inspect those cases using the Aladin Sky Atlas platform (Bonnarel et al. 2000). Five of these are valid cross-matches, while one source failed a by-eye confirmation of the Gaia DR2 catalog overlaid on Pan-STARRS images as the baseline.

The remaining 13,989 sources had more than one object within the PM radius of the nearest star. Of these, 3835 are listed with variable star classifications in the Gaia Variability Catalog (Rimoldini et al. 2019). In particular, 3732 have matching variability classification (without considering subtypes) between the OGLE and Gaia variability catalogs. The remainder (103) have mismatched classification labels, from which 89 are sources either classified as RR Lyrae or Cepheids by one catalog or the other with measured periods of $<$1 day. A similar result was obtained for nine of 103 objects, which are labeled as eclipsing binary systems by OGLE but have a different label in Gaia catalogs. This confusion is expected due to the small number of observations ($\sim$30 data points) and uneven windowing present in Gaia light curves when compared to the denser and longer-baseline OGLE time series. Therefore, after a visual inspection of light curves from both OGLE and Gaia, we confirmed 98 cross-matches and adopted the OGLE classification. We discarded the remaining five sources due to a catastrophic mismatch in their classification type and visual inspection of the light curves between the two catalogs. A further 461 of 13,989 sources are flagged as variables in the Gaia DR2 catalog but have no assigned variability subtype. For this subsample, we check the Gaia colors ($G_{\rm BP} - G_{\rm RP}$) and effective temperatures ($T_{\rm eff}$) against the corresponding ranges per class for the 19,545 confirmed objects, informed by the known value ranges available in the literature (Catelan & Smith 2015), allowing us to validate the 461 sources as likely correct matches.

Finally, the remaining 9693 sources, from the previous count of 13,989 variable stars, have more than one neighbor within their PM radius and their nearest-neighbor match has no variability information provided by Gaia. To validate this subset, we first filtered following the same temperature and

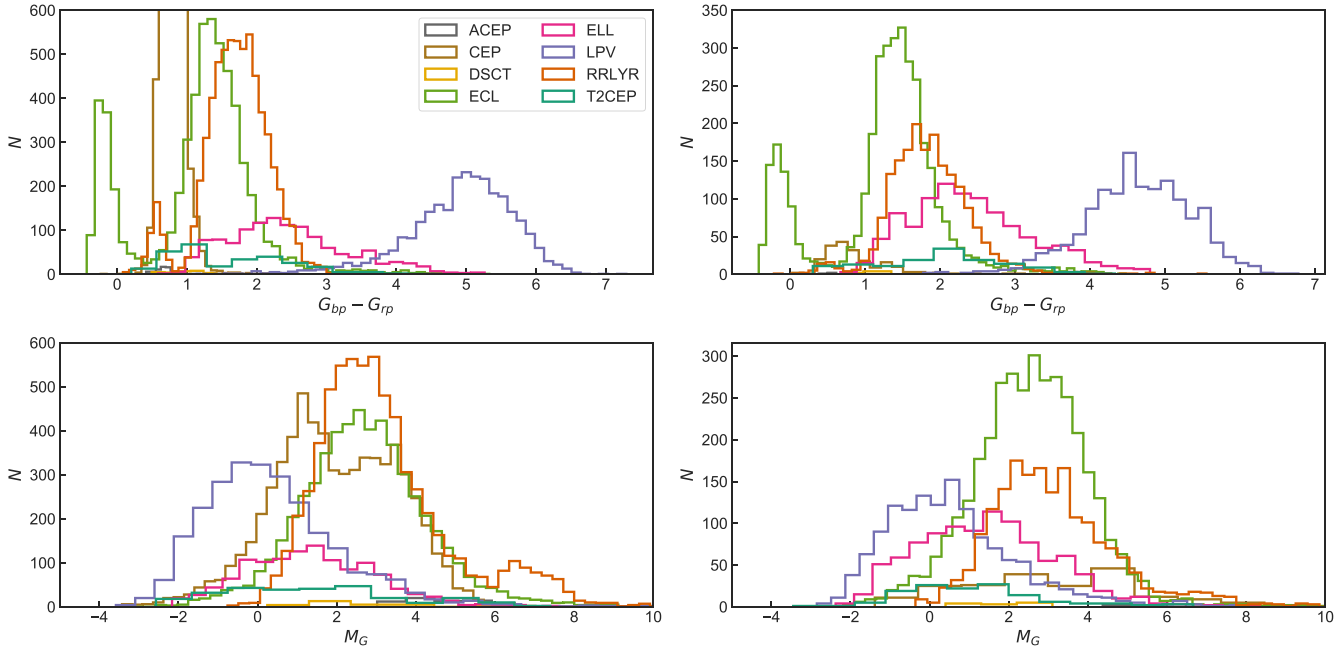

Figure 9. Distributions of Gaia color ($G_{BP} - G_{RP}$, upper row) and absolute g-band magnitudes (M_G, lower row) color-coded by variability classes. The left two panels show the confirmed cross-matches (sources with only one match within PM radius and with matching variability class), while the right panels show the sources validated by CMD comparisons (last group in Table 4).

