Astronomy and Computing 42 (2023) 100685

Contents lists available at ScienceDirect

Astronomy and Computing

journal homepage: www.elsevier.com/locate/ascom

Full length article

Evaluation metrics for galaxy image generators

S. Hackstein [a,*], V. Kinakh [b], C. Bailer [a], M. Melchior [a]

[a] Institute for Data Science, University of Applied Sciences North Western Switzerland (FHNW), 5210 Windisch, Switzerland
[b] Department of Computer Science, University of Geneva, 1211 Geneva, Switzerland

ARTICLE INFO

Article history:
Received 11 November 2022
Accepted 27 December 2022
Available online 5 January 2023

Dataset link: https://www.kaggle.com/c/galaxy-zoo-the-galaxy-challenge

Keywords:
Deep learning
Generative models
Computer-vision
Evaluation
Galaxy morphology

ABSTRACT

A major problem with deep generative models is verifying that the generated distribution resembles the target distribution while the individual generated sample is indistinguishable from the original data. In particular, for application in astrophysics we need to be sure that the generated data matches our prior knowledge and that the generated samples entail all object types with the correct frequency and diversity. We currently lack objective ways to systematically assess these quality aspects, where human inspection reaches its limits, as this requires detailed analysis of a large data volume. In this work, we identify reasonable metrics for the quality of galaxy image generators. To this end, we compare a small set of conditional image generators, trained on galaxy images with classification labels for visual morphology features. Our main contribution is a new set of cluster-based metrics for matching the generated distribution to the target distribution. Furthermore, we use the Wasserstein distance on proxies for galaxy morphology as well as a number of other metrics commonly used for image generators. The newly introduced cluster-based metrics are good proxies for the quality of the generated distribution and are suited for automatized identification of mode collapse. Furthermore, the cluster metrics allow for a qualitative interpretation of the generated distribution. The metrics based on morphological statistics provide a useful tool to probe the physical soundness of generated samples. Finally, we find that kernel inception distance used with an InceptionV3 model pre-trained on ImageNet is a good proxy for the overall quality of galaxy image generators, although it cannot be interpreted that easily.

1. Introduction

Upcoming astronomical surveys with telescopes, such as Euclid (Laureijs et al., 2012) and LSST (Abell et al., 2009), will provide a wealth of data with billions of galaxy images. These hold unprecedented insight in highly researched astrophysical and cosmological questions, such as formation and evolution of galaxies, the cosmic distribution of dark matter, as well as the expansion history of the Universe (Laureijs et al., 2011). However, the huge amount of images is far too big to be investigated by astrophysicists on an individual basis. Instead, fast and systematic extraction of galaxy properties from their images is required in order to inform and constrain physical models. There are a number of computational tools that have been developed and are already in use by the astrophysical community (Rodriguez-Gomez et al., 2019; Boquien et al., 2019; Shamir, 2011). Though these allow for a systematic extraction of properties, they require too much computational resources to provide big collections of high quality mock images in a reasonable amount of time. Hence, it is necessary to replace these tools by faster methods, e. g. by using machine learning techniques, such as deep neural networks (Lovell et al., 2019; Ferreira et al., 2020; Walmsley et al., 2022, e. g.).

It is mandatory to test the accuracy of automated analysis tools and especially machine learning methods on unseen sets of data before applying them to real data. While for example a classifier model trained on supervised data can be tested easily by splitting the training data into training and test sets, testing a full inference pipeline that combines several physical and deep learning models is a much more illusive task. In particular, such pipelines should be tested on different possible scenarios to evaluate whether they are prone to distinguish between competing physical models. This requires the production of collections of synthetic galaxy images. Galaxies can be generated from first principles using semi-analytical models (Somerville and Davé, 2015; Lacey et al., 2016), which allows to control the distribution of generated galaxy types. However, such models have to make simplifying assumptions, e. g. perfect rotational symmetry of the galaxy, and thus cannot generate the full variety of galaxies observed in the Universe. Another approach is to use full-fledged physical numerical simulations, such as cosmological simulations that allow for fine resolution in high density peaks

* Corresponding author.
E-mail address: stefan.hackstein@fhnw.ch (S. Hackstein).

https://doi.org/10.1016/j.ascom.2022.100685

and thus to generate the galaxies associated with the large-scale structure (Pillepich et al., 2018). Such simulations produce the most coherent datasets, as they allow for precise predictions of competing models of cosmology and galaxy formation, as well as realistic propagation effects that affect observations. Unfortunately they are computationally expensive and thus do not enable generating of large datasets for a decent number of competing physical models, which are required for the evaluation of extensive inference pipelines. A promising approach is to use deep generative models, such as variational autoencoders (VAE) (Kingma and Welling, 2014) or generative adversarial networks (GAN) (Goodfellow et al., 2014a), which have the potential to generate galaxy images that are indiscernible from real images. Several models have been proposed in the literature (Regier et al., 2015; Castelvecchi and others, 2017; Ravanbakhsh et al., 2017; Dia et al., 2019; Fussell and Moews, 2019; Smith and Geach, 2019; Lanusse et al., 2021; Bretonnière et al., 2022; Smith et al., 2022; Holzschuh et al., 2022), which have demonstrated the great potential of deep learning techniques. However, we still lack the ability to control the distribution of morphological properties in the generated collection of galaxy images. Still, this is mandatory in order to test whether differences in this distribution can be identified, e. g. using galaxy classification via machine learning. Furthermore, the training of machine learning pipelines often requires the use of large balanced datasets, whereas the distribution of galaxy types in the Universe is extremely unbalanced. We are therefore required to generate balanced datasets. Note that creating new data that matches particular labels is much less expensive than labeling data, thus generating balanced datasets should be preferred over building them by labeling observed data. However, although we can control the labels used with a conditional generative model, there are currently no metrics that are capable to assess in a qualitative way the resemblance of the target distribution by generated data. Still, this is mandatory to verify that the generator indeed produces the desired distribution of data.

Assessing the discrepancy in statistical characteristics between target and generated samples is a major challenge in the evaluation of generative models. In this work, we probe an extensive set of evaluation metrics for galaxy image generators. Some of these metrics are commonly used in machine learning while others are physically motivated and tailored to galaxy images. We identify useful metrics for assessing different aspects of the quality of generative models. This includes per sample image quality, distribution characteristics as well as physical properties. These metrics further allow us to assess specific generated datasets, such as balanced sets or samples that represent competing physical models. In particular, we introduce a new set of cluster-based metrics that assess the distribution of features in a generated set in a qualitative way. These enable the identification of mode collapse and problematic object types in order to guide improvement of generative models. Note that their application is not limited to the context of galaxy images.

For our study, we use RGB images of galaxies from the Sloan Digital Sky Survey (York et al., 2000), provided by Galaxy Zoo data challenge. The label information has been collected in a citizen science approach (Fortson et al., 2012). For conditional training on these labels, we require a fast and reliable tool for automatized classification of visual morphology of generated galaxy images. This will allow to define an additional loss term that will improve the training of the generator. Earlier works investigated the use of different techniques, mostly based on convolutional neural networks (CNNs), (Dieleman et al., 2015; Barchi et al., 2017; Domínguez Sánchez et al., 2018; Primack et al., 2018; Khalifa et al., 2018; Walmsley et al., 2020). For this study, we build on the work by the Galaxy Zoo Challenge Winner Dieleman et al. (2015), to construct a deep neural network for high accuracy classification of visual galaxy morphology features. This morphological image classifier is combined with two conditional deep generative models, a model based on BigGAN (Brock et al., 2018) as well as a simple conditional VAE. We further use the InfoSCC-GAN (Kinakh et al., 2021), where classification of visual galaxy morphology features is instead performed using another classifier together with an encoder. Finally, we use an archetypal mode collapsed generator. We demonstrate that the evaluation metrics presented in this work allow us to identify which of the competing models provides the best generated samples with regard to different aspects of quality.

