Clustering of Astronomical Transient Candidates Using Deep Variational Embedding

Nicolás Astorga*, Pablo Huijse†*, Pablo A. Estévez *†, Francisco Förster ‡†
*Department of Electrical Engineering
Universidad de Chile, Santiago, Chile
{nastorga, phuijse, pestevez }@ing.uchile.cl
†Millennium Institute of Astrophysics, Santiago, Chile
‡Center for Mathematical Modeling
Universidad de Chile, Santiago, Chile

***Abstract*—The exponential growth of the data collected by telescopes have turned astronomy into a data-drive science. The detection of astronomical transient events, short-lived and bright phenomena such as the Supernovae, is currently a main science driver of many astronomical surveys. There is an opportunity for the application of machine learning methods for the automatic detection of astronomical transients.**

In this paper we focus on the unsupervised learning case to perform an exploratory analysis on a dataset of 1,250,000 astronomical transient candidates from the High Cadence Transient Survey. Our contributions can be summarized in 1) The application of Deep Variational Embedding for latent space clustering of a large database of transient candidates obtaining a clustering accuracy of 95.33% and 2) The proposal of an auto-regularization term as a novel approach to solve the common problem of over-regularization in variational autoencoders, we show that using this term not only improves the convergence of the algorithm but also increases the clustering accuracy and reconstruction quality.

***Index Terms*—autoencoder, variational inference, clustering, astronomical images, transients**

I. Introduction

In recent years astronomy have faced a paradigm shift towards data-intensive science. Soon to be deployed instruments such as the Large Synoptic Survey Telescope [1] will produce Petascale databases and approximately 10 million alerts per night corresponding to candidate transient events occurring in the southern hemisphere sky. Astronomical transients correspond to a rapid change in brightness associated to certain astrophysical phenomena, *e.g.* Supernovae, the bright explosion associated to the death of massive stars. Detecting, classifying and characterizing the transient universe is one of the main scientific drivers of such new telescopes.

The abundance of data and the impossibility of performing visual inspection routines have created an opportunity for researchers on statistics and machine learning to develop supervised methods to solve big-data problems related to the automatic discrimination of astronomical transients [2], [3]. More recently, models that do not require feature engineering such as the convolutional deep neural network have also find success in the detection of supernovae transients [4].

However despite the large amount of existing data, the number of labeled data is relatively scarce, difficulting the use of fully-supervised models. This motivates the use of unsupervised methods to not only better understand the data structure but also to detect clusters of similar objects.

The variational autoencoder [5] combines an unsupervised neural network with a generative model allowing the extraction of rich latent codes from the data. In this paper we propose an unsupervised learning method to cluster astronomical transients based on Deep Variational Embedding (VADE) [6] using Convolution Neural Networks [7] as feature extractors. The VADE model modifies the original VAE by including a Mixture of Gaussians (MoG) prior that is more appropriate for clustering. We modify the VADE architecture to incorporate a model for the decoder variance which facilitates the training of the deep autoencoder, yielding better data reconstruction and allowing us to estimate the noise in the data.

II. Data

Core-collapse supernovae (SNe) are highly-energetic explosions occurring when a massive star can no longer sustain itself against gravity. These transient astronomical events are characterized by a sudden and steep rise in luminosity followed by a relatively slower decay and diffusion phase. In this work we use data from the High Cadence Transient Survey (HiTS, [3]), an observation campaign that aimed for the real-time detection of SNe in order to help confirm or discard theoretical astrophysical models. HiTS observations were made using the Dark Energy Camera (DECam, [8]), a 520 Mpixel CCD imager mounted at the Cerro Tololo observatory near La Serena, Chile.

The HiTS data processing pipeline [3] detects transient sources using a procedure based on image differences which are obtained by subtracting a template image (usual sky condition) from a science image taken at a different time. Before subtraction the images are aligned and also matched in terms of the quality of their point-spread function (PSF) to mitigate differences due to atmospheric conditions. The difference image is normalized using a local estimation of the noise obtaining an SNR image. Transient candidates are selected from the SNR image as 21×21 pixel stamps centered around pixels that varied more than 5σ with respect to the background.

978-1-5090-6014-6/18/$31.00 ©2018 IEEE

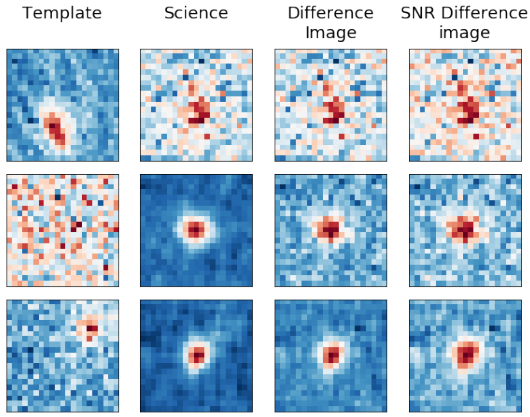


Fig. 1. Examples of simulated stellar transients (positive class).