Table 4
Cross-match Validation Summary

Count	Description
34,653	Size of 2″ cross-match between OGLE and Gaia
19,545	Objects that only have 1 match in 2″ radius
1,113	Objects that have >1 match in 2″ radius but only 1 match in PM radius of nearest neighbor
5	Objects with a cross-match angular distance outside the nearest neighbor's PM radius that passes visual inspection
1	**Object with an angular distance outside the PM radius that does not pass visual inspection**
	4,296 Objects with >1 match in both 2″ radius and nearest-neighbor PM radius and variability information
3,732	Objects with the same variability class in OGLE and Gaia
461	Objects flagged as variable with no assigned class, similar T_{eff} and colors to 19,545 confirmed
98	Objects with adjacent variability classes and visually similar light curves
5	**Objects with fatally different classes and visually dissimilar light curves**
	9,693 Objects with >1 match in both 2″ radius and nearest-neighbor PM radius and no variability information
7,620	Objects that have G_{BP} and G_{RP} measurements in Gaia DR2, validated by CMD placement
539	*Objects whose nearest neighbor lacks G_{BP} and G_{RP} measurements, second-closest neighbor validated by CMD placement*
2	*Objects whose second-closest neighbor has the same variability class in OGLE and Gaia*
755	**Objects that have G_{BP} and G_{RP} measurements in Gaia DR2, degenerate with the main sequence**
465	**Objects with three or more Gaia sources within the nearest-neighbor PM radius**
162	**Objects whose second-closest neighbor is degenerate with the main sequence or Gaia colors too blue for the OGLE class**
151	**Objects do not have inferred distances in the Bailer-Jones catalog for any stars in the cross-match radius**

Note. Lines in bold font were dropped from the cross-match as definite or possible mismatches. For lines in italic font, the second-closest neighbor provided the correct match.

color criteria described in the previous paragraph. However, without additional information on the variability of these sources from Gaia, we further analyzed these sources by inspecting the position of the nearest Gaia source in the color–magnitude diagram (CMD), using the confirmed sample and the locus for known pulsating variables (Gaia Collaboration et al. 2019) as a ground truth comparison. There are 8374 of 9693 sources that have G_{BP} and G_{RP} measurements in Gaia DR2, which we combined with the estimated distances from Bailer-Jones et al. (2018; hereafter the Bailer-Jones catalog) in order to account for known issues with Gaia parallax measurements in crowded areas like the plane of the Galaxy. To avoid color and magnitude degeneracies with possible mismatches, we remove 775 sources within the main-sequence region that could contaminate our sample, validating 7620 nearest-neighbor cross-matches. We reject the 465 objects with

three or more sources in the search radius, except for the two objects whose second-closest neighbor has matching Gaia and OGLE classes. The 853 objects that remain are missing G_{BP} and G_{RP} for the nearest-neighbor match, but the only other star in the cross-match radius had measured colors. There are 151 of 853 objects that do not have estimated distances in the Bailer-Jones catalog and we reject these cross-matches. Since none of the objects in our catalog with confirmed Gaia classes were missing Gaia colors, we placed the second-closest Gaia match on the CMD for the remaining 701 sources and found that 539 sources were in the correct place on the CMD for their OGLE classes. We reject the 162 sources that were degenerate with the main sequence or had Gaia colors that were too blue for the OGLE class. Lastly, after all validation steps, we consolidate our data set with 33,114 valid cross-matches. There are 32,573 of these matches that were with the nearest neighbor and 541 matches were with the second nearest neighbor, all within the PM radius of the Gaia source. Figure 9 shows the distribution of Gaia color and absolute g-band magnitude for confirmed cross-matches and sources validated using the CMD comparison decribed above. We make the OGLE-III/Gaia DR2 cross-match publicly available on Zenodo: doi:10.5281/zenodo.3820679.