This paper is organized as follows. In Section 2 we give details on the GalaxyZoo dataset. We describe the evaluation metrics in Section 3 and the model architectures in Section 4. We show our results in Section 5, discuss them in Section 6 and finally conclude in Section 7. The program code, model architectures and training loops can be found in our online repositories.[1,2]

2. Dataset

The Galaxy Challenge[3] by the Galaxy-Zoo citizen science project (Lintott et al., 2008; Fortson et al., 2012; Willett et al., 2013) provides a dataset of more than 60.000 RGB images of galaxies collected by the Sloan Digital Sky Survey (York et al., 2000).

The original images of 424×424 pixels are cropped to the central 207×207 pixels, thus to fit the canvas to the central galaxy of interest while removing neighbors in most cases. Then the images are scaled down to 64×64 pixels. This reduces the memory and time required for training. A sample of the resulting images is shown in Fig. 1. Furthermore, we augment this data during training by random rotation, flipping and translating up to 4 pixels along the x and y axis of the image; these are all variations that naturally arise in astrophysical data. In order to preserve the information from the original images, we apply these operations before cropping. Finally, in order to train the generative models, the pixel color values in the range $[0, 255]$ are rescaled to $[-1, 1]$, which is common practice to enhance performance of machine learning models.

Each galaxy image comes with a vector of 37 label scores for visual morphology features provided by the Galaxy Zoo project. The participants are shown a galaxy image and provide label information by filling a questionnaire (for more information, visit the kaggle webpage[4]). The labels in the Galaxy Zoo challenge reflect the distribution of answers given by participants: 37 float values between 0 and 1. These values are normalized according to the hierarchical structure of questions, indicated in Fig. A.7.

By the very nature of galaxies in the Universe, some types are very common, making up as much as 30% of the whole dataset, while others are extremely rare. Hence, the dataset of the Galaxy Zoo challenge shows a strong imbalance in galaxy types, which is shown in Fig. A.7.

Stratified splitting is not straight forward with the ambiguous labels given in the dataset. However, the evaluation metrics probed in this work formerly compare the overall distribution of data, either regarding physical parameters or feature space, and are thus not very sensitive to the frequencies of rare object types (see Appendix A). For the purpose of this paper, a simple random splitting is sufficient, as we require the separate sets to follow the same underlying distribution. We thus randomly split the full data into training (90% of the full dataset), validation (5%) and test sets (5%). In order to exclude sampling effects when

[1] https://github.com/shackste/galaxy-generator

[2] https://github.com/vkinakh/galaxy-zoo-generation

[3] https://www.kaggle.com/c/galaxy-zoo-the-galaxy-challenge

[4] See footnote 3.

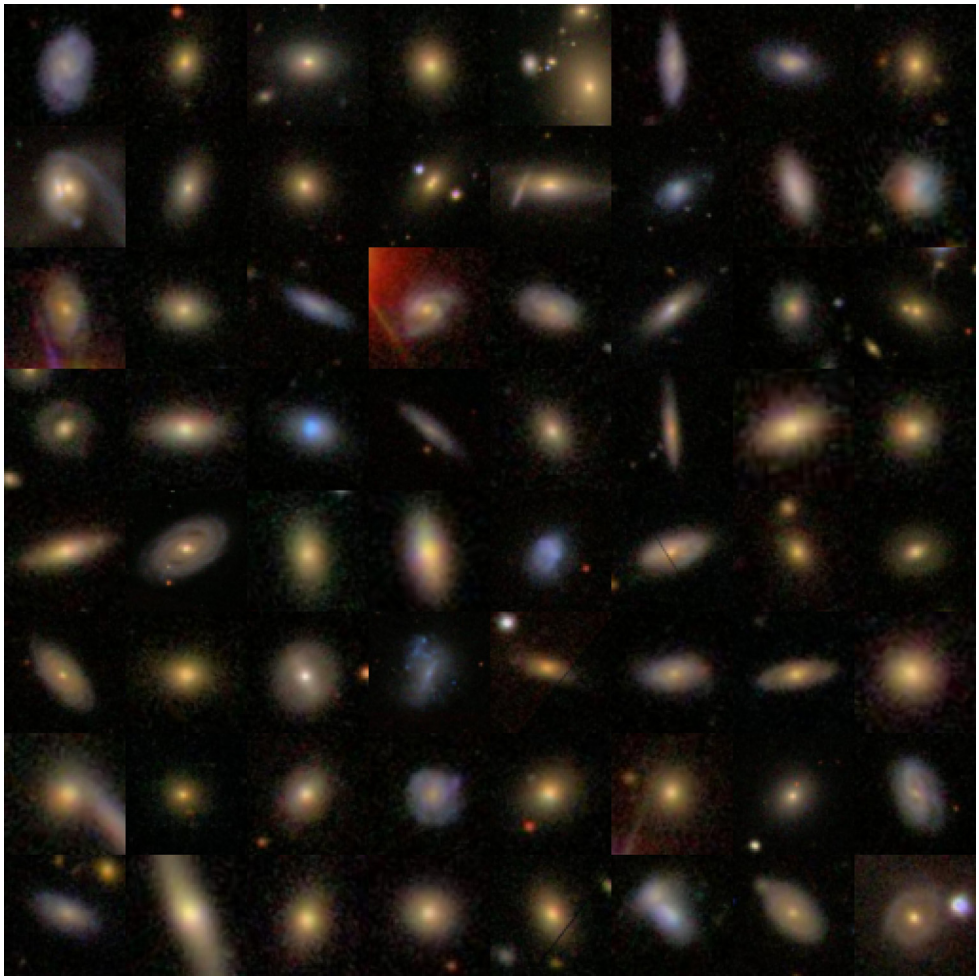

Fig. 1. Exemplary sample of 64 images from the dataset.

comparing the competing generative models, we use the identical split in all our experiments. Note that this ratio of splits was used instead of the more common 60-20-20 as a trade off between numerical stability and computational efficiency, as the required computation time of some metrics increases dramatically with the sample size. However, we argue that this does not harm the assessment of evaluation metrics performed in this work. Still, future studies that assess the quality of model architectures should use larger test and validation sets.

3. Evaluation metrics

A major problem of using generative models in physics is to make sure that the generated images reflect the quality and variety found in real data. In particular, for the case of galaxies, we have to generate physically sound images of all types of galaxies observed in the Universe, including their variation in shape, morphology and image quality. Furthermore, we need methods to verify that a generated set of galaxies truly represents the physical scenario encoded in the input variables. In this work, we probe a number of evaluation metrics that assess the performance of galaxy image generators. In this section, we introduce a new set of cluster-based metrics (Section 3.1) and describe metrics based on morphological proxies (Section 3.2). We further consider a number of metrics commonly used in machine learning: inception score (IS), Fréchet inception distance (FID), kernel inception distance (KID) and perceptual path length (PPL), Chamfer and Wasserstein distance as well as aggregated label control accuracy (ALCA). References and brief explanations in Appendix B.