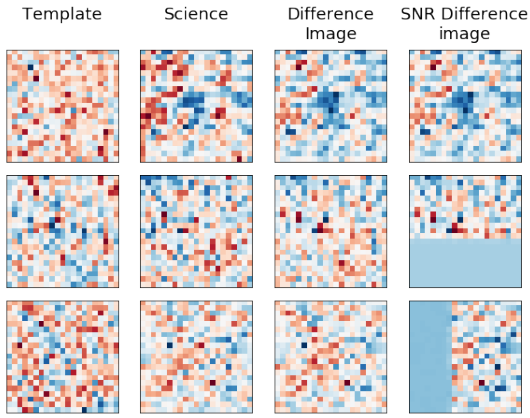


Fig. 2. Examples of artifacts (negative class).

In this work we consider data from the 2013 HiTS campaign, where 120 sq. deg. of the sky were observed in the u-band every 2 hours during 4 consecutive nights. The HiTS 2013 training dataset [3], [4] has 1,604,174 transient candidates, with half of them corresponding to stellar transients (positives examples) and the other half to artifacts (negative examples) produced by errors in the CCDs, badly aligned subtraction, badly removed cosmic rays, statistical fluctuation of the background, among other causes. Our unsupervised models are trained using the SNR difference stamps. The labels in the dataset are only used to evaluate the quality of the models. The dataset is splitted into the usual training, validation and test set, each one containing 1,250,000, 100,000 and 100,000 respectively. Figures 1 and 2 show examples of positive and negative transient candidates, respectively.

III. Literature Review

A. Variational autoencoder

The variational autoencoder (VAE) is a generative model [5] that extends the conventional autoencoder by adding a stochastic latent layer and a probabilistic decoder from which new data can be sampled. VAE can be seen as a probabilistic graphical model consisting of two fundamental sections:

- An encoder or inference model denoted by $q_\phi(z|x)$. This is a neural network with parameters ϕ that encodes the data X into latent variables z as shown in Fig. 3a. The user decides the amount of latent variables and whether they are continuous [5] or discrete [9].
- A decoder or generative model $p_\theta(x|z)$. This is also modeled by a neural network with parameters θ that uses the latent variables z obtained from $q_\phi(z|x)$ to perform a reconstruction $\hat{X}$ of the data as shown in Fig. 3b.

The VAE tries to approximate the real distribution $p(x)$ of the data through a variational approximation. The loss function to be optimized is known as the evidence lower bound (ELBO) which is obtained from the log likelihood $p(x)$ using Jensen's inequality

$$\begin{aligned}\log p(x) &= \log \int_z p(x,z)dz \\ &\geq \mathbb{E}_{q_\phi(z|x)}\left[\log \frac{p_\theta(x,z)}{q_\phi(z|x)}\right] \quad (1)\\ &\equiv \mathcal{L}(\theta,\phi,x) = \mathbb{E}_{q_\phi(z|x)}\left[\ln(p_\theta(x|z)\right] \\ &- \mathrm{D}_{KL}(q_\phi(z|x)||p(z)). \quad (2)\end{aligned}$$

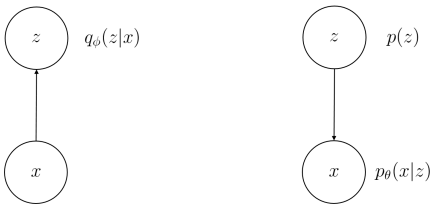
From Eq. (1) to Eq. (2) it is assumed that the joint probability distribution $p(x,z)$ can be decomposed as a generative model of the form $p_\theta(x|z)p(z)$.

Note that the terms in the Eq. (2) can be interpreted as the reconstruction error of the input data plus a regularization term related to the latent variables. For Eq. (2) to be differentiable, the regularization term (Kullback-Leiber divergence) is generally required to have an analytical form. For mathematical convenience the prior term $p(z)$ is a unitary Gaussian $\mathcal{N}(0,I)$ and the posterior variational approximation $q_\phi(z|x)$ is a diagonal Gaussian $\mathcal{N}(\mu_\phi(x), \mathrm{diag}(\sigma^2_\phi(x)))$, where both parameters $\mu_\phi(x)$ and $\sigma_\phi(x)$ are obtained by neural networks (encoder). Using these assumptions the following estimator of Eq. (2) is obtained

$$\begin{aligned}\mathcal{L}(\theta,\phi,x^{(i)}) &= \frac{1}{2}\sum_{j=1}^{J}\left[1+\log\sigma_j^2(x^{(i)}) - \mu_j^2(x^{(i)}) - \sigma_j^2(x^{(i)})\right] \\ &+ \frac{1}{L}\sum_{l=1}^{L}\log p_\theta(x^{(i)}|z^{(i,l)}), \quad (3)\end{aligned}$$

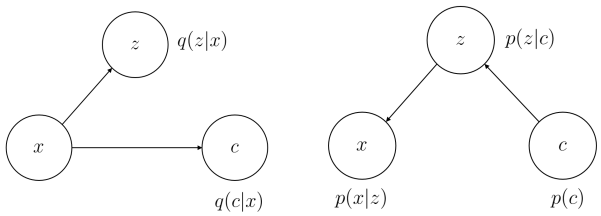
where the subscript i is associated to the samples, the superscript J denotes the number of dimensions of the latent variables and L is the number of Montecarlo samples $\hat{z}$ generated by the distribution of the inferential model $q_\phi(z|x)$.

