Monthly Notices
of the
ROYAL ASTRONOMICAL SOCIETY

MNRAS **496,** 2346–2361 (2020)
Advance Access publication 2020 June 16

doi:10.1093/mnras/staa1647

Detecting outliers in astronomical images with deep generative networks

Berta Margalef-Bentabol,[1,2]★ Marc Huertas-Company,[2,3,4,5] Tom Charnock,[6] Carla Margalef-Bentabol,[7] Mariangela Bernardi,[1] Yohan Dubois,[6] Kate Storey-Fisher[8] and Lorenzo Zanisi[9]

[1]*Department of Physics and Astronomy, University of Pennsylvania, Philadelphia, PA 19104, USA*
[2]*LERMA, Observatoire de Paris, PSL Research University, CNRS, Sorbonne Universités, UPMC Univ. Paris 06, F-75014 Paris, France*
[3]*Univeristé de Paris, 5 Rue Thomas Mann, F-75013 Paris, France*
[4]*Departamento de Astrofísica, Universidad de La Laguna, E-38206 La Laguna, Tenerife, Spain*
[5]*Instituto de Astrofísica de Canarias, E-38200 La Laguna, Tenerife, Spain*
[6]*Institut d'Astrophysique de Paris, UMR 7095, CNRS, UPMC Univ. Paris VI, 98 bis boulevard Arago, F-75014 Paris, France*
[7]*Babbly co, Toronto, ON M4M1W7, Canada*
[8]*The Center for Cosmology and Particle Physics, New York University, New York, NY 10003, USA*
[9]*Department of Physics and Astronomy, University of Southampton, Highfield SO17 1BJ, UK*

Accepted 2020 June 7. Received 2020 June 7; in original form 2020 March 4

ABSTRACT

With the advent of future big-data surveys, automated tools for unsupervised discovery are becoming ever more necessary. In this work, we explore the ability of deep generative networks for detecting outliers in astronomical imaging data sets. The main advantage of such generative models is that they are able to learn complex representations directly from the pixel space. Therefore, these methods enable us to look for subtle morphological deviations which are typically missed by more traditional moment-based approaches. We use a generative model to learn a representation of *expected* data defined by the training set and then look for deviations from the learned representation by looking for the best reconstruction of a given object. In this first proof-of-concept work, we apply our method to two different test cases. We first show that from a set of simulated galaxies, we are able to detect ∼90 per cent of merging galaxies if we train our network only with a sample of isolated ones. We then explore how the presented approach can be used to compare observations and hydrodynamic simulations by identifying observed galaxies not well represented in the models. The code used in this is available at https://github.com/carlamb/astronomical-outliers-WGAN.

Key words: software: data analysis – methods: data analysis.

1 INTRODUCTION

In recent years, the amount of astronomical data produced both by observations and simulations has exponentially increased in volume and complexity. This trend is expected to continue in the near future with surveys such as LSST and *EUCLID* becoming available. Processing and extracting meaningful physical information from these new data sets is a new challenge for the community.

To provide the necessary computational relief, machine learning techniques are becoming more and more popular as a way to address the increasing complexity. In particular, supervised machine learning has proven to be very successful when large amounts of labelled or annotated data are available for classification (e.g. Huertas-Company et al. 2015; Cabrera-Vives et al. 2017; Jacobs et al. 2017; Kim & Brunner 2017; Domínguez Sánchez et al. 2018; Huertas-Company et al. 2018; Sreejith et al. 2018; Davidzon et al. 2019), regression (e.g. Tuccillo et al. 2018; Bonjean et al. 2019; Pasquet et al. 2019), and segmentation (e.g. Boucaud et al. 2020).

Supervised algorithms rely on annotated data sets and are thus, not well suited to the discovery of new types of unknown objects which will certainly be present in future surveys. In order to fully unlock the discovery potential of machine learning we have to leverage unlabelled data. Unsupervised algorithms aim to learn the underlying distribution of the data and find patterns without relying on annotated data. Such unsupervised methods can be, hence, used to detect objects whose properties deviate from the *expected* or *normal* objects given a data distribution. These abnormal objects are commonly referred to as outliers or anomalies. Anomaly detection is an active field in machine learning and has a broad range of applications ranging from fraud detection to surveilance (e.g. Frery et al. 2017; Zhang, Vukotic & Gardner 2018) to early diagnosis of disease outbreaks (Wong et al. 2005).

★ E-mail: berta.margalef@gmail.com

In astronomy, outliers can represent artifacts in the data, pipeline errors or new physics. In the case of data or pipeline artifacts, it is important to further analyse them to better reduce systematics. On the other hand, any novel findings can potentially lead to interesting new science (Norris 2017) or objects which differ between models and observations. Several machine learning methods have already been successfully applied to detect outliers in astrophysical data sets, such as unknown classes of objects or objects belonging to rare classes. For example, self-organizing maps have been used to detect unusual quasars (Fustes et al. 2013) and spectroscopic outliers (Meusinger et al. 2012), random forests have been used to detect anomalous SDSS spectra (Baron & Poznanski 2017), one-class support vector machines have been employed for novel detection in the *WISE* survey (Solarz et al. 2017) and clustering algorithms have been utilized to detect anomalous data in light-curves (Protopapas et al. 2006; Giles & Walkowicz 2018).

Anomaly detection in astronomical images is, however, a more complex task because of the high dimensionality (i.e. high number of parameters, in this case the number of pixels in the image) and limited amount of data. Traditional machine learning approaches, such as those mentioned earlier, typically rely on some reduced set of summary statistics (such as photometry, spectroscopic features, or shape measurements), which usually discard a large amount of information about the image complexity and, therefore, can miss some subtle morphological anomalies. Deep generative models, on the other hand, are a modern class of unsupervised methods with the ability to learn complex representations of high-dimensional data in such a way that they can generate new examples drawn approximately from the same distribution as the original data set. Generative Adversarial Networks (GANs; Goodfellow et al. 2014) provide a framework for training such deep generative models. They have gained popularity in recent years for their ability to produce extremely realistic images of everyday scenes (Radford, Metz & Chintala 2015; Karras et al. 2018) and have also been used in astronomy to generate realistic galaxy images (Ravanbakhsh et al. 2017). Recently, work has shown that GANs can be efficiently used for anomaly detection in medical image data sets (Schlegl et al. 2017; Murphy et al. 2018). They present a promising application for anomaly detection in astronomical images. One of the key advantages of a generative model-based approach for anomaly detection is that the model can learn to represent complex data directly from the pixel distribution without relying on specific galaxy properties, that could, otherwise, introduce biases from the methods used to obtain such properties.

This work, as the first in a series of studies, aims to achieve two goals: to test of the ability of generative models to identify outliers in astronomical imaging data sets (i.e. images that are significantly different to the expected or 'normal' data) and their capability to globally discern differences between data sets. To address the first goal, we test the approach in a well-defined sample of simulated galaxies from the cosmological hydrodynamic simulation Horizon-AGN (Dubois et al. 2014). We define isolated galaxies as the normal objects and then quantify how frequently merging galaxies are detected as outliers. We chose this example for the potential to have a large sample of well-defined anomalies (the mergers) and therefore the ability to draw statistically significant conclusions. For that reason, in this first test, we refrain from testing our method with human-labelled anomalies on real data, as the sample of anomalies would not be sufficiently large. For our second task, we explore whether Wasserstein GANs (WGANs, discussed in Section 3.3; Arjovsky, Chintala & Bottou 2017; Gulrajani et al. 2017) can be employed to quantify differences between observations and simulations.

The paper is structured as follows: In Section 2 we present the data we use for this work. In Section 3 we explain the methodology for anomaly detection. Section 4 is devoted to describing different possible applications. And finally, we summarize our findings in Section 5.

2 DATA

For this work, we use both simulated data from the Horizon-AGN cosmological hydrodynamical simulation (Dubois et al. 2014) and observed galaxies from the CANDELS survey (Grogin et al. 2011; Koekemoer et al. 2011).

2.1 Horizon-AGN

We refer the reader to Dubois et al. (2014) for complete details of the simulation suite. Horizon-AGN is a cosmological hydrodynamical simulation run in a $L_{\rm box} = 100\,h^{-1}$ Mpc cube with initial conditions drawn from *WMAP*-7 cosmological parameters (Komatsu et al. 2011). The total volume contains 1024^3 dark matter (DM) particles, with a DM mass resolution of $M_{\rm DM,res} = 8 \times 10^7\,{\rm M}_\odot$. The simulation is run with the adaptive mesh refinement code RAMSES (Teyssier 2002), and the initially uniform grid is refined down to a minimum cell size of 1 kpc constant in physical length. Gas is allowed to cool down to 10^4 K through H and He collisions with a contribution from metals using a Sutherland & Dopita (1993) model. Gas is heated from a uniform UV background after $z_{\rm reion} = 10$ following Haardt & Madau (1996). Star formation occurs in regions where the gas density reaches a critical density of $n_0 = 0.1\,{\rm H\,cm^{-3}}$ and it is modelled with a Schmidt law: $\rho_* = \epsilon_* \rho / t_{\rm ff}$, where ρ_* is the star formation rate density, ρ is the gas density, $\epsilon_* = 0.02$ (e.g. Kennicutt 1998) the constant star formation efficiency and $t_{\rm ff}$ is the local free-fall time of the gas. Stellar feedback is included assuming a Salpeter (1955) initial mass function (IMF), and occurs via stellar winds, supernovae type II and type Ia, with mass, energy, and metal release of six chemical species: O, Fe, C, N, Mg, and Si. Black hole (BH) feedback is also included in the simulation as modelled in Dubois et al. (2012), with BHs releasing energy in a quasar (heating) mode for a high accretion rate (Eddington ratio >0.01) and in radio mode (jet) for low accretion rates (Eddington ratios <0.01).

In Horizon-AGN, galaxies are identified using the ADAPTAHOP structure finder (Aubert, Pichon & Colombi 2004) over the stellar distribution. The merger trees for the identified galaxies are built using the procedure outlined in Tweed et al. (2009), considering 758 time-steps that cover a redshift range spanning from $z = 7$ to $z = 0$ and with a time difference of 17 Myr in average between two successive time-steps.

2.1.1 *Mock images*

From the output of the simulation, we produce mock observations that will be used to train the generative models. In particular, mock images are produced to replicate the properties of the *HST*-CANDELS images in the *H*-band (F160W), using the SUNSET code (e.g. Kaviraj et al. 2017; Laigle et al. 2019), which models the emission of all galaxy particles to produce an image in the observed-frame. For each identified galaxy in the simulation, we define a cubic volume centred around the galaxy with an edge length of eight times the radius of the galaxy (in this case, defined as the average between

the three semi-axes obtained when fitting an ellipsoid to the stellar mass distribution of the galaxy). This volume should contain the stellar particles from the main galaxy as well as those from any close companion, in order to capture any secondary progenitor in the image for the case of galaxy mergers. The stellar particles contained within the volume are used as an input to SUNSET, along with the spectral response of the *H*-band filter of the WFC3 camera. SUNSET computes the fluxes corresponding to the inputs using the stellar models of Bruzual & Charlot (2003) and a Chabrier (2003) IMF. It is assumed that each particle is well described by a simple stellar population, for determining the contribution of each particle to the Spectral Energy Distribution (SED). For this work, we chose not to include dust effects in the image generation for computational reasons. This should not be a problem for tests involving only simulated data but can significantly affect the comparison with observations. We will discuss this in Section 4.2.2. Finally, the integration of the SED in each pixel and the redshift of the galaxy are used to generate an image in the observed frame. The physical size of the pixel is re-scaled for every image to 0.06 arcsec, to match the resolution of the CANDELS *H*-band images. The flux is then scaled using the *H*-band zero-point of CANDELS to match the S/N. Finally, to generate a realistic mock observation, the rescaled images are convolved with the corresponding PSF. These steps are repeated for three different projections along the main axis of the simulations (X,Y,Z) so that for every 3D cube three images are produced. The final sample built that way consists of 1524 118 mock images which include all galaxies with $\log(M_*/\mathrm{M}_\odot) > 10$ and $0.5 < z < 3$, with 250 snapshots in the redshift range. For the purpose of this work, we set a fixed image size of 64×64 pixels.