All metrics considered in this work measure the difference between distributions of two data collections, either in physical parameters or in feature space, see Fig. 2. Thus, metrics which are commonly used to measure the image quality, which quantifies how well individual generated images resemble real data, will also entail information on the distribution quality, i. e. the resemblance of the distribution of data points in the target set. Naturally, two distinct sets picked randomly from the same distribution will not be identical. Thus, for each metric we estimate a reference value to be expected for an ideal generator, which can be obtained by evaluating the sampling error as difference between the equally sized validation and test set. The closer to this reference, the better the generator. For the generative models we obtain metrics by comparing the test set to images generated from validation labels. This way, well generated data has the same distribution of classes as the validation set. Thus, for an ideal generator the metrics deliver values very close to the reference.

Many of the metrics used in this work are computed on a reduced feature representation taken from a neural network, see Fig. 2. We consider several reduction networks, namely

IV3 the InceptionV3 (Szegedy et al., 2015) neural network trained on the ImageNet (Deng et al., 2009) dataset, which is common practice. The feature space has 2048 dimensions.

VGG the features from the last pooling layer of the VGG16 (Simonyan and Zisserman, 2015) neural network, pre-trained

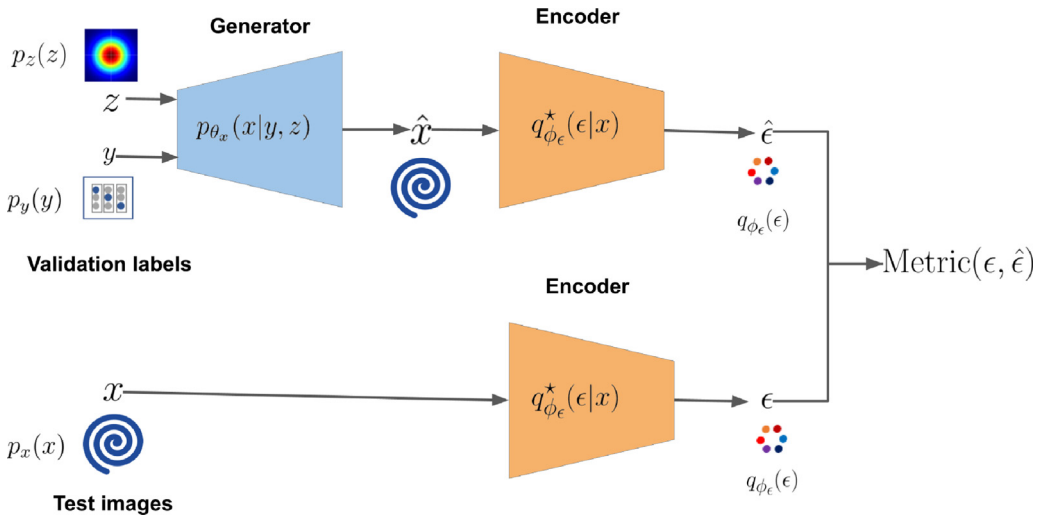

Fig. 2. Schematic overview of the pipeline to compute the evaluation metrics. All images from the test set are reduced to target features using one of the four encoders. Then, images are generated using all labels in the validation set together with random latent vectors. Features are obtained using the same encoder. Finally, the metric is computed comparing these two sets of features.

on the ImageNet dataset. The feature space has 2048 dimensions.

SSL SimCLR neural network (Chen et al., 2020) with ResNet50 (He et al., 2016) base model trained on the galaxy image training set with contrastive loss (Appendix D.3). The feature space has 2048 dimensions.

VAE the encoder of a basic variational autoencoder, trained on reconstruction loss and Kullback–Leibler-loss on the galaxy image training set (Appendix D.4). The latent space has 16 dimensions.

The ImageNet dataset contains a variety of natural images, but no galaxy images. Thus, it is probably not an optimal choice for pre-training in the context of galaxies. We test that hypothesis by comparing IV3 and VGG to an SSL and VAE-based approach.

Note that metrics that consider distances in feature space, such as Wasserstein, Chamfer and the cluster based statistics introduced in the next section, are subject to the curse of dimensionality (Aggarwal et al., 2001). Thus, these distance metrics are computed formerly on the low dimensional feature space of the VAE. In contrast, the metrics commonly used in machine learning (Appendix B) are computed on the much larger feature spaces of IV3 and SSL as well as on the VAE space.

3.1. Cluster based statistics

To compare sample distributions, a feature space can be split into equally sized bins with the sample density compared across the bins. However, this is limited to very low dimensional spaces as the number of required bins grows exponentially with dimensionality.

Kynkäänniemi et al. (2019) propose precision and recall metrics for generative models using k-nearest neighbors in feature space. The distribution of a dataset is modeled as a manifold build by hyperspheres around all feature vectors, which entail the k nearest neighbors. The precision is the amount of generated samples within the reference manifold, while the recall considers reference samples in the generated manifold. These metrics allow to detect out-of-distribution generation (low precision), mode collapse and lack of diversity (low recall). However, they do not test whether object types are produced in the correct frequency and cannot distinguish whether object types lack diversity or are not produced at all, neither show which types are problematic and require more attention. In conclusion, even if these metrics result in perfect scores, this does not imply that the generator is capable to exactly resemble the target distribution. Furthermore, time complexity is quadratic, as pairwise distances have to be computed for all samples in both, the set of reference and the generated set.

As an alternative we propose metrics based on bins obtained via K-means clustering in feature space (Hartigan and Wong, 1979). This method groups the data into k clusters, such that the total distance of data points to the nearest cluster center is minimized. When the cluster center positions have converged, the data points are considered to belong to the cluster represented by the center at lowest distance. These metrics allow for precise assessment of the frequency and diversity of particular object types. Thus, they enable the verification of the exact resemblance of the target distribution as well as to guide the improvement of generative models, all while having linear time complexity.

We perform the clustering in a 16 dimensional feature space, obtained from our VAE model (Appendix D.4). The choice of 16 dimensions is motivated on the one hand to allow for a minimum quality of the resulting images, which is required to ensure that the latent space is meaningful. On the other hand, the number of dimensions should not be too high to prevent the curse of dimensions in distance calculations (Aggarwal et al., 2001). Comparing results of the elbow method (Thorndike, 1953) and gap statistics (Tibshirani et al., 2001), see Appendix E, we find that the dataset is best described by $k = 13$ clusters. For sets of $\approx$3000 samples this choice also ensures numerical stability in the clusters, i. e. > 100 samples per cluster.

First, we fit the k cluster centers to the test set, which is considered to be the target. Second, we estimate the distribution characteristics of the test set. These can be used to renormalize results of other datasets to simplify comparisons. Third, we estimate the characteristics of the validation set, which is considered to be the reference, by using the same cluster centroids. This serves as a reference expected for an ideal generator. The closer to this reference, the better the generator. Finally, models are evaluated by assessing images generated from validation labels, again using the same cluster centroids obtained from the test set.

We propose three metrics:

Cluster error $\mathcal{E}$ We count the number of samples $\hat{n}_c$ in each of K clusters and compute the difference to the target

number n_c. Then, $\mathcal{E}$ is the sum of squared differences over all clusters.

$$\mathcal{E} = \frac{1}{K}\sum_{c=1}^{K}\frac{(\hat{n}_c - n_c)^2}{n_c^2} \tag{1}$$

This metric measures whether the interesting regions in feature space, i. e. the clusters, are populated with the same number of samples as in the target distribution. A value of 0 indicates a perfect match. Larger values indicate deviation from the target, i. e. over- and underproduction of some types. Thus, this metric has the potential to identify mode collapse.

In practice, we renormalize all scores by the value obtained for the reference set. Thus, a score of 1 indicates an ideal result. Significantly higher values indicate mode collapse while lower values may signal overfitting on the target set.