The VAE is trained by gradient descent using Eq. (3) as a loss function. To perform backpropagation to the section of the network prior to the samples $\hat{z}$, the reparameterization trick is used. This consists of changing the sampling of a blackbox normal distribution $\mathcal{N}(\mu_\phi(x), \mathrm{diag}(\sigma^2_\phi(x)))$ to a differentiable sampling of the form $\hat{z} = \mu_\phi(x) + \varepsilon \odot \sigma_\phi(x)$, where $\varepsilon \sim \mathcal{N}(0,I)$ and $\odot$ is the element-wise multiplication. In this way the gradient can backpropagate through $\mu_\phi(x)$ and $\sigma_\phi(x)$.



(a) Inference model. (b) Generative model.

Fig. 3. Probabilistic graphical models for the classical variational autoencoder.



(a) Inference model. (b) Generative model.

Fig. 4. Probabilistic graphical models for deep variational embedding.

B. Deep variational embedding

Deep variational embedding (VADE) [6] is an extension of the variational autoencoder developed for clustering applications that sets a Mixture of Gaussians (MoG) prior $p(z, c) = p(z|c)p(c)$ over the latent variables z, where $c \in [1, K]$ is a categorical latent variable parametrized by π. In VADE the joint probability $p(x, z, c)$ of the generative model is decomposed as $p(x|z)p(z|c)p(c)$ and the inferential model $q(z, c|x)$ as $q(z|x)q(c|x)$, the corresponding graphical models are shown in figures 4a and 4b, respectively. The probability distributions of the generative model are $p(c) = \text{Cat}(c|\pi)$, $p(z|c) = \mathcal{N}(z|\mu_c, \sigma_c^2 I)$ and $p(x|z) = \text{Ber}(x|\mu_x)$ or $\mathcal{N}(z|\mu_x, \sigma_x^2 I)$, where $\text{Cat}(\cdot)$ and $\text{Ber}(\cdot)$ stand for the categorical and Bernoulli distributions, respectively. For the inferential model as with the classical VAE, we have $q(z|x) = \mathcal{N}(\mu(x), \text{diag}(\sigma_\phi^2(x)))$ and the term $q(c|z)$ is obtained by an approximation shown in Eq. (6).

Using these assumptions the ELBO can be obtained using Jensen's inequality

$$\begin{aligned}\log p(x) &\geq \mathbb{E}_{q(z,c|x)}\left[\log \frac{p(x,z,c)}{q(z,c|x)}\right] \\ &= \mathbb{E}_{q(z,c|x)}[\log p(x|z) + \log p(z|c) + \log p(c) \\ &\quad - \log q(z|x) - \log q(c|x)] \equiv \mathcal{L}_{ELBO}(x). \end{aligned} \tag{4}$$

By replacing the corresponding probability distributions an analytical expression of Eq. (4) is obtained

$$\begin{aligned}\mathcal{L}_{ELBO}(x) &= \frac{1}{L}\sum_{l=1}^{L}\log p_\theta(x^{(i)}|z^{(i,l)}) \\ &- \frac{1}{2}\sum_{c=1}^{K}\gamma_c\sum_{j=1}^{J}(\log\sigma_c^2|_j + \frac{\sigma^2|_j}{\sigma_c^2|_j} + \frac{(\mu|_j - \mu_c|_j)^2}{\sigma_c^2|_j}) \\ &+ \sum_{c=1}^{K}\gamma_c\log\frac{\pi_c}{\gamma_c} + \frac{1}{2}\sum_{j=1}^{J}(1 + \log\sigma^2|_j), \end{aligned} \tag{5}$$

where K is the number of clusters, μ and σ^2 are the parameters of $q(z|x)$, μ_c and σ_c are the parameters of the components of the MoG, and γ_c is $q(c|x)$, *i.e.* the membership probability of sample x to cluster c which is estimated using the following approximation from [6]

$$\begin{aligned}\gamma_c \equiv q(c|x) \approx p(c|z) &\equiv \frac{p(c)p(z|c)}{\sum_{c'=1}^{K}p(c')p(z|c')} \\ &= \frac{\pi_c \exp\left(-\frac{(\hat{z}-\mu_c)^2}{2\sigma_c^2}\right)}{\sum_{c'=1}^{K}\pi_{c'}\exp\left(-\frac{(\hat{z}-\mu_{c'})^2}{2\sigma_{c'}^2}\right)}. \end{aligned} \tag{6}$$

Finally, we can use $\arg\max_{c\in[1,K]}\gamma_c$ to assign a sample to a particular component of the MoG.