2.2 CANDELS

We use *H*-band images from the five CANDELS (Grogin et al. 2011; Koekemoer et al. 2011) fields: UDS, COSMOS, GOODS-S, GOODS-N and EGS. The parent sample comes from the catalogue of Dimauro et al. (2018). Our final selection is made of *H*-band selected galaxies with magnitudes brighter than $F160 = 23.5$, $\log(M_*/\mathrm{M}_\odot) > 10$ and $0.5 < z < 3$, to match the sample of galaxies from the Horizon-AGN simulation in stellar mass and redshift. The final sample consists of 17 611 CANDELS images.

We use the official catalogues of redshifts (spectroscopic redshifts are used when available) and stellar masses from CANDELS. The UDS and GOODS-S photometric redshifts were determined using the method described in Dahlen et al. (2013). Stellar masses are drawn from the catalogue presented in Santini et al. (2014) using these photometric redshifts. For the COSMOS, GOODS-N, and EGS fields, the photometric redshifts and stellar masses are discussed in Nayyeri et al. (2017), Barro et al. (2019), and Stefanon et al. (2017), respectively.

We additionally use the structural parameters (Sérsic index, n, effective radius, R_{eff}, and axial ratio, q) published in Dimauro et al. (2018), obtained from 2D single Sérsic fits on the *H*-band (F160W), and the deep-learning based visual morphologies from Huertas-Company et al. (2015).

3 METHOD

3.1 Deep neural networks

Artificial Neural Networks (ANN; Hassoun 1995), are computational techniques vaguely inspired by the connections that are established between the neurons in the brain and their ability to store and process information. An ANN consists of a collection of connected nodes or units (or neurons). The connections (or synapsis) are directed and have associated weights. Those weights are determined by training (or learning).

The nodes of a network are typically arranged in layers. The particular arrangement of the nodes into layers and the connection patterns between them is called the architecture of the neural network. Input layers contain the nodes that receive their input from an external source, output layers provide the output of the network, and the layers in between are referred to as hidden layers. Each node of the hidden layers (hidden unit) is a mathematical function that receives inputs from units in the previous layer and computes an output that is transmitted to other units in the next layer based on the connecting weights. The goal is to use the network as a complex non-linear function that provides some desired output for each input. A cost function or loss is defined to quantify how far is the desired output from the network's actual output. Training is the process of determining the best set of weights to minimize the cost function for a given data set.

Deep-learning Networks (or Deep Networks) are ANNs comprised of many more hidden layers than traditional ANNs and typically have more complex architectures and mathematical functions in their units. Convolutional Neural Networks (CNNs) are a particular architecture of deep networks that were developed within the context of image processing and computer vision applications (Fukushima 1988). CNNs are comprised of one or more convolutional layers followed by one or more fully connected layers. The convolutional layers take as input a set of feature maps (e.g. the colour channels of an image) and convolve each of these with a set of learnable filters to produce the output feature maps. Each layer adds more abstraction to the original input and produces a more informative set of features for the next layer. The fully connected layers have every node in a layer connected to every node in the following layer. They act as a classifier, taking as input the features from the last convolutional layer. The architecture of a CNN is designed to take advantage of the 2D structure of an input image, preserving the spatial relationship between pixels, i.e. they are able to learn translationally invariant features from the data. By exploiting the translational symmetry of the data, CNNs have shown to produce great results for pattern recognition in images.

3.2 Generative advesarial networks, GANs

Generative adversarial networks (GANs) were first introduced in Goodfellow et al. (2014). In the original formulation, they consist of two networks that are trained simultaneously: a generator (the generative model to be trained) and a discriminator (a classification model). The generator, G_θ, with parameters $\boldsymbol{\theta}$, produces new samples from the approximated target data distribution whilst the discriminator, D_ψ, with parameters $\boldsymbol{\psi}$, aims to distinguish the generated samples from the true target distribution.

The input for the generator is a random vector, $\boldsymbol{z}$, usually drawn from a normal or uniform distribution, and the output is drawn from the approximate target distribution $\widetilde{\boldsymbol{x}} = G_\theta(\boldsymbol{z})$, usually an image, although not necessarily so. The discriminator is trained as a standard classifier optimized to distinguish real and generated images. The output of the discriminator describes whether the features are likely to be from the true distribution or not. Once the discriminator is trained to optimize the parameters, ψ, for a given set of θ, the discriminator parameters are fixed, and the generator is trained to maximize the distance from the category designated as a *generated* image. In doing so, the features which distinguish

the two categories in the discriminator are backpropagated through to the generator, allowing the generator to create generated images with the features representative of the real ones. The networks keep training alternately until a Nash equilibrium (Nash 1950) is reached and the generator creates images that are equally categorized by the discriminator as the real ones. The objective of the combined networks can be formulated as the minimax objective of distance $V(P_\mathrm{r}, P_\mathrm{g})$ as follows:

$$V(P_\mathrm{r}, P_\mathrm{g}) = \min_{\boldsymbol{\theta}\in\mathbb{R}^{N_G}} \max_{\boldsymbol{\psi}\in\mathbb{R}^{N_D}} \mathop{\mathbb{E}}_{\boldsymbol{x}\sim P_\mathrm{r}} [\log D_\psi(\boldsymbol{x})] + \mathop{\mathbb{E}}_{z\sim P_z} [\log(1 - D_\psi(G_\theta(\boldsymbol{z})))], \quad (1)$$

where P_r is the real data distribution and P_g is the distribution of samples of generated targets $\widetilde{\boldsymbol{x}} = G_\theta(\boldsymbol{z})$ obtained from the latent distribution $\boldsymbol{z} \sim P_\mathrm{z}$.

3.3 Wasserstein generative adversarial networks, WGANs

GANs have shown unprecedented achievements for many generative tasks, particularly in image generation. However, the original GAN formulation often suffers from convergence problems, when the network fails to find a Nash equilibrium (Salimans et al. 2016), or mode collapse, that results in the generator producing limited varieties of samples. Since GANs were introduced, several improvements have been proposed in the literature that help stability in the training phase (e.g. Salimans et al. 2016; Neyshabur, Bhojanapalli & Chakrabarti 2017; Thanh-Tung, Tran & Venkatesh 2019). One of them is the Wasserstein-GAN (WGAN) model which is based on the Wasserstein-1 distance as the metric to measure the similarity between a real and a generated distribution. This type of network has been shown to be more stable and reach convergence more easily than the original formulation of GANs and prevents mode collapse (Arjovsky et al. 2017). Although similar in style to traditional GANs, WGANs are theoretically separate. In principle, the difference with WGANs is that the discriminator, D_ψ is replaced by another network, often called a critic, C_ψ. Instead of classifying images into real or generated categories, the critic gives an estimation of the Wassestein distance, which describes the amount of work necessary to transport a generated distribution to a target one. In our case, the distribution is the pixels of a collection of images. In times gone by, the Wasserstein distance has also been known as the Monge–Ampère–Kantorovich distance and the Earth-mover distance (EMD). The name EMD arises since one can think of a probability distribution as a pile of earth where the EMD would be the minimal work needed to move one pile to the other. Work is defined as the amount of earth/mass that was moved times the travelled distance. Mathematically, the Wasserstein distance between two probability distributions P_r and P_g can be expressed as the supremum over the set of all 1-Lipschitz functions, $\boldsymbol{C}$, via:

$$W(P_\mathrm{r}, P_\mathrm{g}) = \sup_{C_\psi\in\boldsymbol{C}} \left[\mathop{\mathbb{E}}_{\boldsymbol{x}\sim P_\mathrm{r}} [C_\psi(\boldsymbol{x})] - \mathop{\mathbb{E}}_{z\sim P_\mathrm{z}} [C_\psi(G_\theta(\boldsymbol{z}))] \right]_{\boldsymbol{\theta}=\boldsymbol{\theta}^*}, \quad (2)$$

where $\boldsymbol{\theta}^*$ is some fixed set of parameters of the generator. In order to implement a WGAN, we approximate the 1-Lipschitz functions in equation (2) with a neural network, i.e. the critic, that is trained by maximizing the following cost function:

$$L = \left[\mathop{\mathbb{E}}_{\boldsymbol{x}\sim P_\mathrm{r}} [C_\psi(\boldsymbol{x})] - \mathop{\mathbb{E}}_{z\sim P_\mathrm{z}} [C_\psi(G_\theta(\boldsymbol{z}))] \right]_{\boldsymbol{\theta}=\boldsymbol{\theta}^*}. \quad (3)$$

However, the function C_ψ learned by the critic has to be a 1-Lipschitz function in order to calculate the approximate Wasserstein distance. A differentiable function is 1-Lipschitz if and only if it has gradients with norm at most 1 everywhere. This can be enforced in the WGAN using gradient penalty that penalizes the model if the gradient norm moves away from norm value of order unity, which results in adding a regularization term in the loss function. Therefore, the new loss function for the critic that we maximize has the following form:

$$L_C = \left[\mathop{\mathbb{E}}_{\boldsymbol{x}\sim P_\mathrm{r}} [C_\psi(\boldsymbol{x})] - \mathop{\mathbb{E}}_{z\sim P_\mathrm{z}} [C_\psi(G_\theta(\boldsymbol{z})] + \lambda \mathop{\mathbb{E}}_{\hat{\boldsymbol{x}}\sim P_{\hat{\mathrm{x}}}} [(\|\nabla_{\hat{\boldsymbol{x}}} C_\psi(\hat{\boldsymbol{x}})\|_2 - 1)^2] \right]_{\boldsymbol{\theta}=\boldsymbol{\theta}^*}, \quad (4)$$

where $\hat{\boldsymbol{x}} = \epsilon\boldsymbol{x} + (1-\epsilon)\widetilde{\boldsymbol{x}}$ is uniformly sampled from the straight line between a pair of data points sampled from the distribution of P_r and samples of $\widetilde{\boldsymbol{x}} \equiv G_\theta(\boldsymbol{z})$ with $\boldsymbol{z} \sim P_z$. ϵ is a mixing parameter, uniformly sampled between 0 and 1. λ (the gradient penalty) is a hyperparameter that is, in practice, tuned to achieve optimal performance.