ORCID iDs

Jorge Martínez-Palomera https://orcid.org/0000-0002-7395-4935
Joshua S. Bloom https://orcid.org/0000-0002-7777-216X
Ellianna S. Abrahams https://orcid.org/0000-0002-9879-1183

References

Aguirre, C., Pichara, K., & Becker, I. 2019, MNRAS, 482, 5078
Andrae, R., Fouesneau, M., Creevey, O., et al. 2018, A&A, 616, A8
Bachelet, E., Norbury, M., Bozza, V., & Street, R. 2017, AJ, 154, 203
Bai, S., Zico Kolter, J., & Koltun, V. 2018, arXiv:1803.01271
Bailer-Jones, C. A. L., Rybizki, J., Fouesneau, M., Mantelet, G., & Andrae, R. 2018, AJ, 156, 58
Bellm, E. C., Kulkarni, S. R., Graham, M. J., et al. 2018, PASP, 131, 018002
Benavente, P., Protopapas, P., & Pichara, K. 2017, ApJ, 845, 147
Bengio, Y., Simard, P., & Frasconi, P. 1994, ITNN, 5, 157
Bloom, J. S., Richards, J. W., Nugent, P. E., et al. 2012, PASP, 124, 1175
Böhm, V., Lanusse, F., & Seljak, U. 2019, arXiv:1910.10046
Bonnarel, F., Fernique, P., Bienaymé, O., et al. 2000, A&AS, 143, 33
Boone, K. 2019, AJ, 158, 257
Burgess, C. P., Higgins, I., Pal, A., et al. 2018, arXiv:1804.03599
Cabrera-Vives, G., Reyes, I., Förster, F., Estévez, P. A., & Maureira, J.-C. 2017, ApJ, 836, 97
Carrasco-Davis, R., Cabrera-Vives, G., Förster, F., et al. 2019, PASP, 131, 108006
Catelan, M., & Smith, H. A. 2015, Pulsating Stars (New York: Wiley)
Cho, K., van Merrienboer, B., Gulcehre, C., et al. 2014, arXiv:1406.1078
Dia, M., Savary, E., Melchior, M., & Courbin, F. 2019, arXiv:1909.12160
Dieleman, S., Willett, K. W., & Dambre, J. 2015, MNRAS, 450, 1441
Drake, A. J., Catelan, M., Djorgovski, S. G., et al. 2013, ApJ, 763, 32
Engel, J., Agrawal, K. K., Chen, S., et al. 2019, in Int. Conf. on Learning Representations (New Orleans, LA: ICLR), https://openreview.net/forum?id=H1xQVn09FX
Förster, F., Maureira, J. C., San Martín, J., et al. 2016, ApJ, 832, 155
Gabbard, H., Messenger, C., Heng, I. S., Tonolini, F., & Murray-Smith, R. 2019, arXiv:1909.06296
Gaia Collaboration, Prusti, T., de Bruijne, J. H. J., et al. 2016, A&A, 595, A1
Gaia Collaboration, Brown, A. G. A., Vallenari, A., Prusti, T., et al. 2018, A&A, 616, A1
Gaia Collaboration, Eyer, L., Rimoldini, L., et al. 2019, A&A, 623, A110
Goldstein, D. A., D'Andrea, C. B., Fischer, J. A., et al. 2015, AJ, 150, 82
Goodfellow, I., Bengio, Y., & Courville, A. 2016, Deep Learning (Cambridge, MA: MIT Press)
Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., et al. 2014, arXiv:1406.2661
Guillochon, J., Nicholl, M., Villar, V. A., et al. 2018, ApJS, 236, 6
Guo, P., Duan, F., Wang, P., et al. 2019, MNRAS, 490, 5424
He, K., Zhang, X., Ren, S., & Sun, J. 2016, in IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) (New York: IEEE)
Hinton, G. E., & Salakhutdinov, R. R. 2006, Sci, 313, 504
Hochreiter, S., & Schmidhuber, J. 1997, Neural Comp., 9, 1735
Hunter, J. D. 2007, CSE, 9, 90
Ichinohe, Y., & Yamada, S. 2019, MNRAS, 487, 2874
Ivezić, Ž., Kahn, S. M., Tyson, J. A., et al. 2019, ApJ, 873, 111
Jamal, S., & Bloom, J. S. 2020, ApJS, 250, 30
Jimenez Rezende, D., & Mohamed, S. 2015, arXiv:1505.05770
Karras, T., Laine, S., Aittala, M., et al. 2019, arXiv:1912.04958
Kessler, R., Bernstein, J. P., Cinabro, D., et al. 2009, PASP, 121, 1028