Cluster distance $\mathcal{D}$ We compute the distance $\hat{d}_i$ to the corresponding cluster center for each of N samples. Then, $\mathcal{D}$ is the root-mean-square (RMS) of these distances. The result is renormalized by the RMS of the target set d, thus values closer to 1 are preferred.

$$\mathcal{D} = \frac{1}{d}\sqrt{\frac{1}{N}\sum_{i=1}^{N}\hat{d}_i^2} \tag{2}$$

This metric measures whether the samples populate the correct regions in feature space with sufficient diversity. Values larger than 1 indicate that the sample contains images outside the target distribution. Lower values signal too little sample diversity within the object types.

Cluster standard deviation $\mathcal{S}$ For the distances to the cluster center $\hat{d}_i$, we further compute $\mathcal{S}$ as standard deviation from $\mathcal{D}$. The result is renormalized by the standard deviation $\mathcal{S}_{\text{target}}$ obtained for the target set, thus values closer to 1 are preferred.

$$\mathcal{S} = \frac{1}{\mathcal{S}_{\text{target}}}\sqrt{\frac{1}{N}\sum_{i=1}^{N}(\hat{d}_i - d\cdot\mathcal{D})^2} \tag{3}$$

This is an additional measure for the sample diversity that captures information illusive to $\mathcal{D}$. Values above 1 indicate images outside the target distribution, while values below 1 signal too little sample diversity within the object types.

In combination, these three metrics indicate whether the evaluated distribution fills similar regions of feature space with similar frequency and diversity as the target distribution, thus whether the distributions of two datasets fit each other exactly. Values sufficiently close to the reference indicate that the generative model reproduces the data distribution reasonably well. Moreover, the individual metrics allow for a qualitative assessment of this distribution. It is further possible to compute these metrics for the individual clusters. The results can be used to identify which object types are problematic and what aspects need to be improved, thus guide the improvement of generative models.

3.2. Morphological statistics

The morphological classification of a galaxy from its image is a common problem in astrophysics. To solve this task, several proxies have been developed that can be used to identify the morphological type of the galaxy. While there are many others, in this work we use ellipticity; Gini and M20 coefficients (Gini-m20; Lotz et al., 2004); Concentration, Asymmetry and Smoothness (CAS; Conselice, 2003); as well as Multimode, Intensity and Deviation (MID; Freeman et al., 2013; Peth et al., 2016). We give a brief overview on these proxies in Appendix C. We choose this limited set of proxies, as they can be efficiently computed without the need of any further information on the galaxy, such as redshift, optical size or original resolution. While more proxies should be considered in the future, such as Sérsic profiles or galaxy radius, the presented set of proxies is sufficient for the proof of concept intended in this work. These can be computed using Statmorph,[5] (Rodriguez-Gomez et al., 2019) which is an open-source python package built to extract morphological proxies of galaxies from their images. More detailed information can be found in Rodriguez-Gomez et al. (2019) and the references therein. While the image quality metrics commonly used in machine learning (Appendix B) are computed in feature space, the metrics based on morphological proxies consider the image quality in pixel space, i. e. how well an image resembles a physically sound galaxy.

The input expected by Statmorph is a 2D array of optical brightness, which can easily be obtained from an RGB image by a superposition of the three color channels. Note that these images are stored in compressed JPEG format Wallace (1991), thus do not exactly represent the full precision measurements obtained by the telescope. Though morphological proxies are usually computed on full precision minimally processed imagery, the deep models used in this paper require curated data in consistent format. As the latter criterion is mandatory, we trade-off precision of the morphological proxies for using processed images of consistent format. However, the morphological proxies we use in this work consider properties mostly on large-scales and thus will not be harmed strongly by the reduced precision. Furthermore, the application in this paper requires metrics for the representation we want to be able to generate instead of the metrics for the original galaxy. Thus, it is sufficient to consider reasonable representations of galaxy images that fit the corresponding morphological properties instead of considering full precision imagery of real observations. By construction, this condition is satisfied by the dataset we use here. We further note that both, the original data and generated samples, will have the same systematic error due to this shortcoming, thus allowing for a sound statistical comparison.

We evaluate the distribution of the aforementioned morphological proxies within a set of galaxy images (see also Lanusse et al., 2021, who compared the distribution of CAS visually). In particular, we compare real galaxy images to a generated sample. Properties are computed for all images in both sets using the statmorph python package. Finally, we estimate the Wasserstein distance Appendix B.6 of the resulting point clouds, similar to Holzschuh et al. (2022). The distance is computed for the individual groups of properties, namely for the ellipticity, the Gini-M20, the CAS and the MID, as well as for all of the measures combined.

4. Models

In this work we compare the performance of a small number of conditional image generators. We improve conditional training by using a classifier. To that end, we build a pipeline to classify the rotationally invariant galaxies. It combines features of ResNet18 (He et al., 2016) fed with different rotations of the same image. This model is pre-trained with an meansquared error (MSE) loss (see Appendix D.1). As generators we use:

[5] https://github.com/vrodgom/statmorph

BigGAN an adaptation of the BigGAN model (Brock et al., 2018). It is trained with a hinge loss (Lin et al., 2002) as well as an MSE class loss, using the classifier above (see Appendix D.2).

InfoSCC-GAN (Kinakh et al., 2021) a pipeline that uses a contrastive generator. Conditional generation is trained using a feature classifier in combination with a contrastive encoder. The generator is trained with an adversarial loss and an MSE class loss (see Appendix D.3).

cVAE a basic conditional variational autoencoder. It is trained with an L1 reconstruction loss, a KL loss and an MSE class loss using the same classifier as BigGAN (see Appendix D.4).

collapsed a dummy model that always returns the same image, augmented as explained in Section 2. This model serves as an archetypal reference to identify metrics that can indicate a mode collapse (see Appendix D.5).

For the evaluation of the metrics it is advisable to compare a small set of trained models with obvious advantages and drawbacks. BigGAN and InfoSCC-GAN are sophisticated models expected to generate images of decent quality. In contrast, cVAE is known to generate blurry images and, by construction, the collapsed model delivers high quality images with minimal diversity. Comparing their results will reveal which metrics are good proxies for different aspects of generator quality. All models are trained using the same split of training-validation-test data, as explained in Section 2. A more detailed description of the architectures and training loops of the models are given in Appendix D. In this work, we do not use fully optimized models, neither do we perform a cross-validation, as both would drastically increase the time required for computation without improving the results of this paper, as we focus on identifying metrics suitable to compare competing models that generate galaxy images. For future works that target to compare an extensive list of optimized models, it would be beneficial to compare deep generators to datasets created using physical simulations, such as IllustrisTNG (Pillepich et al., 2018).

5. Results

In order to identify evaluation metrics well suited to assess the quality of galaxy image generators, we assess a list of metrics using a small set of conditional generative models, listed in Section 4. These models are trained to generate galaxy images that resemble those provided by the Galaxy Zoo challenge (Section 2). We compare two sophisticated models, BigGAN and InfoSCC-GAN, and two simple models, cVAE and a collapsed model. The cVAE produces worse resemblance of the target distribution and the worst image quality. The collapsed model repeatedly outputs the same images taken from the training data. It serves as an archetypal example of a mode collapsed generator, that generates high quality images with extremely small diversity. Comparing the more sophisticated models with the cVAE shows which evaluation metrics are good proxies for image quality. Comparing with the collapsed model indicates which metrics provide good proxies for accordance with the target distribution.

We assess the four trained models using the evaluation metrics explained in Section 3. With each model, we generate images using all labels in the validation set ($N = 3079$) together with random latent vectors. The metrics are estimated by comparing to images in the test set ($N = 3079$). For reference, we also compare the images of the validation and test set. Both sets follow the same distribution as they are randomly drawn from the full dataset. By construction, the reference obtained this way for each metric represents the target expected for an ideal generator.