Since the parameters $\boldsymbol{\theta}$ of the generator G_θ do not enter into the first term of equation (3), its derivative with respect to $\boldsymbol{\theta}$ is zero and as such we can define the generator-only loss as:

$$L_G = \left[- \mathop{\mathbb{E}}_{z\sim P_\mathrm{z}} [C_\psi(G_\theta(\boldsymbol{z}))] \right]_{\boldsymbol{\psi}=\boldsymbol{\psi}^*}. \quad (5)$$

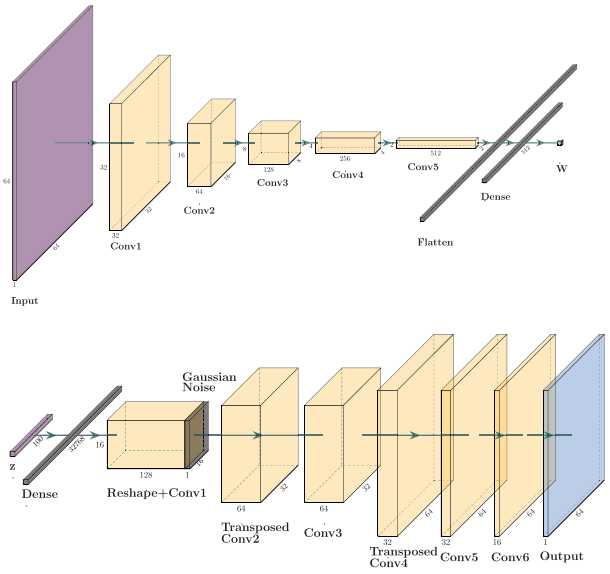

Figure 1. Critic (top) and generator network (bottom). The critic in this work consists of five convolutional layers and two dense layers. It takes as input an image of 64 × 64 pixels and outputs a real number. The generator network is made of one dense layer and six convolutional layers. It takes as an input a random vector of size 100 and outputs an image of size 64 × 64 pixels.

3.4 Training procedure

We implement CNN architectures for the critic and the generator, both shown in Fig. 1. The generator network takes as input a random vector of size 100 and outputs an image of size 64 × 64 pixels. It consists of one dense layer and six convolutional layers. The critic consists of five convolutional layers and two dense layer. It takes as input an image of 64 × 64 pixels and outputs a real number. The final architectures used in this work have been achieved through manual optimization and will not necessarily suit other applications.

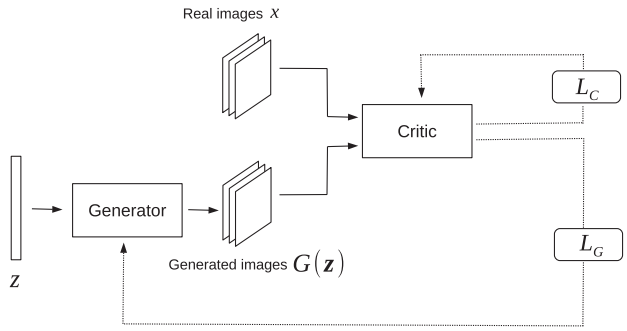

Figure 2. Schematic representation of the WGAN training. Given a batch of real and generated images, the critic is trained for n_{critic} iterations to approximate the Wasserstein distance, by minimizing L_c whilst keeping the weights of the generator fixed. Afterwards, the generator's weights are updated for a single iteration, whilst the critic weights are held constant so that it minimizes the approximate Wasserstein distance.

The WGAN is trained following the standard procedure outlined in Gulrajani et al. (2017). Similar to the original GAN, the two networks of the WGAN are trained alternately. A schematic representation of the WGAN training is shown in Fig. 2. We use a default value of $\lambda = 10$ for the gradient penalty coefficient, a number of critic iterations per generator iteration $n_{\text{critic}} = 10$, batch size $m = 32$ and Adam optimizer with the following hyperparameters: $\alpha = 0.00005$, $\beta_1 = 0.5$, $\beta_2 = 0.9$. We use KERAS[1] with TENSORFLOW[2] as the backend. The exact algorithm is shown in Appendix A.

3.5 Anomaly detection method

Our goal is to use the trained WGAN to detect outliers. The main underlying idea is that after training is completed, the generator G_θ should be able to take a point $\boldsymbol{z}$ from the latent space and generate an image that resembles the images used for training (normal images). However, whenever an image does not come from the distribution of normal images then it will not be possible to generate a similar image from any point, $\{z|z \in \mathbb{R}^{N_z}\}$, in the latent space, $\mathbb{R}^{N_z}$, and it will be in some sense anomalous.

Therefore, in order to identify if a given image $\boldsymbol{x}_t$ is an outlier, we need first to look for the closest image the trained network can generate from the latent space and then quantify the degree of similarity between the generated image $\widetilde{\boldsymbol{x}}' \equiv G_\theta(\boldsymbol{z}')$ and the original one $\boldsymbol{x}_t$. In this work, we follow the method described in Schlegl et al. (2017) to find the $\boldsymbol{z}'$ vector that generates the closest image to a given input image. With the weights of the WGAN fixed, we train a neural network μ_ϕ composed of two fully connected layers that maps a noise vector, $\boldsymbol{y}$, of the same size as the latent space into the actual latent space, $\boldsymbol{z}$. This output is fed to the WGAN to generate an image. The shallow network is optimized using a loss function with two components, a residual loss L_R and a critic loss L_F. The residual loss enforces the visual similarity pixel to pixel between the generated image $G_\theta(\boldsymbol{z})$ and $\boldsymbol{x}_t$. The critic loss pushes the generated image to lie on the learned manifold of trained images (i.e. have the same types of features). The total loss is defined as the weighted sum of both components:

$$L_A = \gamma L_R + (1-\gamma) L_F \tag{6}$$

[1] https://keras.io/
[2] https://www.tensorflow.org/

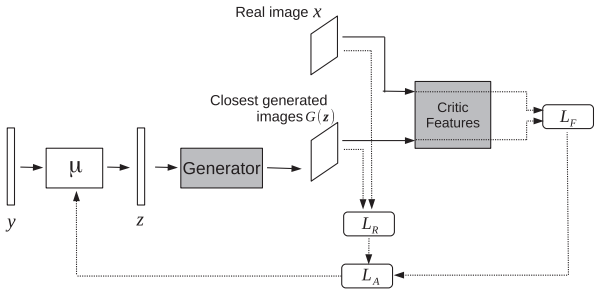

Figure 3. Schematic representation of the anomaly detection training. The grey colour represents that the weights of a network are fixed during training. Given a real image and a noise vector, the network μ finds the anomaly score and the closest generated image by minimizing the combined loss L_A (see equation 6) whilst keeping the weights of the generator and critic fixed. The critic features box represent the critic network without the last two dense layers, and it outputs a feature map obtained before the fully connected layers.

$\gamma \in (0, 1)$ is a hyperparameter that weights the two contributions to the final loss. Here we use a value of $\gamma = 0.7$ (we note that our results do not change significantly when choosing different values of γ). Each contribution is defined as follow:

$$L_R(\boldsymbol{z}') = |\boldsymbol{x} - G_\theta(\boldsymbol{z}')| \tag{7}$$

$$L_F(\boldsymbol{z}') = |c_\varphi(\boldsymbol{x}) - c_\varphi(G_\theta(\boldsymbol{z}'))|, \tag{8}$$

where c_φ is the output of the last convolutional layer of the critic, C_ψ, i.e. the set of informative features obtained before the fully connected layers. A schematic representation of the training for the network μ_ϕ is shown in Fig. 3.

An anomaly score AS is then defined as the loss at the last iteration, when the training has converged (i.e. the loss is not decreasing any further; in this case convergence is reached after about 500 iterations) and the closest image in terms of equation (6) has been found:

$$AS = \gamma L_R^o + (1-\gamma) L_F^o, \tag{9}$$

where L_R^0 and L_F^0 are the residual and critic loss at the last iteration, respectively.

The procedure is not optimal from a performance perspective since images need to be processed individually. It can easily be improved by performing a global optimization over multiple images and then applying a simple gradient descent to refine as shown in Storey-Fisher et al. (in preparation). Since computing time is not critical for this work in which the sample of images to test is not enormous, we keep this original implementation.

Note that, whilst the anomaly score does not have a true meaning independent of the training of each of the critic, the generator and μ_ϕ, anomalous data can be identified by comparing it to the distribution of the training data, which the generator is trained to approximately draw from. For this reason, we measure the anomaly score for all the images in our training set which acts as the calibration of the anomaly detector. Any new image with an anomaly score significantly outside the bulk of scores for the training images is then quantified as being, in some way, anomalous.

4 APPLICATIONS

In the following section, we explore several cases in astronomy for our WGAN-based anomaly detector. We first calibrate how the anomaly detector performs with a sample of known anomalies.

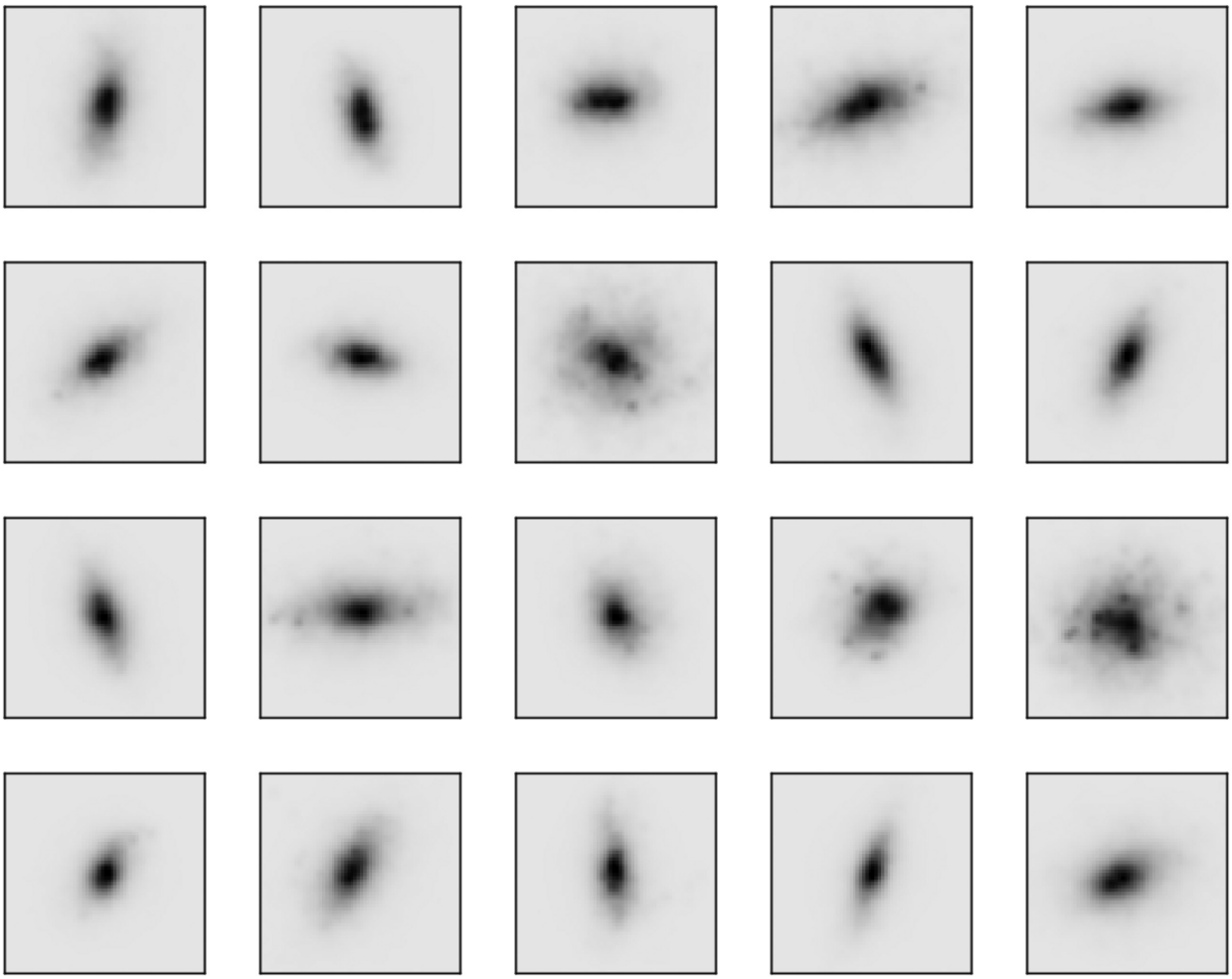

Figure 4. Examples of mock images (observed in the *H*-band) from the Horizon-AGN simulation that are used as training data. The images have a size of 64 × 64 pixels, and the pixel scale is 0.06 arcsec pixel^{-1}.