Kessler, R., Narayan, G., Avelino, A., et al. 2019, PASP, 131, 094501
Kingma, D. P., & Ba, J. 2014, arXiv:1412.6980
Kingma, D. P., & Welling, M. 2013, arXiv:1312.6114
Kluyver, T., Ragan-Kelley, B., Pérez, F., et al. 2016, in Positioning and Power in Academic Publishing: Players, Agents and Agendas, ed. F. Loizides & B. Scmidt (Amsterdam: IOS Press), https://eprints.soton.ac.uk/403913/
Kullback, S., & Leibler, R. A. 1951, Ann. Math. Statist., 22, 79
Kumar, P., MacDonald, I., Brown, D. A., et al. 2014, PhRvD, 89, 042002
Lample, G., Zeghidour, N., Usunier, N., et al. 2017, arXiv:1706.00409
Lochner, M., McEwen, J. D., Peiris, H. V., Lahav, O., & Winter, M. K. 2016, ApJS, 225, 31
Mahabal, A., Rebbapragada, U., Walters, R., et al. 2019, PASP, 131, 038002
Martínez-Palomera, J., Förster, F., Protopapas, P., et al. 2018, AJ, 156, 186
Martínez-Palomera, J. 2022, jorgemarpa/PELS-VAE, v0.1.1, Zenodo, doi:10.5281/zenodo.7217216
Mustafa, M., Bard, D., Bhimji, W., et al. 2019, ComAC, 6, 1
Muthukrishna, D., Narayan, G., Mandel, K. S., Biswas, R., & Hlozek, R. 2019, PASP, 131, 118002
Naul, B., Bloom, J. S., Pérez, F., & van der Walt, S. 2018, NatAs, 2, 151
Nun, I., Protopapas, P., Sim, B., & Chen, W. 2016, AJ, 152, 71
Pascanu, R., Mikolov, T., & Bengio, Y. 2012, arXiv:1211.5063
Paszke, A., Gross, S., Massa, F., et al. 2019, in 33rd Conf. on Neural Information Processing Systems (NeurIPS 2019) (Vancouver: NeurIPS), https://proceedings.neurips.cc/paper/2019/file/bdbca288fee7f92f2bfa9f7012727740-Paper.pdf
Pedregosa, F., Varoquaux, G., Gramfort, A., et al. 2011, J. Mach. Learn. Res., 12, 2825
Pichara, K., & Protopapas, P. 2013, ApJ, 777, 83
Pichara, K., Protopapas, P., & León, D. 2016, ApJ, 819, 18
Pietrukowicz, P., Kozłowski, S., Skowron, J., et al. 2015, ApJ, 811, 113
Prša, A., Conroy, K. E., Horvat, M., et al. 2016, ApJS, 227, 29
Rajeswar, S., Subramanian, S., Dutil, F., Pal, C., & Courville, A. 2017, arXiv:1705.10929
Richards, J. W., Starr, D. L., Miller, A. A., et al. 2012, ApJS, 203, 32
Richards, J. W., Starr, D. L., Butler, N. R., et al. 2011, ApJ, 733, 10
Rimoldini, L., Holl, B., Audard, M., et al. 2019, A&A, 625, A97
Sánchez, B., Domínguez, R., & Lares, M. J. 2019, A&C, 28, 100284
Sesar, B., Ivezić, Ž., Grammer, S. H., et al. 2010, ApJ, 708, 717
Smolec, R. 2005, AcA, 55, 59
Spergel, D., Gehrels, N., Baltay, C., et al. 2015, arXiv:1503.03757
The PLAsTiCC team, Allam, T., Jr, Bahmanyar, A., et al. 2018, arXiv:1810.00001
Tröster, T., Ferguson, C., Harnois-Déraps, J., & McCarthy, I. G. 2019, MNRAS, 487, L24
Tsang, B. T. H., & Schultz, W. C. 2019, ApJL, 877, L14
Udalski, A., Szymanski, M., Kaluzny, J., Kubiak, M., & Mateo, M. 1992, AcA, 42, 253
Udalski, A., Szymanski, M. K., Soszynski, I., & Poleski, R. 2008, AcA, 58, 69
van der Walt, S., Colbert, S. C., & Varoquaux, G. 2011, CSE, 13, 22
Wes McKinney 2010, in Proc. 9th Python in Science Conf., ed. S. van der Walt & J. Millman (Austin, TX: SciPy), 56
Yi, K., Guo, Y., Fan, Y., Hamann, J., & Wang, Y. G. 2020, arXiv:2001.11651
Yu, F., & Koltun, V. 2015, arXiv:1511.07122
Zhang, L. 2019, arXiv:1903.04687
Zorich, L., Pichara, K., & Protopapas, P. 2020, MNRAS, 492, 2897