C. Issues due to over-regularization in VAE

In the VAE setting, the real distribution $p(x)$ is approximated by maximizing the ELBO, *i.e.* the likelihood of the reconstructed data minus the divergence between the latent codes and the chosen prior (Eq. 2). In general, to obtain a low reconstruction error, a latent space that has codified all the useful information of the input is needed. During the early training stages of the VAE the latent space carries little information from the input and the reconstruction is poor. In some cases this causes the optimization to set on a bad local minimum where the latent codes do not depart from the prior [10]. Other studies show that the collapse of the latent codes to the prior also occurs when very complex decoders $p_\theta(x|z)$ are used [11], [12]. This can be explained by rewriting the ELBO as

$$\begin{aligned}\mathcal{L}_{ELBO} &= -\text{D}_{KL}(p_{data}(x)||p_\theta(x|z)) \\ &\quad - \mathbb{E}_{p_{data}(x)}[\text{D}_{KL}(q_\phi(z|x)||p_\theta(z|x))], \end{aligned} \tag{7}$$

where if the distribution $p_\theta(x|z)$ is flexible enough, there is a member in the conditional distribution $p^*(x) = p_\theta(x|z)$ for all $z \in \mathcal{Z}$ that can fit the real distribution $p_{data}(x)$. This means that z becomes independent of x, *i.e.* $p_\theta(z|x) = p(z)$ and $q_\phi(z|x) = p(z)$ [12] producing an uninformative latent space very close to prior.

To overcome the optimization challenges related to the over-regulatization problem a variety of techniques have come out, *e.g.* using annealing procedures on the KL divergence [10] or thresholding the values of the regularization term [13]. In this paper we propose an auto-regularization term explained in Section V.

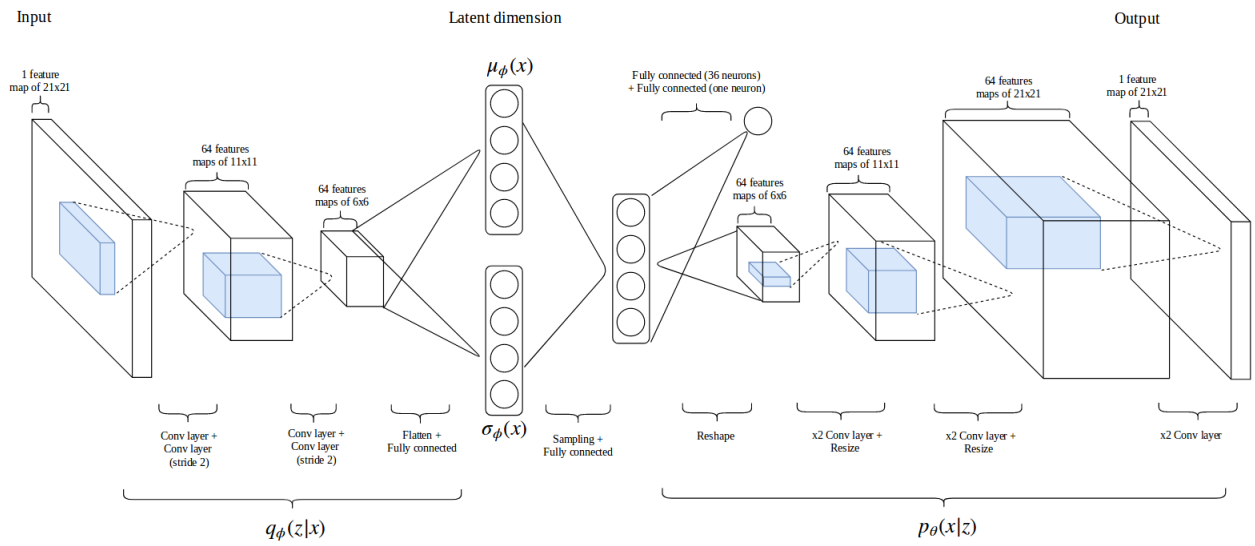


Fig. 5. Proposed VADE architecture.

IV. Model Architecture

Fig. 5 shows the architecture of the VADE model. The encoder neural network $q_\phi(z|x)$ receives an input image of 21×21 pixels (SNR difference stamp) which passes through four convolutional layers and one last fully connected layer (FCL) until it reaches the latent dimension. The first and third convolutional layers in the encoder use 3×3 kernels with stride 1. Feature map dimensionality reduction is implemented in the second and fourth layers using 3×3 convolutions with stride 2. The feature maps obtained by the fourth convolutional layer are flattened to obtain one feature vector which passes through two parallel FCL to estimate $\mu_\phi(x)$ and $\sigma^2_\phi(x)$, respectively. The number of neurons in these FCL layers corresponds to the dimensionality of the latent space and is considered as an hyperparameter whose influence is explored in Section VII.

The decoder model $p_\theta(x|z)$ takes the sampled latent variables $\hat{z}$ obtained using $\mu_\phi(x)$, $\sigma^2_\phi(x)$ and the reparametrization trick and passes them through a FCL. These neurons are reshaped to obtain 64 feature maps of 6×6 which are connected to six convolution layers with 3×3 kernels and stride 1. Nearest neighbor interpolation is used to increase the dimensionality of the feature maps to 11×11 after the second layer and to 21×21 after the fourth layer. The last convolutional layer reduces the number of filters to match the dimensionality of the input.