In particular, we quantify how accurately images of mergers are detected as anomalous when our training sample (the 'normal' sample) is a set of isolated galaxies. We split this application in two cases: the first case focuses on detecting anomalies due to the presence of a neighbour object, and the second one focuses on analysing more subtle merger-induced morphological disturbances. The last application consists of using the anomaly score to compare a sample of images from the Horizon-AGN simulation to real galaxies from the CANDELS survey, to quantify the difference or similarity between such data sets.

4.1 Galaxy mergers as anomalies

4.1.1 Training

The training set consists of images of isolated galaxies (no mergers), which we call the 'normal images', and the test set consists of images of galaxy mergers. The selection of interacting galaxies in the simulations is done by checking an increase in galaxy mass due to the contribution of more than one progenitor from the previous time-step (Rodriguez-Gomez et al. 2015; Abruzzo et al. 2018). If a galaxy has more than one progenitor and the ratio between the stellar mass provided by the secondary and the main progenitor is equal or larger than 1:10, then that galaxy will be considered a merger. When a merger is identified, we build a *merger sequence* by going back in time in the merger tree until the companion is four effective radii away from the central galaxy. We call all these images pre-mergers. We also follow forward in time after the merger event for the same number of time-steps. These images are called post-mergers.

Isolated galaxies, on the contrary, satisfy the condition of having only one progenitor when going back in time 1 Gyr and only one descendant when going forward in time 1 Gyr. Images for both data sets are generated as explained in Caro (2018). The final training sample is made of 531 922 isolated galaxies. Examples of these images are shown in Fig. 4. We also show the stellar mass and redshift distributions for our training and test samples in Fig. 5. Notice that the distributions are significantly different given the restrictive constraints used to define the sample of isolated galaxies. By imposing no interactions in a 2 Gyr time window we remove very massive galaxies from the sample. This is not a problem since we are aiming to calibrate the sensitivity of the WGAN anomaly detector in identifying out-of-distribution objects. It should be noted that mergers are not anomalous scientifically speaking but are outside of our training data and as such we want to detect them.

We train a WGAN network, described in Section 3.3 for $\sim\times10^6$ epochs using only isolated images. As a first exploratory step, Fig. 6 shows some examples from the test set indicating their anomaly score, the closest generated image, and the residual image

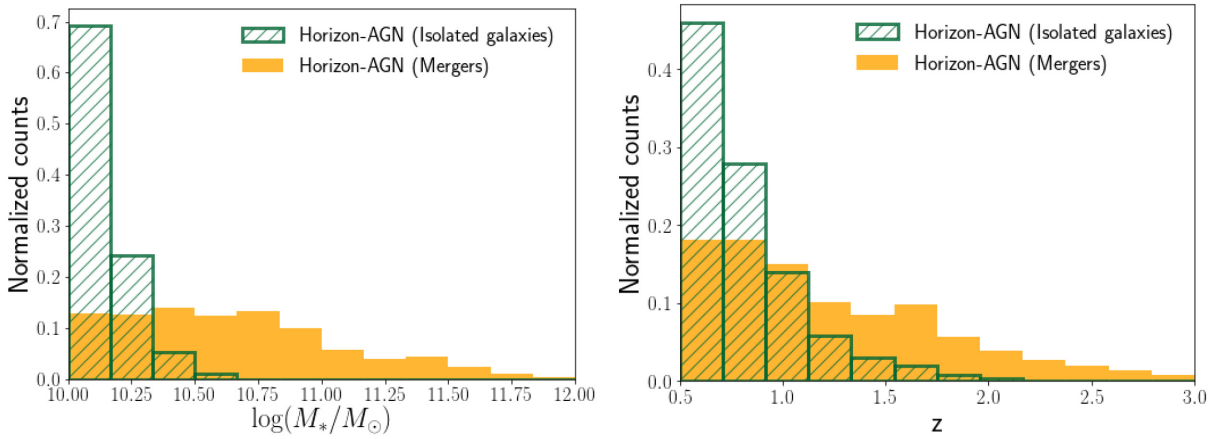

Figure 5. Stellar mass (left-hand panel) and redshift distribution (right-hand panel) of the training data (isolated galaxies in striped green and mergers images in solid yellow). We can see obvious deviations between the distributions of isolated galaxies and mergers which should be quantifiable as anomalous by the anomaly WGAN detector.

(derived by subtracting the closest generated image to the original image). We see that for images with low anomaly score values, the residuals are low as expected, due to the network being able to generate very similar images. For high anomaly score images, the residuals are larger because the network cannot generate a similar image. In several cases, the anomaly is due to the presence of a secondary source as one would naturally expect. Fig. 7 shows a 2D representation of the last layer of the critic (which is used to compute L_F) computed with t-Distributed Stochastic Neighbour Embedding (t-SNE; van der Maaten & Hinton 2008). t-SNE is a technique for dimensionality reduction that helps with the visualization of high-dimensional data sets. In the high dimensional space, it models the probability distribution that dictates the relationships between neighbours around each point. Then in the low-dimensional space, it recreates, as close as possible, the same distribution. When points are close to one another in the high-dimensional space they will tend to be close to one another in the low-dimensional space as well. Notice that the axis of the low-dimensional space are arbitrary and have no particular interpretation. We show a subsample of normal galaxies and a subsample of galaxies with a neighbour in the image. This visualization suggests that the network is well trained for our purpose, and the critic is, indeed, able to separate these two classes well. Therefore, using the critic as part of the anomaly score should provide information with which we can detect anomalies. The next sections quantify the performance.

4.1.2 Results: anomalies caused by a companion in the image

We now quantify how accurately we can detect anomalies due to the presence of a companion in the image. Given that our training sample consists of isolated galaxies, we can assume that the presence of a secondary object will contribute the most to the degree of anomaly on an image when compared with the training sample.

We, therefore, select all the images with at least a secondary object in the image (this will be our known anomalous sample in this application). All these images belong to the pre-merger phase. However, not all pre-mergers have a secondary object in the image. This is because, in order to speed up the training process and due to memory capability, we have cropped the original images from the simulations to 64 × 64 pixels, and therefore, in some cases, we have artificially removed the companion object that will eventually merge with the main galaxy.

We compute the anomaly score for the training sample as well as for the test sample and compare their distributions in Fig. 8. The figure clearly shows that the AS distribution for mergers peaks at larger values than the one for isolated galaxies. To quantify the anomaly detection method, we define a threshold-based method. Images that have an AS larger than the threshold are considered anomalous (or inconsistent with the training sample), while images with AS lower than the threshold, are considered 'normal' (or consistent). We use three different thresholds defined as the value that contains 85, 90, and 95 per cent of the training galaxies within the generative distribution. Using these thresholds, we find 86, 80, and 67 per cent of the anomalous samples are correctly identified as anomalies, respectively. We compare our results with a more traditional method of outlier detection, the k-means clustering method (MacQueen 1967). For that, we use non-parametric measures of structure used to quantify the broad morphology: CAS (concentration C, asymmetry A and clumpiness S; Conselice 2003), and Gini/M20 parameters (e.g. Abraham, van den Bergh & Nair 2003; Lotz, Primack & Madau 2004). We calculate these parameters for the two samples (training and test) using the code STATMORPH, a PYTHON package for calculating non-parametric morphological diagnostics of galaxy images (Rodriguez-Gomez et al. 2019), and apply the k-means method with two clusters. We find that 99 per cent of the training sample belongs to one cluster while 74 per cent of the pre-mergers with a neighbour lie in the other cluster. These results are summarized in Table 1.

We further investigate the incorrectly classified pre-merger galaxies, and find that the majority of these are caused by a high flux ratio between the main galaxy and the brightest neighbour. We hence compute the flux ratio F_r between the main galaxy and the companion using SEXTRACTOR (Bertin & Arnouts 1996) and then divide the test sample into three bins depending on the flux ratio ($F_r < 1.5$, $1.5 < F_r < 2$, $F_r > 2$). For flux ratios lower than 1.5, we find that 96, 93, and 86 per cent are correctly classified as anomalous, using the three thresholds mentioned above, respectively. For galaxies between 1.5 and 2 times as bright, there is only a small decrease in these percentages (94, 90, and 76 per cent). It is only when the companion is 2 times fainter than the central galaxy (46 per cent of our test sample) that the percentage of galaxies correctly classified drops to 75, 66, and 48 per cent, respectively, for the three

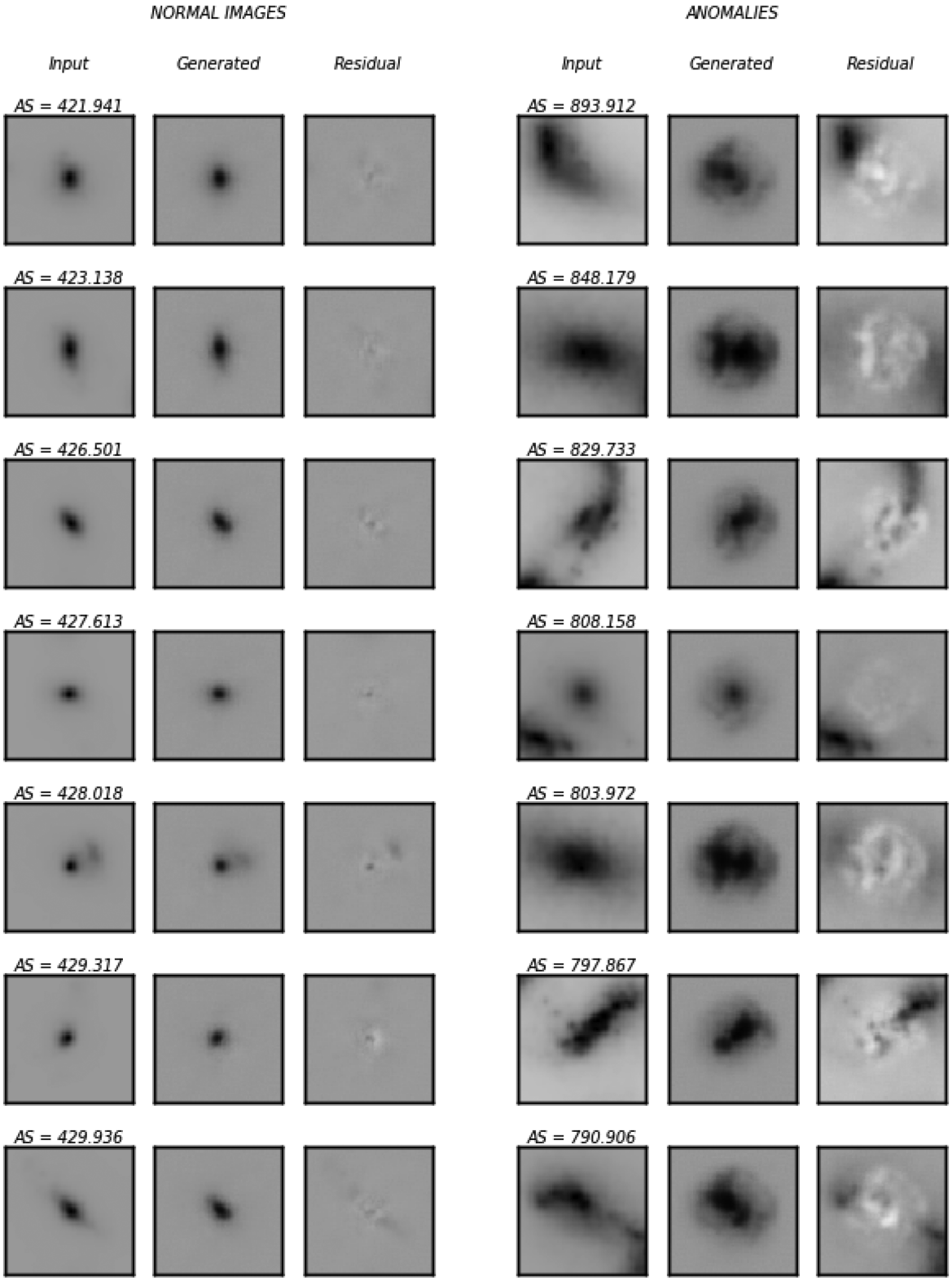

Figure 6. Examples of images draw from the test sample with low anomaly score (left-hand panel) and high anomaly score (right-hand panel). In the first column, we show the input image, in the second column, the closest generated image obtained in the anomaly detection method, and in the third column, the residual (pixel by pixel difference) between the input and the generated. The normal images show low anomaly score values and a very similar image can be generated by the network. For the test images the anomaly score is high and no close image can be generated which results in large residuals.