Table 1
Reproduction quality of common visual morphology features by the different generative models. For features that are generated well in quality and diversity, the generated model shows a checkmark. For checkmarks in brackets, the corresponding feature can be identified, but lacks in quality and/or diversity. Features that are not represented at all are marked with a cross.

Feature	BigGAN	InfoSCC-GAN	cVAE	Collapsed
Elliptical	✓	✓	(✓)	✗
Spiral arms	✗	✗	✗	✓
Bar	(✓)	(✓)	✗	✗
Bulge	(✓)	(✓)	✗	✗

5.1. Generated images

As an example, we use 16 labels from the test set and generate images with all four models in order to roughly showcase the difference in generated data quality. These are shown in Fig. 3, together with the original images. The cVAE very roughly reproduces the shapes found in the test images, however with a high level of blurriness. By construction, the collapsed model repeatedly shows high quality images of the same galaxy. The more sophisticated models, BigGAN and InfoSCC-GAN, reproduce the shapes found in the test images. Apart from small details, such as spiral arms or color gradients, the quality of the images is rather good and it is not trivial to distinguish them from real images.

In Table 1, we summarize which prominent visual morphological features are reproduced well by the generators. This data has been obtained by visual inspection of a much larger samples. While large scale visual morphological features, such as ellipses, are reproduced reasonably well by the two sophisticated models, BigGAN and InfoSCC-GAN, all models fail to produce features on smaller scale, such as spiral arms, or lack diversity and/or quality regarding bulges and bar features. While clear spiral arm galaxies are quite abundant in the dataset, there is a lack of clear bulges (see Appendix A). Thus, in the future, the training of models might be enhanced by artificially increasing the abundance of such rare types (see Appendix A). Instead, improving the generation of spiral arm galaxies likely requires to use model architectures that generate superior image quality, such as diffusion models (Ho et al., 2020; Song et al., 2020; Rombach et al., 2022).

Note that the small sample shown here is not sufficient for visual inspection regarding the quality of all features and their distribution in the generated sets. However, no reasonable number of example images will suffice for this task, as the two sophisticated models already defy quality comparison by human inspection, as assessing the quality of features and their distribution in the full dataset requires far too many hours of expert work. Thus, instead of cherry picking more examples, we provide the dedicated reader with an online tool to compare the images generated from given labels by the models presented in this work.[6]

5.2. Clustering

The application of the cluster metrics requires to verify that the clusters are well separated in numbers and that the clusters indeed coincide with certain galaxy types. In Fig. 4 we show the filling of clusters in the target set, i. e. the ground truth, the validation set, i. e. the reference, as well as the different generators. The filling is obtained by predicting the cluster affiliation, given by the cluster center at smallest distance from a given data point, subsequently computing the histogram of affiliations and

[6] https://tinyurl.com/generategalaxies

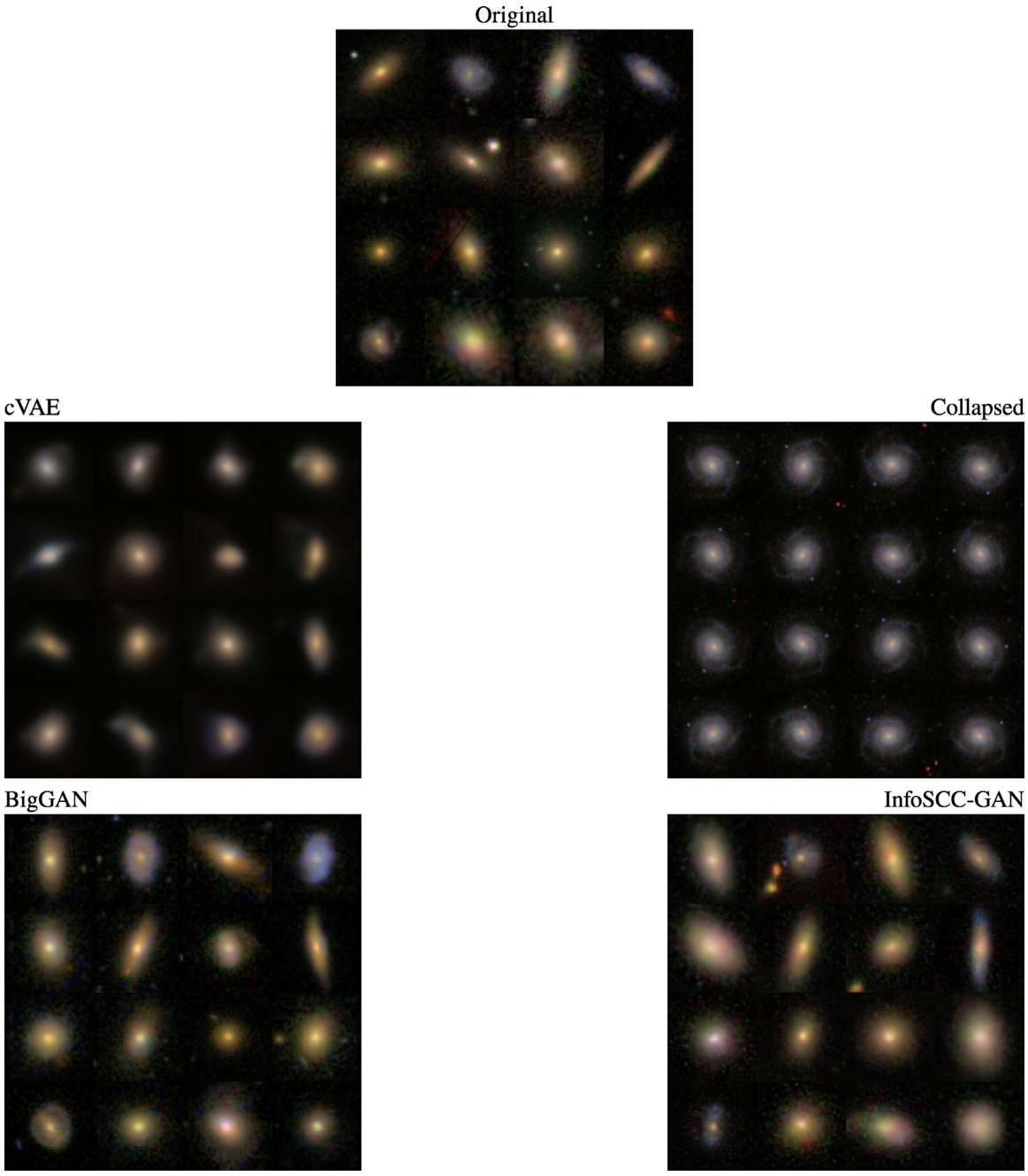

Fig. 3. Samples of real images from the test set (top) and images generated from corresponding labels for all models (Section 4).