All layers in the model use ReLU activation functions and Batch Normalization [14]. The number of feature maps in all convolutional layers is 64. The hyperparameters of the architecture were chosen using the validation set. In this case the distribution of the decoder $p_\theta(x|z)$ that best fit the data is Gaussian, *i.e.* we use the mean square error (MSE) cost for the data likelihood in Eq. (5). Note that both VADE and VAE architectures are equal, the only difference being the regularization term. This does not mean that the latent space of both models will be the same, as we expect the VADE model to separate the data in Gaussian clusters.

V. Auto-regularization

To deal with the over-regularization problems presented in Section III-C we propose to add a neural network connected to the latent variables $\hat{z}$ that estimates the variance $\sigma^2(\hat{z})$ of the Gaussian distribution associated to the decoder. The reconstruction term of Eq. (5) considering $\sigma^2(\hat{z})$ is

$$\log p_\theta(x|z) = -\frac{(x-\mu_x)^2}{2\sigma^2(\hat{z})} - \frac{1}{2}\log(2\pi\sigma^2(\hat{z})). \qquad (8)$$

The variance $\sigma^2(\hat{z})$ is estimated using a 2-layered neural network. The first layer has 36 neurons with ReLU activation function. The second layer has 1 neuron with a sigmoid activation function. We parametrize the sigmoid output as $-0.2\log\sigma^2(\hat{z})$, *i.e.* we constrain the variance to $\sigma^2(\hat{z}) \in [e^{-5}, 1]$. We found that constraining the variance is important to avoid numerical instabilities during training. These bounds are data-dependant and were found through validation.

Note that with this the model has two outputs for each realization $\hat{z}$ of an image x, a 21×21 reconstruction μ_x and a single number associated to the variance $\sigma^2(\hat{z})$. The value of $\sigma^2(\hat{z})$ controls the trade-off between the reconstruction error and the KL divergence between the latent codes and the prior (Eq. 5), although it does it independently for each image. This differs from previous works where the trade-off is controlled by a global constant. Also, with the proposed approach there is no need to tune an extra parameter by cross-validation as $\sigma^2(\hat{z})$ is learned from the data. Examples with smaller $\sigma^2(\hat{z})$ are less influenced by the prior allowing for better reconstruction to be achieved. In practice this allows the model to escape bad local

minima where the latent codes have collapsed to the prior, facilitating the training of deep autoencoder architectures.

In what follows we refer to the decoder variance as an auto-regularization term due to its properties as an automatic weight between reconstruction and regularization of the VADE cost. In the results sections we present the advantages of incorporating the decoder variance in terms of reconstruction quality and clustering accuracy. We also discuss about the relationship of the decoder variance and the noise in the input.

VI. Methodology

In this section we outline the methodology to obtain our results using the VADE model with auto-regularization. The architecture presented in the last section is trained using Eq. (5). All the parameters of the model are initialized using Xavier's rules [15] except for μ_c, σ_c and π_c. For the initial values of the first two VADE parameters we assign them the mean and standard deviation of a diagonal mixture of Gaussian fitted to the latent variables of a VAE pretrained model. For the initial value of the categorical prior π_c we create a K-length vector with all its entries equal to $\frac{1}{K}$. Learning and momentum rates are adapted with ADAM [16] and the initial learning rate is set to $0.333 \cdot 10^{-4}$. The training was run until convergence of the ELBO which varies between 40 and 80 epochs depending of the hyperparameters settings. The model is feed with mini-batches of 100 examples. The implementation was programmed in Tensorflow [17] and run on a Nvidia Geforce Titan X Pascal graphics processor unit (GPU). The VADE models were trained for 160 epochs (12 hours), divided into 80 epochs of pretraining with VAE cost function and 80 epochs of the VADE training.

The training was performed without using labels. However, to evaluate the performance of the models we use the labels to estimate the clustering accuracy as

$$accuracy = \frac{1}{N} \sum_{c=1}^{K} \max(p_c, n_c), \tag{9}$$

where p_c and n_c are the number of positive (stellar transient) and negative (artifact) examples that belong to cluster c, respectively, N is the number of examples in the database and K is the number of clusters. A high accuracy means that the classes are well separated by the data partitioning given by the clusters. The clustering accuracy of VADE is estimated using the average γ_c over 10 Monte-carlo samples (L=10). The accuracies shown in Section VII were obtained on the test set using validation set for hyperparameter model tuning.

The VADE is compared to other latent variable extraction methods such as VAE, incremental PCA [18] and online dictionary learning (DL) [19]. For VAE, PCA and DL we perform clustering over their latent codes using GMM.

VII. Results

First we present the results related to the exploration of the hyperparameters of the VADE model. Table IIa shows the clustering accuracy of VADE using different number of clusters (K in Eq. 5). In this experiment a latent dimensionality of 21 is used, *i.e.* the root square of the input dimensionality. The highest accuracy is obtained with $K = 10$. Table IIb shows the clustering accuracy using $K = 10$ but changing the number of latent dimensions in the VADE model (J in in Eq. 5). The accuracy increases with the number of latent dimensions Table IIa reaching its maximum at $J = 21$.