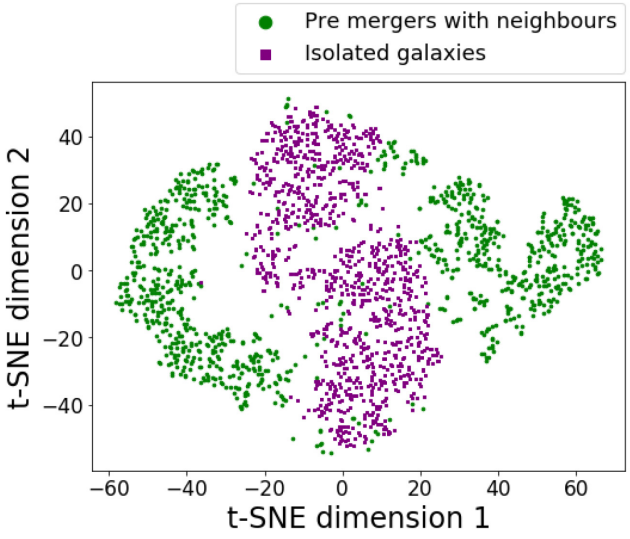

Figure 7. Output of the last convolutional layer of the trained critic after reducing it to two dimensions using the t-Distributed Stochastic Neighbour Embedding method (noting that this is just one realization of the t-SNE). We show where normal images (purple) and images of galaxies with neighbours (green, images that are most different to the normal set) lay in this plane.

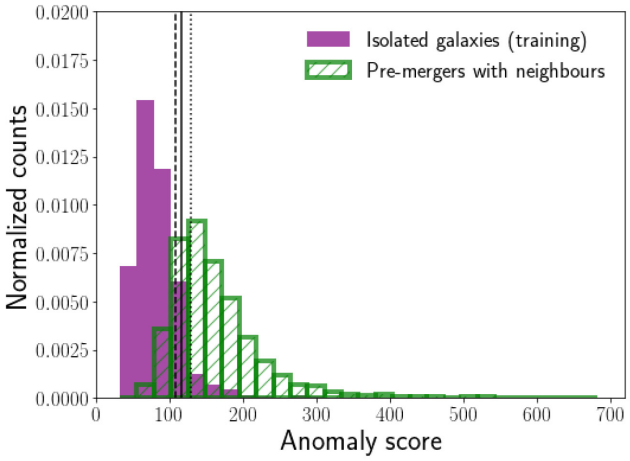

Figure 8. Anomaly score distribution for the isolated galaxies (training data, purple) and the pre-mergers with neighbours (green). The black dash, solid, and dotted line represent the three thresholds defined as the value that, respectively, contain 85, 90, and 95 per cent of the training galaxies within the generative distributions. The distribution for the pre-mergers peaks at a higher value than the training data, with clear separations between the distribution. This indicates that the majority of pre-mergers with neighbours are inconsistent with the training sample.

different thresholds. We find that while for $F_r < 1.5$ and $1.5 < F_r < 2$, our results are comparable to the k-means method (91 and 93 per cent of the test sample are correctly classified according to the k-means method), when considering galaxies with a companion two times fainter, the k-means method performs significantly worse at detecting them as anomalous (only 39 per cent, compared with 75, 66, and 48 per cent, respectively, for the three different thresholds used in our method). These results are summarized in Table 2. We have also explored whether the distance between the main galaxy and the companion has an effect on the anomaly score, but we have found no correlation.

Table 1. Accuracy of the threshold-based anomaly detector for different thresholds, and of k-means anomaly detector method. Each method (columns) is evaluated for test sets (rows) of isolated and pre-merger galaxies. Thresholds 1, 2, and 3 represent the three thresholds defined as the value that contains, respectively, 85, 90, and 95 per cent of the training galaxies within the generative distribution. The last column indicates the accuracy according to the k-means clustering method.

	Accuracy of the anomaly detector			
	Threshold			
	1	2	3	k-means
Isolated	85 per cent	90 per cent	95 per cent	99 per cent
Pre-mergers	86 per cent	80 per cent	67 per cent	74 per cent

Table 2. Accuracy of the threshold-based anomaly detector for different thresholds, and of k-means anomaly detector method. Each method (columns) is evaluated for test sets (rows) of pre-merger galaxies divided according to their flux ratio between the central galaxy and the brightest neighbour (F_c/F_n). Thresholds 1, 2, and 3 represent the three thresholds defined as the value that contains, respectively, 85, 90, and 95 per cent of the training galaxies within the generative distribution. The last column indicates the accuracy according to the k-means clustering method.

	Accuracy of the anomaly detector			
	Threshold			
	1	2	3	k-means
$F_r < 1.5$	96 per cent	93 per cent	86 per cent	91 per cent
$1.5 < F_r < 2$	94 per cent	90 per cent	76 per cent	93 per cent
$F_r > 2$	75 per cent	66 per cent	48 per cent	39 per cent

4.1.3 Results: anomalies caused by merger induced morphological perturbations

In the previous section, we have seen how the anomaly detection method is able to detect anomalous galaxies with high accuracy when a relatively bright companion is found in the image. Here we investigate how accurately the method works when the companion is not present (i.e. for the pre-merger galaxies that do not show a neighbour object in the image and images of the post-merger phases). This exercise is intended to test the robustness of the WGAN-based anomaly detector given more subtle merger-induced perturbations in the main galaxy light distribution. For this application our anomalous sample, therefore, consists of galaxies in the pre-merger phase that do not have a neighbour object as well as galaxies in the post-merger phases.

Fig. 9 shows the anomaly score distributions for isolated pre-mergers (or pre-mergers without neighbours) and post-mergers. As expected, we observe that even if both samples have anomaly score distributions extending to larger values than the training set, the majority of the samples overlap with the training data. Using the same three thresholds as before, 46, 39, and 30 per cent of these images are classified correctly (respectively for the three thresholds), while only 8 per cent when using the k-means clustering method. Overall, it appears the isolated pre-mergers are detected as anomalous slightly more often than the post-mergers. We summarize these results in Table 3. These results indicate that the presence of a secondary object is not the only cause of anomaly in these images, although it is the dominating factor, and that the WGAN is able to detect more subtle morphological differences between the samples. Our results also demonstrate the WGAN method is better suited to finding outliers than traditional clustering methods, particularly when the differences with the training sample

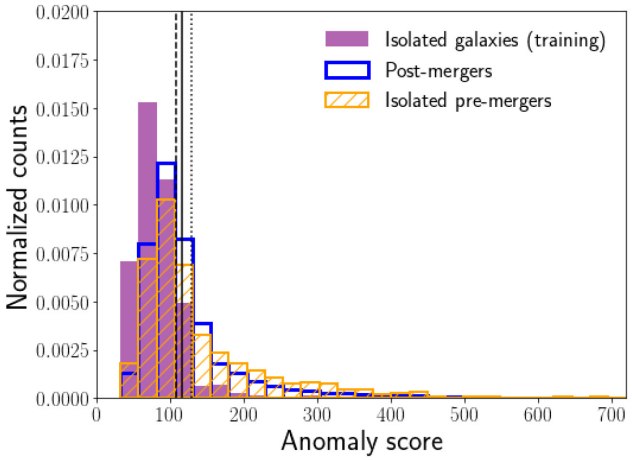

Figure 9. Anomaly score distribution for the isolated galaxies (training data, purple), the isolated pre-mergers (orange), and post-mergers (blue). The black dash, solid, and dotted line represent the three thresholds defined as the value that contain, respectively, 85, 90, and 95 per cent of the training galaxies within the generative distribution. The distribution for the isolated pre-mergers and post-mergers (test sample) extend to larger values than the training data, even though there is significant overlap between the distribution. There are many samples which are inconsistent with the training data.

Table 3. Accuracy of the threshold-based anomaly detector for different thresholds, and of k-means anomaly detector method. Each method (columns) is evaluated for test sets (rows) of isolated pre-mergers, post-merger galaxies, and the combinations of both sets. Thresholds 1, 2, and 3 represent the three thresholds defined as the value that contain, respectively, 85, 90, and 95 per cent of the training galaxies within the generative distribution. The last column indicates the accuracy according to the k-means clustering method.

	Accuracy of the anomaly detector			
	Threshold			
	1	2	3	k-means
Isolated pre-mergers+Post-mergers	46 per cent	39 per cent	30 per cent	8 per cent
Isolated pre-mergers	52 per cent	45 per cent	37 per cent	7 per cent
Post-mergers	45 per cent	38 per cent	28 per cent	9 per cent

are subtle and cannot be fully described by global measures, as they are in this case. We further investigate what features describe the difference between the anomalous images and the training set.

Fig. 10 shows the fraction of samples outside of the training set as a function of the position of the image on the merger sequence. The time is normalized such as that $t = -1$ shows the time at which the companion is at four effective radii from the central galaxy, $t = 0$ is the time at which the two galaxies become one in the merger tree. We distinguish between pre-mergers with and without neighbours for comparison. Images in which the neighbour is present are the most anomalous, as seen in the previous section. But the fraction does not change significantly with time from the merger. The fraction of anomalous images for the pre-mergers without companions remains mostly constant as well, at around 45 per cent. In the merger phase, the percentage of samples outside of the training set reaches 56 per cent and decreases with time from the merger at about 35 per cent as the system relaxes. The trends we observe prevail even when we use different thresholds, and only differ by a scaling factor.