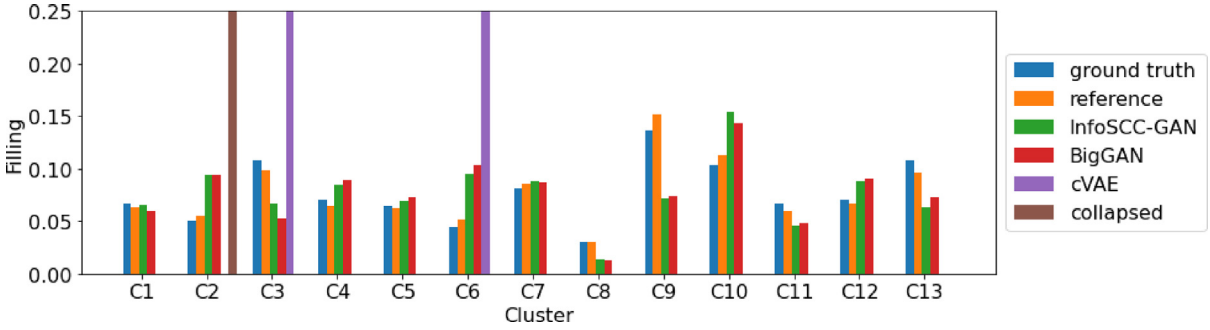

Fig. 4. Percentage filling of clusters by the target set (ground truth), the validation set (reference) as well as sets generated by the various models. The filling is obtained by computing the histogram of cluster affiliation and renormalizing to 1.

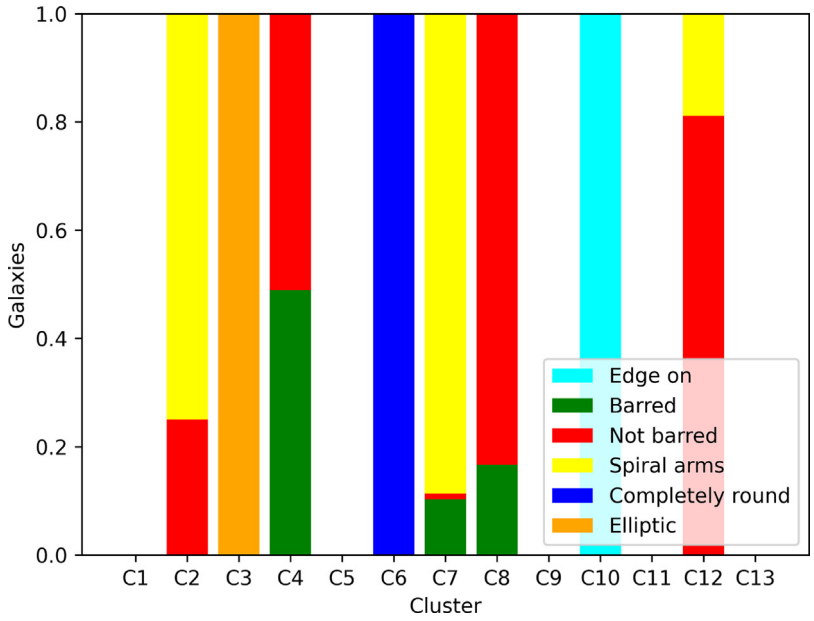

Fig. 5. Cluster association for galaxy images of the most abundant clear types (label score > 0.9). All bars are renormalized to a height of 1.

finally renormalizing to 1. Note that the cluster error metric $\mathcal{E}$ is computed on the non-renormalized histogram, as the generated and real sets are of identical size.

Overall, the fillings of different clusters are sufficiently similar for the real data, considering that the dataset is highly imbalanced (see Appendix A). The sophisticated models, BigGAN and InfoSCC-GAN, both provide a rather good resemblance of the distribution of the real datasets. However, the match is far from perfect – consider for example clusters C2, C9 and C13 – and further improvement is required for the distribution of generated types to match the target. The less sophisticated models, cVAE and the collapsed model, both only occupy 2 or 1 clusters, respectively. This is to be expected, as the collapsed model always outputs the same spiral galaxy, which coincides with cluster C2. Though the cVAE generates more diversity, all the outputs are smooth galaxies, either completely round or elliptical.

In Fig. 5 we present the association of the clusters to galaxy types. We consider features with the most abundant clear galaxy types (i. e. label score > 0.9) that reasonably span the range of different types. We than associate the features of these images with the most nearby cluster and count how many data points of different types are within each cluster. Finally, we renormalize the total number in each cluster to 1 and show the association of clusters with the abundant types.

As can be seen in Fig. 5, Clusters C3, C6 and C10 are well associated with galaxies that clearly appear elliptical (question 7, answer 2, Willett et al., 2013, , Tab. 2), completely round (question 7, answer 1) or edge-on (question 2, answer 1), respectively. Considering galaxies with spiral arms (question 4, answer 1), barred and non-barred galaxies (question 3, answers 1 and 2), we see that these overlap in clusters C2, C7, and C12, while C4 and C8 only contain the latter two types. Since the appearance of spiral galaxies is much more diverse than that of smooth galaxies, it is no surprise that they occupy more clusters. Furthermore, the fact that spiral types partly overlap with barred and non-barred types is expected, as these features are not mutually exclusive. However, the mixing of barred and non-barred galaxies in both clusters, C4 and C8, is likely caused by the limited capacity of the VAE to recreate those features. Moreover, the clusters that lack any clear type data points, C1, C5, C9, C11 and C13, are associated with galaxy images with less clear visual morphology, which is the majority of images in the dataset.

In future works it will be beneficial to use a latent space of higher dimension together with a more sophisticated encoder, in order to enhance information in feature space. This will likely allow to improve the separation of galaxy types into clusters in the sense that there are more clusters and that these can be more clearly identified with certain galaxy types. However, Euclidean distance is prone to the curse of dimensionality and thus looses its meaning in high dimensional space (Aggarwal et al., 2001). Thus, a k-means approach to identify clusters should use a different distance metric that is suited to work in high dimensional space, such as cosine distance or Minkowski distance.

5.3. Morphological statistics

In Fig. 6 we present morphological CAS statistics for several samples. The validation set is almost identical to the target, confirming that they provide a good reference for an ideal generator. The generated sets instead show stronger deviation from the target. We thus expect the obtained metrics for these samples to be significantly higher than the reference, especially for the collapsed model. Unfortunately, the quality of the cVAE images is too low to obtain reasonable morphology proxies, which are far outside the reasonable range and thus cannot be shown. In general, the morphology metrics can be used to tune the generated sample to be balanced or to resemble competing physical models.

5.4. Evaluation metrics

We assess several evaluation metrics (Section 3) for image generators using a small set of trained models (Section 4) with known advantages and drawbacks. Comparing values between these models shows what aspects of generator quality can be measured using these metrics. This can either be the image quality – how realistic is the individual image – or the distribution quality — how well a generated set resembles the target distribution. Results are shown in Table 2. The metrics are estimated by comparing images of the test set, used as target, with images generated using the labels in the validation set. The values expected for an ideal generator are provided for reference in the first column.

To obtain the uncertainty intervals, we compute average and standard deviation of metrics obtained from 30 generated sets.