Table IIc compares the accuracy of VADE and GMM, the later applied over the latent codes of the conventional VAE, incremental PCA and online DL. All methods use 10 clusters and 21 latent dimensions. Error bars are obtained by training the algorithms five times. The VADE model outperforms its competitors and is also more stable across different runs. A Welch's t-test on the accuracies obtained from PCA+GMM and VADE yields a p-value of 0.038, *i.e.* the average accuracies are significantly different.

As we are in an unsupervised scheme it is important to get an insight of the data and the dominant structures present in it. Using the VADE we can obtain prototypes or centroids associated to each cluster by sampling from μ_c and decoding the image associated to it. Fig. 6 shows these reconstructions, where the first and second rows correspond to clusters with a majority of positive (stellar transients) and negative (artifacts) examples, respectively. It is particularly interesting to inspect the behaviors arising in the artifact class. For example, the third prototype corresponds to a source that decreased in brightness, while the fifth prototype shows a source with a saturated region to its left. By understanding the artifacts we can correct and better tune the image difference pipeline.

Next we show the influence of the auto-regularization term in the solutions obtained by the model. Fig. 7 shows data reconstructions obtained with VADE for three particular examples with (second column) and without (third column) using the auto-regularization term. When the auto-regularization term is used the reconstructions are more faithful and the MSE reaches 3.16. If this term is neglected poor reconstructions are obtained, the examples or more difficult to discriminate, and the MSE sets on a local minimum of 4.4.

We compare with [10] where a coefficient β weighting the KL divergence is annealed from 0 to 1 to avoid latent variable collapse. On convergence this strategy obtains a 55.4% validation accuracy, *i.e.* the over-regularization problem is not necessarily avoided by slowly increasing the KL divergence. In our model $\sigma^2(\hat{z})$ can be interpreted as a coefficient that weights the KL divergence over likelihood differently for each sample. Fig. 8a shows that after convergence $\sigma^2(\hat{z}) \in [0.005, 0.04]$, *i.e.* the optimal β may not be close to 1.

The importance of the auto-regularization term is not limited to improving the convergence of the model. The output variance of the decoder (auto-regularization term) can be used to better discriminate noisy data. To illustrate this idea we present in Fig. 8a the average mean square error as a function of the average decoder variance over 10 Montecarlo samples for the validation set data. We note that images which are assigned a large variance by the model are in general harder to reconstruct. Fig. 8b shows examples of low and high variance images and their reconstructions. We don't

TABLE I
HYPERPARAMETER EXPLORATION AND COMPARISON WITH OTHER METHODS

Number of clusters	Accuracy
2	92.32 %
6	90.30 %
10	95.33 %
20	95.02 %

(a) Clustering accuracy of VADE using 21 latent dimensions as a function of the number of clusters.

Latent space dimensionality	Accuracy
2	88.69 %
5	91.87 %
10	94.73 %
21	95.33 %

(b) Clustering accuracy of VADE using 10 clusters as function of the number of latent dimensions.

Method	Accuracy
Incremental PCA + GMM	93.43 ± 1.25 %
Online DL + GMM	90.79 ± 0.22 %
VAE + GMM	87.06 ± 0.69 %
VADE	95.33 ± 0.11 %

(c) Clustering accuracy (10 clusters) using VADE, VAE, PCA and Dictionary Learning (DL). For VAE, PCA and DL the clustering is obtained by fitting a GMM on their latent codes. All methods use 21 latent dimensions.

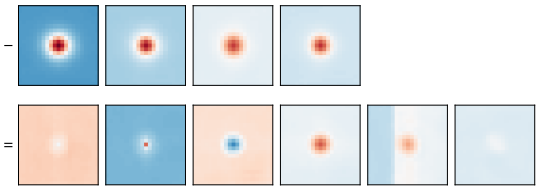


Fig. 6. Reconstruction obtained by sampling from the VADE centroids μ_c. The first and second rows correspond to clusters associated to the positive class (stellar transient) and the negative class (artifacts), respectively.

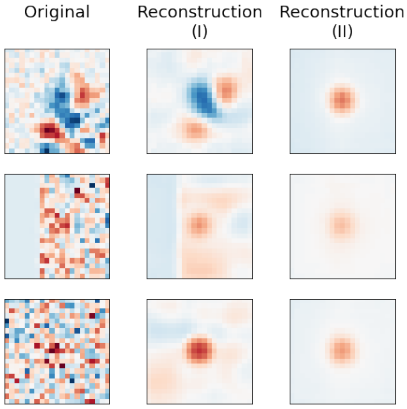


Fig. 7. Examples of SNR images and their reconstructions using the VADE model with (I) and without (II) the auto-regularization term.

expect the model to reproduce the noise as this would be overfitting, hence noisier images (first two rows) obtain larger reconstruction MSE. Images with clear structure (last two rows) are better reconstructed and their predicted variance is lower. This indicates that the decoder variance could be used as an estimation of SNR of the image.