We now investigate other properties from the simulations that can have an effect on the anomaly score: stellar mass, redshift z, effective radius $R_{\rm eff}$, and axial ratio q. The axial ratio is derived using SEXTRACTOR, and the other properties are defined by the

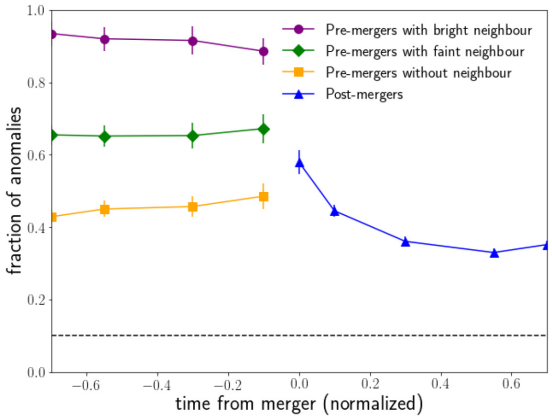

Figure 10. Fraction of samples outside of the training set as a function of time from merger of different types of galaxies. Pre-mergers with a bright neighbour ($F_c/F_n < 2$) in purple, pre-mergers with a faint neighbour ($F_c/F_n > 2$) in green, isolated pre-mergers in yellow and post-mergers in blue. We show the fraction of samples outside of the training set according to the threshold 2, which corresponds to 90 per cent of the training galaxies being consistent.

simulations. We show in Fig. 11 how the fraction of samples inconsistent with the training set change as a function of these properties (we again choose the threshold 2 but note that the choice of the threshold does not affect the trends we observe, they only change by a scaling factor). We observe that there are more anomalous galaxies in the pre-merger stage than in the post-merger phase and that galaxies with a secondary object have, for all properties, higher fractions of anomalies. However, regardless of the galaxy being in the pre- or post-merger phase, or having or not a secondary object, the fraction of anomalies increases with increasing stellar mass and size (larger and more massive galaxies tend to be less consistent with the training set). This is not surprising, as the stellar mass distribution of the training sample does not expand to masses larger than $\log(M_*/M_\odot) = 10.75$, while for the test sample we find galaxies with stellar mass up to $\log(M_*/M_\odot) = 12$ (see Fig. 5). The anomaly score decreases with axial ratio (the most elongated galaxies are more anomalous). This can be explained by the absence of very elongated galaxies in our training sample. Lastly, we observe that there is not a strong correlation with redshift.

One interesting fact to note here is that the distribution of test images is really being compared to the distribution of training images, and as such, by choosing a hard threshold we quantify rare objects (even in the training set) as being less consistent with the bulk of the rest of training set.

4.2 Comparison between observations and simulations

This last application focuses on investigating how we can use the anomaly detection method to compare two sets of data. Here we compare simulated data from Horizon-AGN and data from the CANDELS survey described in Section 2.2 to see if the WGAN is able to distinguish the images coming from different distributions. Assessing how well modern hydrodynamical simulations reproduce the observed properties of galaxies is a complex task because of the large number of parameters involved. The proposed approach has the advantage of collapsing all properties to one unique metric of similarity that encapsulates all morphological features.

For this application, the first thing we do is to add realistic noise to the mock observations to be able to compare the two data sets directly. We first select sky-only regions from the CANDELS-*HST*

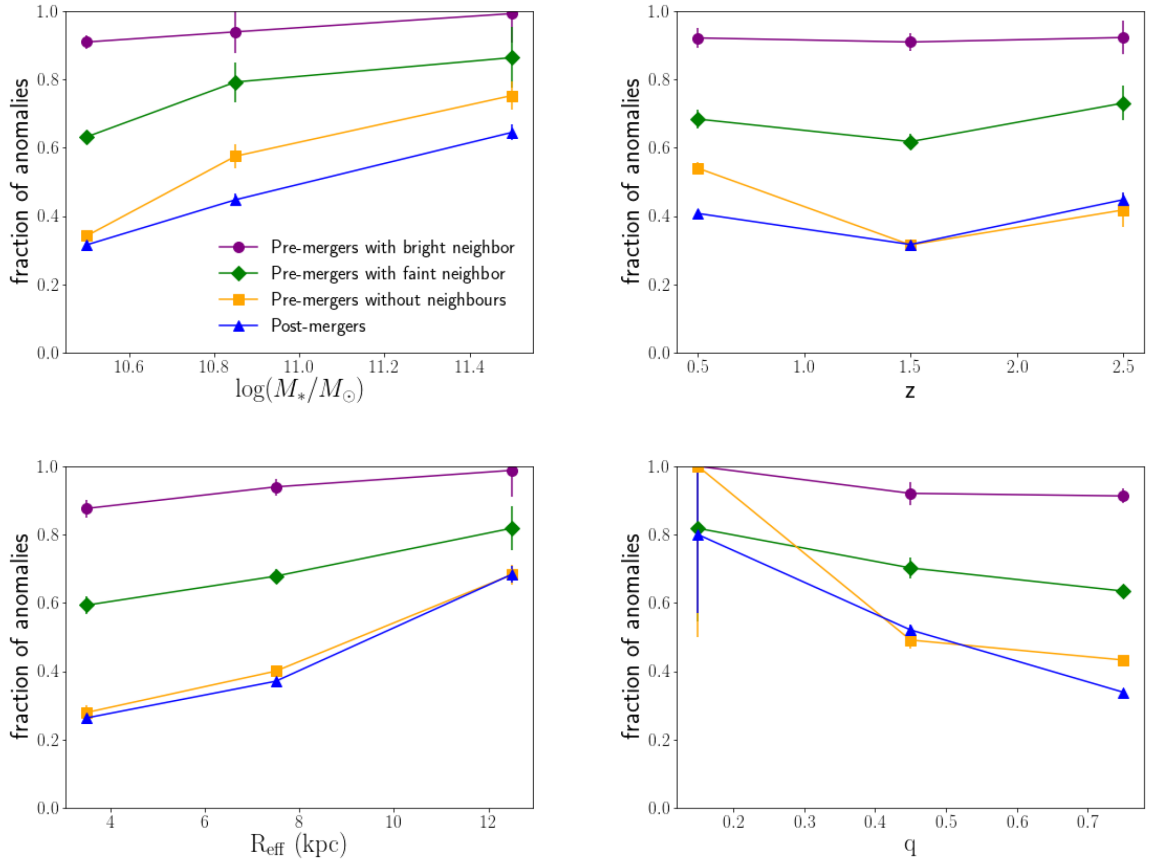

Figure 11. Fraction of samples outside of the training set as a function of stellar mass (top left-hand panel), redshift z (top right-hand panel), effective radius $R_{\rm eff}$ (bottom left-hand panel), and axial ratio q (bottom right-hand panel). In blue we show the fraction of post-mergers, in yellow the fraction of pre-mergers without a neighbour in the image, in purple the pre-mergers with a bright neighbour and in green pre-mergers with a faint neighbour present in the image. The fractions of anomalies increase with mass and size and decreases with axial ratio, while remains constant with redshift.

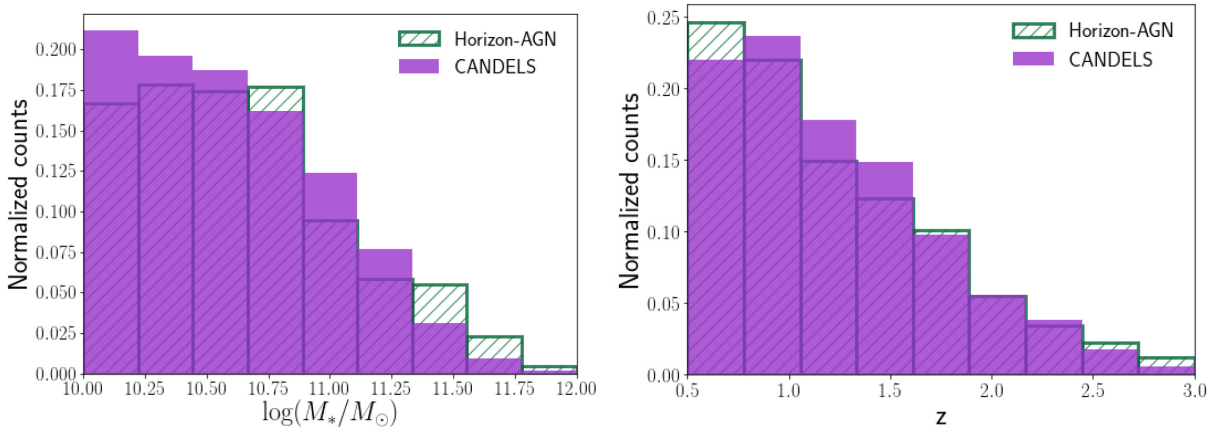

Figure 12. Stellar mass (left-hand panel) and redshift distribution (right-hand panel) of the training data (Horizon-AGN images, in striped green), and the test data (CANDELS images, in solid purple).

observation in the *H*-band, to create noise-only mosaics of 64 × 64 to use randomly with each galaxy from the simulations. For a given galaxy image, we then generate a corresponding Poisson noise, to which we introduced pixel-to-pixel correlation such that 1D autocorrelation power spectral density (PDS) of the sky mosaic matches the source noise. Finally, we add this correlated Poisson noise and a sky mosaic to generate mock images that match the CANDELS-*HST* observations.

4.2.1 Training

We train our WGAN (see Section 3.3) with a training sample composed of all the mock images from the Horizon-AGN simulation (1524 118 images), with added noise. For this application, the test set is comprised of the 17 611 CANDELS images in our sample. The stellar mass and redshift distributions for the training (Horizon-AGN) and test set (CANDELS) are shown in Fig. 12. Both sets

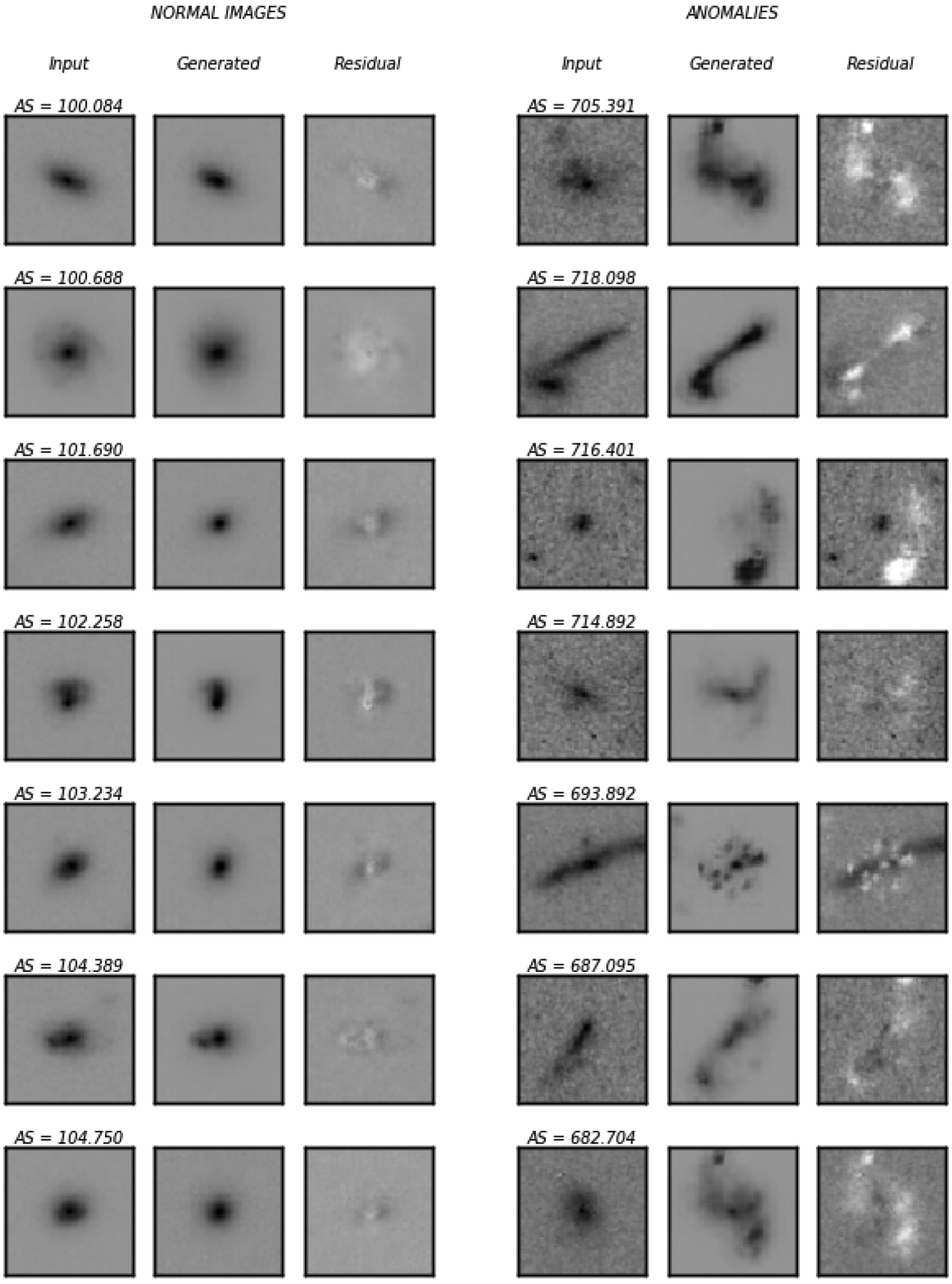

Figure 13. Examples of CANDELS galaxies classified as normal (left-hand panels) and anomalous (right-hand panels). Similar to Fig. 6, in the first column we show the input image, in the second column, the closest generated image, and in the third, the residual image.

cover the same range in stellar mass and redshift, although with a very slight difference in distribution, so any difference between the data sets is not expected to be a consequence of the selection function.