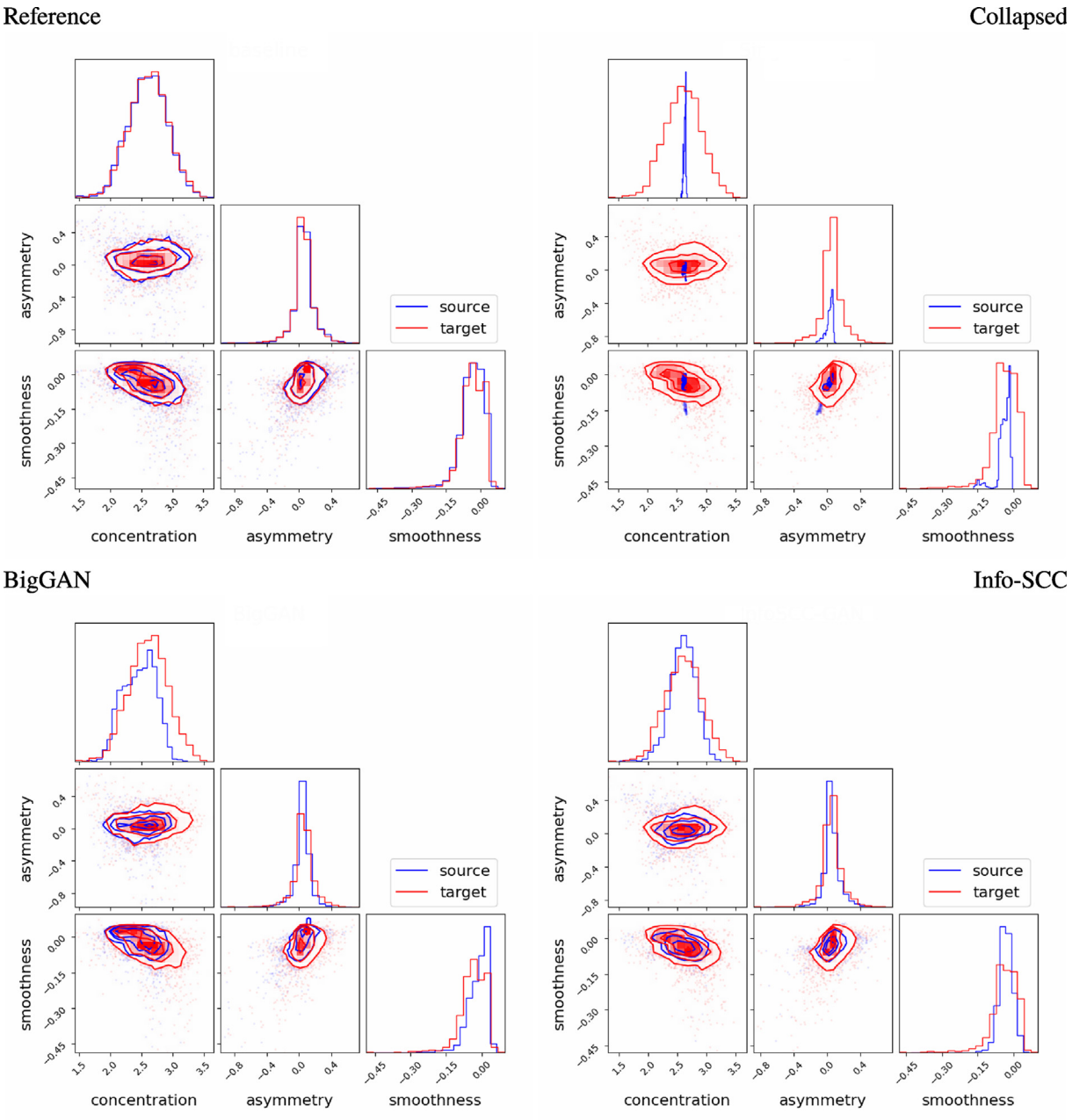

Fig. 6. Corner plots for CAS morphology proxies (Section 3.2). The upper left plot compares the test set to the validation set and serves as a reference for an ideal generator. In the other plots, the test set is compared to images generated by the corresponding model using the labels of the validation set. The difference in the distribution is computed using the Wasserstein distance, results are shown in Table 2. Note that we do not show results for the cVAE, as the bad image quality does not allow to obtain reasonable numbers using the Statmorph code.

These are obtained by using different randomly generated latent vectors while keeping the labels fixed. This way, the uncertainty interval captures the variation of a model regarding the reproduction of the distribution of galaxy types encoded in the labels. The deviation of a metric for a particular model is always well below the difference between models. It is thus sufficient to only present the leading figure with relative deviation. Note that the deviation for the collapsed model by construction is much lower than for the other models, as this model repeatedly returns the same augmented image. For easy comparison, deviations have been rounded up to the same figure as for the other models.

Metrics where the collapsed model is clearly the worst – Wasserstein VAE and $\mathcal{E}$ – are good proxies for the distribution quality of the generated collection. The images of cVAE are blurry and thus less diverse. Hence, cVAE is expected to also have an inferior distribution quality score. The above criterion does not apply to the newly introduced cluster distance $\mathcal{D}$ and cluster standard deviation $\mathcal{S}$ metrics. Still, they are useful for a qualitative interpretation of the distribution of data, and are not necessarily limited to the context of galaxy images. High values of $\mathcal{E}$ correspond to over-/underproduction of some classes. High values of $\mathcal{D}$ and $\mathcal{S}$ show that the generator produces images outside the target distribution. In context of galaxy images, this likely indicates the production of unphysical samples, e.g. elliptical galaxies with spiral arms. However, too low values for $\mathcal{D}$ or $\mathcal{S}$ indicate that images are not sufficiently diverse. In our case, the high results for $\mathcal{E}$ indicate that the collapsed model highly overproduces some galaxy types — it is in mode collapse. This is less severe for the cVAE. In addition, $\mathcal{D}$ below 1 indicates that no unwanted types are generated by these models. However, the rather low values

for $\mathcal{S}$ clearly show that the generated galaxy types are by far not diverse enough.

Metrics for which cVAE is by far the worst are good proxies for image quality. However, all presented metrics that aim to measure image quality evaluate the distribution of features in a dataset found in a deep layer. They thus entail some information on the distribution quality as well. Since the collapsed model results in a similar score as the cVAE for all these metrics, we cannot identify a good proxy solely for image quality. Instead, this aspect has to be inferred by comparing distribution metrics with those that assess both, distribution and image quality.

Metrics where both cVAE and the collapsed model show similar scores, but are clearly worse than the sophisticated models, assess both, image quality and distribution quality, at the same time. This is the case for IS, FID, KID, Chamfer, Wasserstein SSL and ALCA. These metrics are thus good proxies for the overall quality of image generators. However, they do not allow for a qualitative interpretation of the generated results. Interestingly, the Wasserstein distance is much more sensitive to image quality when using features from SSL instead of VAE. This is probably due to the much smaller dimensionality of VAE feature space – 16 instead of 2048 – which lacks a lot of details encoded by SSL.

Finally, by construction, the metrics based on statistics of morphological proxies – CAS, MID, gini-m20, ellipticity and all of them combined – are good metrics for the physical soundness of the individual image as well as the distribution of morphological properties. Thus, these metrics can be understood as image quality metrics in pixel space, in contrast to the image quality metrics discussed before, which are computed in feature space. Hence, these metrics provide useful further information for the evaluation of trained galaxy image generators.

The most promising metric for each aspect of generator quality should put a very strong penalty on the poor performing models. At the same time, there should be a reasonable gap between the reference and the good models, which are still far from being ideal generators. Using these criteria, we identify the most promising metrics for each aspect of generator quality.

Distribution quality: **The cluster error** $\mathcal{E}$ puts the highest penalty on the collapsed model. When used together with the **cluster distance** $\mathcal{D}$ and **cluster standard deviation** $\mathcal{S}$ metrics, they provide great tools for a qualitative assessment of the generated distribution. However, the Wasserstein distance for cVAE is closer to the sophisticated models and thus has a stronger focus on the distribution quality than the cluster-based metrics. Still, the Wasserstein distance does not provide an easy way to assess what aspects of the distribution have to be improved.

Mode collapse: The very high penalty that **cluster error** $\mathcal{E}$ puts on the collapsed model suggests it to be a great metric to identify mode collapse.

Distribution and image quality: The KID SSL and both Chamfer metrics put the highest penalty on cVAE and the collapsed model. Compared to that, the KID metrics – IV3 and VAE more than SSL – leave more room for required further improvement of the sophisticated models while strongly penalizing the poor performing models. The best balance of both aspects is provided by **KID IV3**. Besides that, **ALCA** provides a mandatory sanity check that images indeed show the requested types of galaxies.