We also extend our analysis to find the relation between the clustering accuracy and the variance of the images. We sort the images in the validation set by their average decoder variance (auto-regularization term) and we split them in 10 bins. Fig. 9 shows the clustering accuracy using the data corresponding to each of the bins. From the figure we can see that the accuracy is inversely proportional to the decoder variance, as it is more difficult to cluster noisy images. In the future this extra information could be used to develop a pipeline in which images are processed differently according to their SNR.

VIII. Conclusions and Future Work

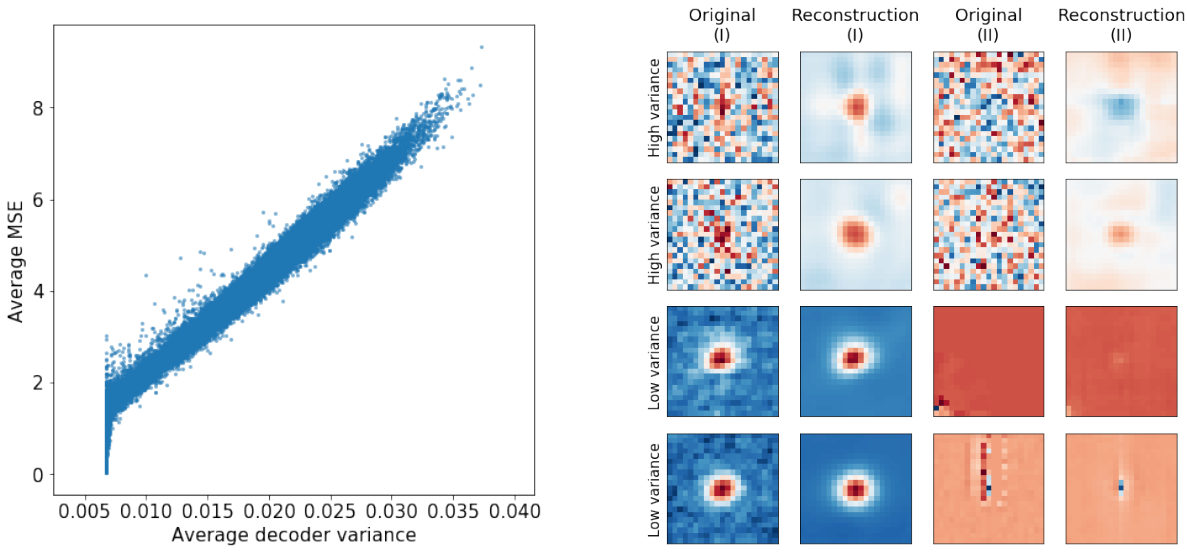
We proposed a neural network model based on deep variational embedding to perform clustering on transient supernova candidates utilizing 1,250,000 SNR difference images from the HiTS survey. The proposed model learns a latent space from which the underlying classes of the data are better separated with respect to the VAE and classical latent variable methods.

An important contribution of this work is the addition of a model for the variance of decoder. This new term acts as a automatic trade-off between reconstruction and regularization, improving the training stability of the VADE and allowing it to converge to better solutions without the need of ad-hoc annealing procedures or thresholding. We studied how the decoder variance relates to the clustering accuracy and the reconstruction error and found that this term could be used to predict the relative amount of noise in the data.

For future work we plan to explore different techniques to make the model more expressive such as better encoder posterior distributions [13], [20], [21] or priors that can better capture the tails of the distribution [22]. We also plan to extend our model to the semi-supervised case where a few labels are available to train [23], [24].

Acknowledgment

Pablo Huijse, Pablo A. Estévez and Francisco Förster acknowledge support from FONDECYT through grants 1170305, 1171678 and 3110042, respectively. F. F. acknowledges support from Basal Project PFB-03. The authors acknowledge support from CONICYT through the Programme of International Cooperation project DPI20140090 and from the Chilean Ministry of Economy, Development, and Tourism's Millennium Science Initiative through grant IC12009, awarded to The Millennium Institute of Astrophysics, MAS.



(a) (b)

Fig. 8. Fig. (a) shows the squared error between the data and its reconstruction as a function of its average decoder variance $\sigma^2(\hat{z})$. In general, the better the reconstruction the smaller the variance. Fig (b) shows positive (I) and negative (II) examples from the dataset and their reconstructions by the VADE model. Noisier examples (first two rows) obtain a larger decoder variance than those with more clear structure (last two rows).

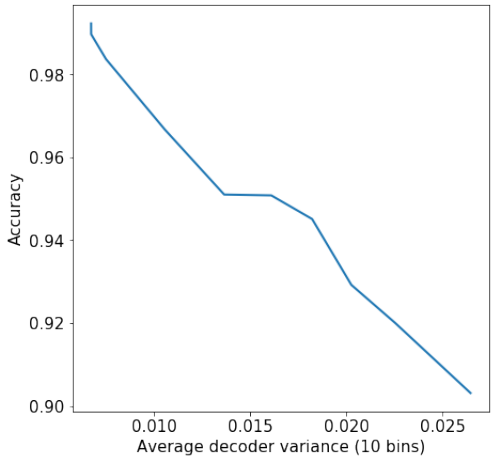


Fig. 9. Accuracy vs average decoder variance. The accuracy was obtained using data bins of size 10,000, sorted by decoder variance. The noisier the data the poorer the clustering accuracy.