Fig. 13 shows examples of CANDELS galaxies indicating their anomaly score, the closest generated image and the residuals. As in Fig. 6, we observe that for galaxies with low anomaly score the network is able to generate a very similar image and, therefore,

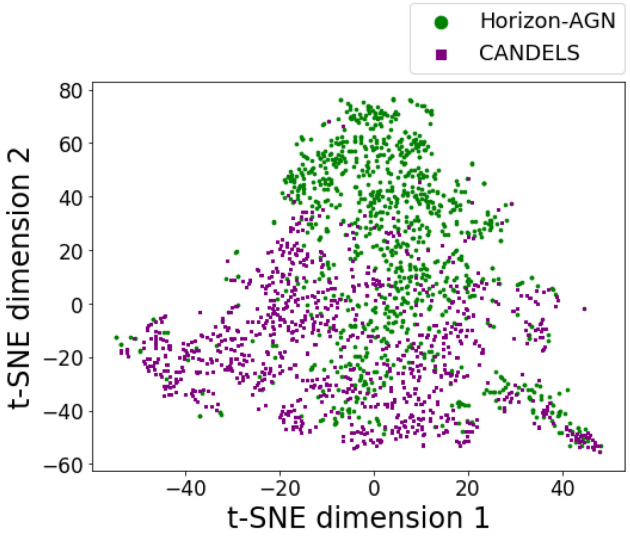

Figure 14. t-SNE space for the output of the last convolutional layer of the trained critic. We show in green a subset of training images and in purple CANDELS images. Note the separation is much less obvious than with Fig. 7 since the distribution of images is much closer.

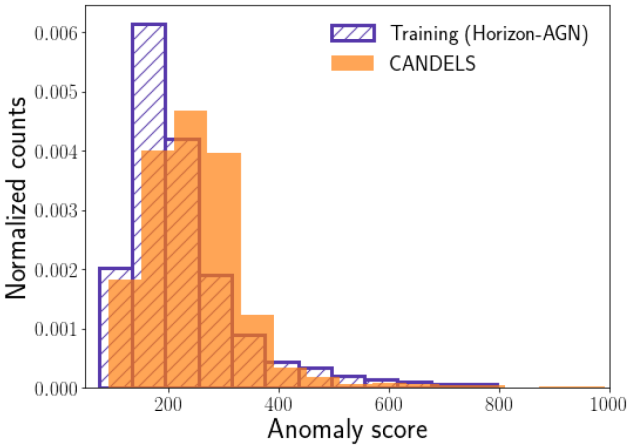

Figure 15. Distribution of anomaly scores for the Horizon-AGN galaxies (training sample), in striped blue, and for the CANDELS data in solid orange. The anomaly scores for the CANDELS galaxies are, overall, higher than the training set, which points at differences between the Horizon-AGN simulations and the CANDELS observations.

the residuals are low, while high anomaly scores result in high residuals (the network is not able to generate similar images). Fig. 14 shows the t-SNE reduced dimensional representation of the output of the last layer of convolutions for both the Horizon-AGN and CANDELS images. We easily observe that the two sets globally populate different parts of the space, although with some degree of overlap, which suggests that the network is able to distinguish the two populations. We investigate the reasons behind this apparent discrepancy in the following section.

4.2.2 Results: Difference between Horizon-AGN and CANDELS

We compute the anomaly scores for Horizon-AGN and CANDELS images, and show their distribution in Fig. 15.

We observe that, even though there is considerable overlap between the two distributions, the anomaly scores for the CANDELS are, overall, higher. If the simulations were to reproduce the observed data perfectly, we would expect the distributions of the anomaly score for both the Horizon-AGN and CANDELS samples to be more consistent. However, the difference in anomaly score distribution suggests that the simulations are not able to completely reproduce the observational data from CANDELS. This could be, in part, because we did not include effects such as dust, but could also include other choices in the physical model of the simulation, or by resolution effects or other prescriptions in the radiative transfer code. Therefore this example has to be seen as an illustration of the potential of this approach to detect global subtle morphological differences between data sets coming from different origins. However, we do not aim to establish robust conclusions given the many limitations. One possible application would be exploring how different simulations, produced with different physical processes, compare with observational data. For that purpose, our WGAN could be trained on images from an observational survey, and then, anomaly scores can be computed for the observational images, and for the images from the different simulations. Comparing the distributions of anomaly scores will give information about which simulations produce images that are more consistent with the observations.

As a preliminary step forward, we show in Fig. 16 the anomaly score distribution for observed galaxies as a function of different galaxy physical properties (stellar mass, effective radius, redshift, axial ratio, Sérsic index, and morphological type). The figure reveals some interesting trends. While there is no effect on the anomaly score due to redshift, globally speaking, massive galaxies tend to be more anomalous. The smallest galaxies tend to have higher AS values, possible due to resolution effects in the simulations. Spheroidal, high Sérsic index galaxies, and point-source/compact are also skewed towards larger AS values. This suggests that compact galaxies might not be well represented in the Horizon-AGN simulation. However, this needs to be investigated further given the limitations of this comparison.

5 SUMMARY

In this first proof-of-concept work, we have explored generative methods as a way to quantify anomalous objects in astronomical imaging data without labels. The method consists of, first, training a WGAN with 'normal' data and calculating the anomaly score of the test sample to quantify the degree of anomaly. The main advantage of such an approach is that it can learn complex representations directly from the pixel space without manual extraction of specific features. It can therefore identify subtle morphological differences and collapse morphological comparisons to one unique metric.

We have tested the method on three different applications:

(i) In the first application we asses how accurately we can detect known differences between the training and test samples. In this case, the WGAN is trained with images of isolated galaxies (with no merger history) from the Horizon-AGN simulation, and used to identify known mergers. We show that the WGAN correctly classifies 80 per cent of the test images as anomalous with a contamination of only 10 per cent (i.e. 10 per cent of the isolated galaxies are incorrectly classified as anomalous). The percentage of anomalous images increases to 92 per cent when not considering images with very faint neighbours in the image.

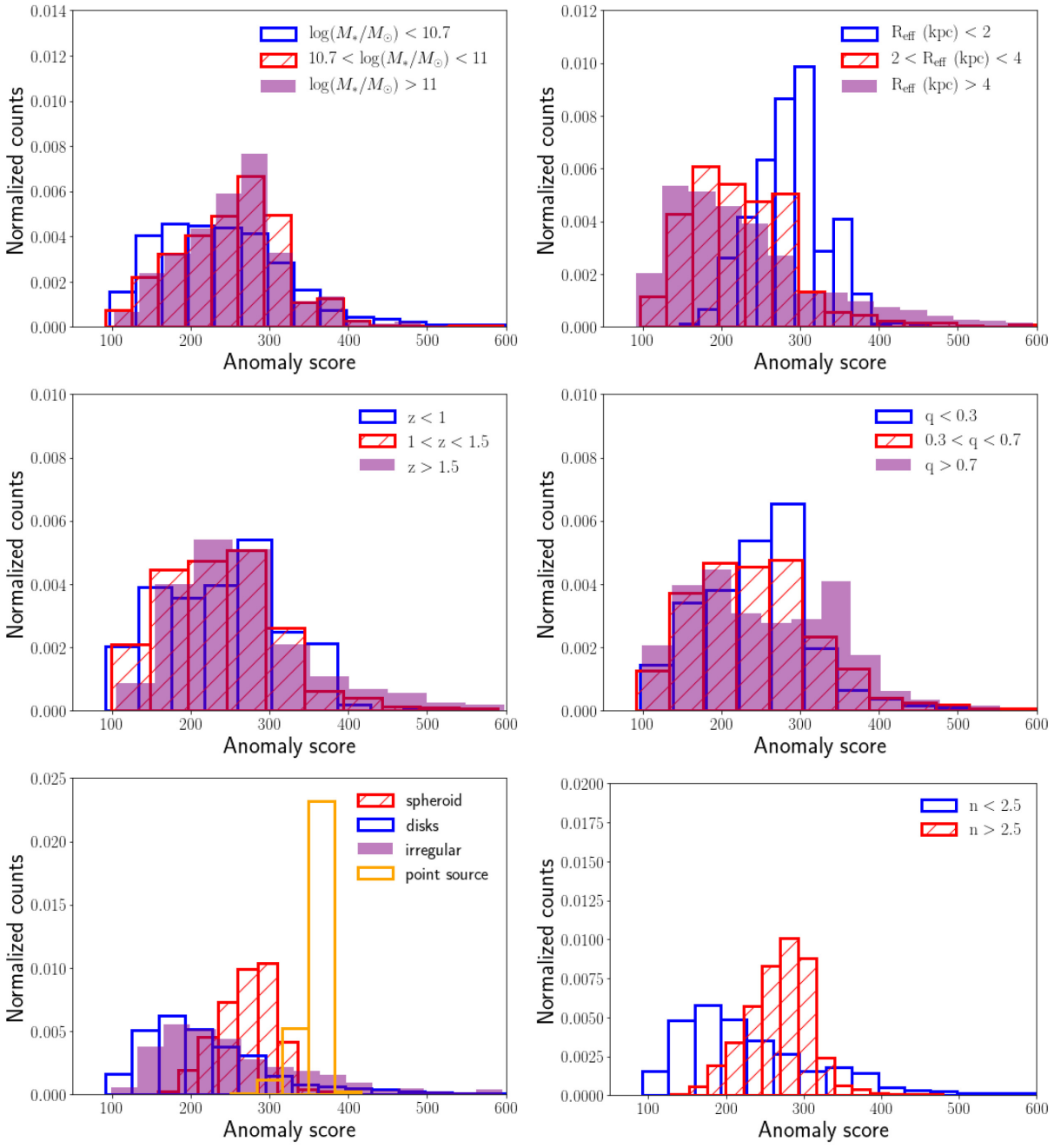

Figure 16. Distribution of the anomaly score for the CANDELS galaxies as a function of different properties (stellar mass, effective radius, redshift, axial ratio, Sérsic index, and morphology type). Each histogram is normalized. These plots show which properties are causing high anomaly scores. We observe that the largest and more massive galaxies tend to have higher anomaly scores, as well as spheroidal galaxies and galaxies with high Sérsic index.

(ii) In the second application we investigate how our method is able to detect anomalies caused by more subtle properties. When investigating a test sample that consist of images of merging galaxies without a visible companion in the image, we find that 45 per cent of the test set is anomalous compared to the training set. In this case, the anomaly is cause by morphological features instead of a secondary source in the image. We observe that the most anomalous objects generally have higher mass and have high axial ratio. This is because the training sample lacks galaxies with these properties. It is, therefore, useful to consider how this anomaly detection method really allows us to introspect biases in the training set as well as the physical model with which we can generate such *realistic* training sets.