REFERENCES

[1] Z. Ivezic, J. Tyson, B. Abel, E. Acosta, R. Allsman, Y. AlSayyad, S. Anderson, J. Andrew, R. Angel, G. Angeli *et al.*, "Lsst: from science drivers to reference design and anticipated data products," *arXiv preprint arXiv:0805.2366*, 2008.

[2] S. G. Djorgovski, A. Mahabal, C. Donalek, M. J. Graham, A. J. Drake, B. Moghaddam, and M. Turmon, "Flashes in a star stream: Automated classification of astronomical transient events," in *E-Science (e-Science), 2012 IEEE 8th International Conference on*. IEEE, 2012, pp. 1–8.

[3] F. Förster, J. C. Maureira, J. San Martín, M. Hamuy, J. Martínez, P. Huijse, G. Cabrera, L. Galbany, T. De Jaeger, S. González-Gaitán *et al.*, "The high cadence transient survey (hits). i. survey design and supernova shock breakout constraints," *The Astrophysical Journal*, vol. 832, no. 2, p. 155, 2016.

[4] G. Cabrera-Vives, I. Reyes, F. Förster, P. A. Estévez, and J. Maureira, "Deep-HiTS: Rotation Invariant Convolutional Neural Network for Transient Detection," *The Astrophysical Journal*, vol. 836, no. 1, pp. 97–104, 2017.

[5] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," in *2nd International Conference on Learning Representations*, 2014.

[6] Z. Jiang, Y. Zheng, H. Tan, B. Tang, and H. Zhou, "Variational deep embedding: An unsupervised and generative approach to clustering," in *International Joint Conference on Artificial Intelligence*, 2017.

[7] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, ser. NIPS'12. USA: Curran Associates Inc., 2012, pp. 1097–1105.

[8] B. L. Flaugher, T. M. Abbott, R. Angstadt, J. Annis, M. L. Antonik, J. Bailey, O. Ballester, J. P. Bernstein, R. A. Bernstein, M. Bonati *et al.*, "Status of the dark energy survey camera (decam) project," in *SPIE Astronomical Telescopes+ Instrumentation*. International Society for Optics and Photonics, 2012, pp. 844 611–844 611.

[9] E. Jang, S. Gu, and B. Poole, "Categorical reparameterization with gumbel-softmax," in *International Conference on Learning Representations*, 2017.

[10] R. T. M. L. S. S. Sønderby, C.K. and O. Winther, "How to train deep variational autoencoders and probabilistic ladder networks." in *29th Conference on Neural Information Processing Systems*, 2016.

[11] X. Chen, D. P. Kingma, T. Salimans, Y. Duan, P. Dhariwal, J. Schulman, I. Sutskever, and P. Abbeel, "Variational lossy autoencoder," *arXiv preprint arXiv:1611.02731*, 2016.

[12] S. Zhao, J. Song, and S. Ermon, "Infovae: Information maximizing variational autoencoders," *arXiv preprint arXiv:1706.02262v2*, 2017.

[13] D. P. Kingma, T. Salimans, and M. Welling, "Improving variational inference with inverse autoregressive flow," *arXiv preprint arXiv:1606.04934*, 2016.

[14] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *32nd International Conference on Machine Learning*, 2015, pp. 448–456.

[15] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, 2010, pp. 249–256.

[16] D. P. Kingma and J. L. Ba, "Adam: A Method for Stochastic Opti-

mization," in *3rd International Conference on Learning Representations*, 2014.

[17] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin *et al.*, "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," *arXiv preprint arXiv:1603.04467*, 2016.

[18] D. A. Ross, J. Lim, R. S. Lin, and M. H. Yang, "Incremental learning for robust visual tracking," *International Journal of Computer Vision*, vol. 77, no. 1, pp. 125–141, 2008.

[19] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online dictionary learning for sparse coding," in *Proceedings of the 26th annual international conference on machine learning*. ACM, 2009, pp. 689–696.

[20] S. Mescheder, Lars. Nowozin and A. Geiger, "Adversarial variational bayes: Unifying variational autoencoders and generative adversarial networks," in *34nd International Conference on Machine Learning*.

[21] D. Rezende and S. Mohamed, "Variational inference with normalizing flows," in *32nd International Conference on Machine Learning*.

[22] Z. L. X. W. C. X. E. Goyal, Prasoon. Hu, "Nonparametric variational auto-encoders for hierarchical representation learning," *arXiv preprint arXiv: 1703.07027v2*, 2017.

[23] D. M. S. W. M. Kingma, Diederik. Rezende, "Semi-supervised learning with deep generative models." in *Conference on Neural Information Processing Systems*, 2014.

[24] C. S. S. W. O. Maaløe, Lars. Sønderby, "Auxiliary deep generative models." in *33rd International Conference on Machine Learning*, 2016.