(iii) The third application shows how the anomaly detection method can be used to compare two data sets. The training set for this example is a complete set of simulated galaxies from the Horizon-AGN simulations, and the test or comparison sample comprises observed images from the CANDELS survey. We show that the anomaly score distribution of the observations tends to peak at larger values compared to that of the simulated data. We further explore what properties were causing the main differences, to better understand how the simulations differ from the observations. We observe that the simulations were not reproducing the smallest galaxies and high Sérsic index galaxies well. This may be, in part, due to the lack of dust treatment in the radiative code, but

could also be due to the resolution effects and/or other radiative processes.

The code to train our WGAN and generate is made public with this work. In future papers we plan to investigate the effect that physical processes from the simulations (such as the addition of dust) have on our analysis when comparing the Horizon-AGN simulation to the CANDELS survey. Additionally, we plan to use the WGAN anomaly detector to look for outliers in the HSC survey (Storey-Fisher et al. in preparation) and investigate its applicability in the pipelines of future surveys such as LSST and EUCLID. As part of the efforts to investigate the practical use of generative models to compare simulations to observations, we are also exploring regressive models (Zanisi et al. in preparation).

ACKNOWLEDGEMENTS

This work was supported in part by NSF (National Science Foundation) grant AST-1816330.

REFERENCES

Abraham R. G., van den Bergh S., Nair P., 2003, ApJ, 588, 218
Abruzzo M. W., Narayanan D., Davé R., Thompson R., 2018, preprint (arXiv:1803.02374)
Arjovsky M., Chintala S., Bottou L., 2017, in Precup D., Teh Y. W., eds, Proc. 34th International Conference on Machine Learning. International Convention Centre, Sydney, p. 214
Aubert D., Pichon C., Colombi S., 2004, MNRAS, 352, 376
Baron D., Poznanski D., 2017, MNRAS, 465, 4530
Barro G. et al., 2019, ApJS, 243, 22
Bertin E., Arnouts S., 1996, A&AS, 117, 393
Bonjean V., Aghanim N., Salomé P., Beelen A., Douspis M., Soubrié E., 2019, A&A, 622, A137
Boucaud A. et al., 2020, MNRAS, 491, 2481
Bruzual G., Charlot S., 2003, MNRAS, 344, 1000
Cabrera-Vives G., Reyes I., Förster F., Estévez P. A., Maureira J.-C., 2017, ApJ, 836, 97
Caro F., 2018, PhD thesis, l'Observatoire de Paris
Chabrier G., 2003, PASP, 115, 763
Conselice C. J., 2003, ApJS, 147, 1
Dahlen T. et al., 2013, ApJ, 775, 93
Davidzon I. et al., 2019, MNRAS, 489, 4817
Dimauro P. et al., 2018, MNRAS, 478, 5410
Domínguez Sánchez H., Huertas-Company M., Bernardi M., Tuccillo D., Fischer J. L., 2018, MNRAS, 476, 3661
Dubois Y., Devriendt J., Slyz A., Teyssier R., 2012, MNRAS, 420, 2662
Dubois Y. et al., 2014, MNRAS, 444, 1453
Frery J., Habrard A., Sebban M., Caelen O., He-Guelton L., 2017, in Ceci M., Hollmén J., Todorovski L., Vens C., Džeroski S., eds, Machine Learning and Knowledge Discovery in Databases. Springer International Publishing, New York, p. 20
Fukushima K., 1988, Neural Netw., 1, 119
Fustes D., Manteiga M., Dafonte C., Arcay B., Ulla A., Smith K., Borrachero R., Sordo R., 2013, A&A, 559, A7
Giles D., Walkowicz L., 2018, in American Astronomical Society Meeting Abstracts #231. p. 332.03
Goodfellow I. J., Pouget-Abadie J., Mirza M., Xu B., Warde-Farley D., Ozair S., Courville A., Bengio Y., 2014, in Ghahramani Z., Welling M., Cortes C., Lawrence N. D., Weinberger K. Q., eds, Advances in Neural Information Processing Systems 27. Curran Associates, Inc., New York, p. 2672
Grogin N. A. et al., 2011, ApJs, 197, 35
Gulrajani I., Ahmed F., Arjovsky M., Dumoulin V., Courville A. C., 2017, in Guyon I., Luxburg U. V., Bengio S., Wallach H., Fergus R., Vishwanathan S., Garnett R., eds, Advances in Neural Information Processing Systems 30. Curran Associates, Inc., New York, p. 5767
Haardt F., Madau P., 1996, ApJ, 461, 20
Hassoun M. H., 1995, Fundamentals of Artificial Neural Networks, 1st edn. MIT Press, Cambridge, MA, USA
Huertas-Company M. et al., 2015, ApJS, 221, 8
Huertas-Company M. et al., 2018, ApJ, 858, 114
Jacobs C., Glazebrook K., Collett T., More A., McCarthy C., 2017, MNRAS, 471, 167
Karras T., Aila T., Laine S., Lehtinen J., 2018, Conference Track Proc., 6th International Conference on Learning Representations, (ICLR). OpenReview.net
Kaviraj S. et al., 2017, MNRAS, 467, 4739
Kennicutt R. C., Jr, 1998, ARA&A, 36, 189
Kim E. J., Brunner R. J., 2017, MNRAS, 464, 4463
Koekemoer A. M. et al., 2011, ApJs, 197, 36
Komatsu E. et al., 2011, ApJS, 192, 18
Laigle C. et al., 2019, MNRAS, 486, 5104
Lotz J. M., Primack J., Madau P., 2004, AJ, 128, 163
MacQueen J. B., 1967, in Cam L. M. L., Neyman J., eds, Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability, Vol. 1. Univ. California Press, Berkeley, p. 281
Meusinger H., Schalldach P., Scholz R. D., in der Au A., Newholm M., de Hoon A., Kaminsky B., 2012, A&A, 541, A77
Murphy E. J., Linden S. T., Dong D., Hensley B. S., Momjian E., Helou G., Evans A. S., 2018, ApJ, 862, 20
Nash J. F., 1950, Proc. Natl Acad. Sci., 36, 48
Nayyeri H. et al., 2017, ApJS, 228, 7
Neyshabur B., Bhojanapalli S., Chakrabarti A., 2017, CoRR, abs/1705.07831
Norris R. P., 2017, Publ. Astron. Soc. Austr., 34, e007
Pasquet J., Bertin E., Treyer M., Arnouts S., Fouchez D., 2019, A&A, 621, A26
Protopapas P., Giammarco J. M., Faccioli L., Struble M. F., Dave R., Alcock C., 2006, MNRAS, 369, 677
Radford A., Metz L., Chintala S., 2015, preprint (arXiv:1511.06434)
Ravanbakhsh S., Lanusse F., Mandelbaum R., Schneider J. G., Poczos B., 2017, in Singh S. P., Markovitch S., eds, Proc. Thirty-First (AAAI) Conference on Artificial Intelligence. AAAI Press, p. 1488
Rodriguez-Gomez V. et al., 2015, MNRAS, 449, 49
Rodriguez-Gomez V. et al., 2019, MNRAS, 483, 4140
Salimans T., Goodfellow I., Zaremba W., Cheung V., Radford A., Chen X., Chen X., 2016, in Lee D. D., Sugiyama M., Luxburg U. V., Guyon I., Garnett R., eds, Advances in Neural Information Processing Systems 29. Curran Associates, Inc., New York, p. 2234
Salpeter E. E., 1955, ApJ, 121, 161
Santini P. et al., 2014, A&A, 562, A30
Schlegl T., Seeböck P., Waldstein S. M., Schmidt-Erfurth U., Langs G., 2017, in Niethammer M. et al., eds, Lecture Notes in Computer Science (IPMI). Springer, Cham, p. 146
Solarz A., Bilicki M., Gromadzki M., Pollo A., Durkalec A., Wypych M., 2017, A&A, 606, A39
Sreejith S. et al., 2018, MNRAS, 474, 5232
Stefanon M. et al., 2017, ApJS, 229, 32
Sutherland R. S., Dopita M. A., 1993, ApJS, 88, 253
Teyssier R., 2002, A&A, 385, 337
Thanh-Tung H., Tran T., Venkatesh S., 2019, International Conference on Learning Representations, available at https://openreview.net/forum?id=ByxPYjC5KQ
Tuccillo D., Huertas-Company M., Decencière E., Velasco-Forero S., Domínguez Sánchez H., Dimauro P., 2018, MNRAS, 475, 894
Tweed D., Devriendt J., Blaizot J., Colombi S., Slyz A., 2009, A&A, 506, 647
van der Maaten L., Hinton G., 2008, J. Mach. Learn. Res., 9, 2579
Wong W.-K., Moore A., Cooper G., Wagner M., 2005, J. Mach. Learn. Res., 6, 1961
Zhang J., Vukotic I., Gardner R., 2018, Future Generation Computer Systems, 93, 1

APPENDIX A: TRAINING ALGORITHM

Given a batch of real and generated images, the critic is trained for $n_{\rm critic}$ iterations to approximate the Wasserstein distance, by maximizing the loss in equation (4) whilst keeping the weights of the generator fixed. Afterwards, the generator's weight are updated for a single iteration, by maximizing equation (5), whilst the critic weights are held constant so that it minimizes the approximate Wasserstein distance. This is repeated until the network has converged. Fig. 2 shows a schematic representation the WGAN training procedure.

while $\boldsymbol{\theta}$ *has not converged* **do**
  for $t = 1, ..., n_{critic}$ **do**
    for $i = 1, ..., m$ **do**
      Sample $\mathbf{x} \sim P_r$, real data.;
      Sample $\mathbf{z} \sim P_z$, latent variable.;
      Sample $\epsilon \sim U[0, 1]$, random number to apply gradient penalty.;
      $\tilde{\boldsymbol{x}} \leftarrow G_\theta(\boldsymbol{z})$, generate image from latent variable;
      $\hat{\boldsymbol{x}} \leftarrow \epsilon \boldsymbol{x} + (1 - \epsilon)\tilde{\boldsymbol{x}}$, apply gradient penalty;
      $L^{(i)} \leftarrow C_\psi(\tilde{\boldsymbol{x}}) - C_\psi(\boldsymbol{x}) + \lambda(\|\nabla_{\hat{\boldsymbol{x}}} C_\psi(\hat{\boldsymbol{x}})\|_2 - 1)^2$, calculate loss.
    end
    $\boldsymbol{\psi} \leftarrow Adam(\nabla_\psi \sum_{i=1}^{m} L^{(i)}/m, \psi, \alpha, \beta_1, \beta_2)$;
  end
  for $i = 1, ..., m$ **do**
    Sample a batch of latent variables $\boldsymbol{z}^{(i)} \sim P_z$.;
  end
  $\boldsymbol{\theta} \leftarrow Adam(\nabla_\theta \sum_{i=1}^{m} -C_\psi(G_\theta(\boldsymbol{z}^{(i)}))/m, w, \alpha, \beta_1, \beta_2)$;
end

Algorithm 1: WGAN with gradient penalty training algorithm (Gulrajani et al. 2017). We use default values gradient penalty coefficient of $\lambda = 10$, number of critic iterations per generator iteration $n_{\rm critic} = 10$, batch size $m = 64$ and Adam hyperparameters: $\alpha = 0.00005$, $\beta_1 = 0.5$, $\beta_2 = 0.9$.

This paper has been typeset from a TEX/LATEX file prepared by the author.