Physical soundness: Ellipticity provides the strongest penalty gradient. However, it does not entail as much physical information as the other **morphological proxies**. These should thus be taken into consideration as well, even though they partly contain the same information. Moreover, corner plots, as shown in Fig. 6, provide a great tool for human inspection of the physical soundness. However, using all morphological proxies in a single metric does not provide an equally well suited gradient of penalty as the separated versions. As the combined metric does not provide any improvement over the separated ones, it can be dropped in future works in favor of the metrics that consider the separated sets of proxies.

The reference value of ALCA further provides an estimate on the quality of the galaxy image classifier. It measures the root-mean-square deviation of labels obtained for the validation images from their original labels. Our score of 0.02 is significantly better than that of Dieleman et al. (2015) (≈ 0.08). Furthermore, since we are using ALReLU as activation functions, rare types have a significantly higher chance of correct classification. This results in over 90% classification accuracy in all questions answered by more than 50% of participants (see Appendix D.1.3). For comparison, Dieleman et al. (2015) obtain average accuracy above 90 % for common types, but only above 70% for rare types (Appendix A).

Furthermore, we assess the entanglement of the latent space of generators by computing PPL on a set of 50,000 images. We find that using features from VGG, the PPL values are of order 10^{-7} for all models, indicating converged and stable generators. PPL thus provides a sanity check for the stability of a generator, but cannot be used to compare the quality of competing generators. Using features from SSL or VAE we find values ranging from 10^{-3} to 10^{2}. This shows that training of the dimension reduction models on the Galaxy Zoo dataset does not improve results, but the use of pre-trained models should be favored. In contrast, the FID metric improved significantly from using feature reduction models trained on the galaxy dataset. For the KID metric, the case is less clear.

Note that for better comparison to the other metrics, the FID scores presented in Table 2 have been computed on the same $\approx$ 3000 test images. This is significantly less than $>$10,000 samples usually required for stable results. However, we also have computed FID scores on the same 50,000 images as PPL and find less than 1% deviation from the results reported in Table 2.

Comparing the results for the two sophisticated models, we see that InfoSCC-GAN consistently provides slightly better scores than BigGAN in the metrics that measure distribution quality. In contrast, BigGAN consistently provides slightly better scores in metrics that measure both, distribution and image quality. We thus conclude that InfoSCC-GAN provides a better representation of the target distribution, while BigGAN generates images of higher quality.

6. Discussion

Our work focuses on identifying evaluation metrics which are best suited to assess the quality of galaxy image generators. To this end, we probe a set of evaluation metrics for galaxy image generators that assess different aspects of the generators quality. Standing out are the newly introduced cluster-based metrics. They provide a qualitative way of assessing the generated distribution regarding the following questions

1. Are all object types found in the training data generated in the right frequency or is the model in mode collapse? (cluster error $\mathcal{E}$)
2. Does the generator produce out-of-distribution objects? (cluster distance $\mathcal{D}$ and cluster standard deviation $\mathcal{S}$)
3. Does the generator capture the diversity within object types? ($\mathcal{D}$ and $\mathcal{S}$)

Table 2
Evaluation metrics (Section 3) to assess the quality of galaxy image generator models (Section 4). Values in the reference column are obtained by comparing images in the validation set ($N = 3079$) to images in the test set ($N = 3079$) and represent the value expected for an ideal generator. The values in the other columns are obtained by comparing the test set to images generated by the given model using the labels of the validation set together with random latent vectors. Results are averaged over 30 sets generated from different latent vectors, the relative standard deviation is given in the brackets. For easy comparison, the very low deviation of the collapsed model has been rounded up to the same significant figure as for the other models. We highlight metrics that are good proxies for distribution quality, both, image quality and distribution quality or physical soundness.

Metric	Reference	BigGAN	InfoSCC-GAN	cVAE	Collapsed
IS	2.2	**2.0** (±0.6%)	1.8 (±0.7%)	1.3 (±0.6%)	1.2 (±0.1%)
FID IV3	$5 \cdot 10^{0}$	$\mathbf{4 \cdot 10^{1}}$ (±0.6%)	$7 \cdot 10^{1}$ (±0.7%)	$2 \cdot 10^{2}$ (±0.6%)	$2 \cdot 10^{2}$ (±0.1%)
FID SSL	$2 \cdot 10^{-2}$	$\mathbf{4 \cdot 10^{-1}}$ (±1.1%)	$9 \cdot 10^{-1}$ (±1.3%)	$3 \cdot 10^{0}$ (±1.0%)	$4 \cdot 10^{0}$ (±0.1%)
FID VAE	$1 \cdot 10^{0}$	$\mathbf{4 \cdot 10^{1}}$ (±1.3%)	$5 \cdot 10^{1}$ (±1.5%)	$1 \cdot 10^{2}$ (±1.2%)	$3 \cdot 10^{2}$ (±0.1%)
KID IV3	$2 \cdot 10^{-5}$	$\mathbf{3 \cdot 10^{-2}}$ (±1.3%)	$4 \cdot 10^{-2}$ (±1.4%)	$2 \cdot 10^{-1}$ (±1.3%)	$3 \cdot 10^{-1}$ (±0.1%)
KID SSL	$1 \cdot 10^{-6}$	$\mathbf{8 \cdot 10^{-5}}$ (±1.9%)	$1 \cdot 10^{-4}$ (±2.1%)	$2 \cdot 10^{-3}$ (±2.0%)	$2 \cdot 10^{-3}$ (±0.1%)
KID VAE	$2 \cdot 10^{-1}$	$9 \cdot 10^{2}$ (±1.9%)	$\mathbf{7 \cdot 10^{2}}$ (±2.0%)	$1 \cdot 10^{3}$ (±2.0%)	$3 \cdot 10^{3}$ (±0.1%)
Chamfer SSL	$1 \cdot 10^{3}$	$\mathbf{4 \cdot 10^{3}}$ (±11.4%)	$5 \cdot 10^{3}$ (±11.6%)	$5 \cdot 10^{5}$ (±11.2%)	$6 \cdot 10^{5}$ (±0.1%)
Chamfer VAE	$4 \cdot 10^{3}$	$\mathbf{2 \cdot 10^{4}}$ (±9.4%)	$3 \cdot 10^{4}$ (±9.3%)	$9 \cdot 10^{5}$ (±8.9%)	$7 \cdot 10^{5}$ (±0.1%)
ALCA	$2 \cdot 10^{-2}$	$\mathbf{8 \cdot 10^{-2}}$ (±1.0%)	$1 \cdot 10^{-1}$ (±1.0%)	$6 \cdot 10^{-1}$ (±0.9%)	$7 \cdot 10^{-1}$ (±0.1%)
Wasserstein SSL	$2 \cdot 10^{-1}$	$\mathbf{4 \cdot 10^{-1}}$ (±0.7%)	$7 \cdot 10^{-1}$ (±0.8%)	$2 \cdot 10^{0}$ (±0.6%)	$2 \cdot 10^{0}$ (±0.1%)
Wasserstein VAE	$6.8 \cdot 10^{1}$	$9.6 \cdot 10^{1}$ (±0.4%)	$\mathbf{8.4 \cdot 10^{1}}$ (±0.8%)	$1.2 \cdot 10^{2}$ (±0.4%)	$1.9 \cdot 10^{2}$ (±0.1%)
$\mathcal{E}$	$1 \cdot 10^{0}$	$2 \cdot 10^{0}$ (±3.5%)	$\mathbf{1 \cdot 10^{0}}$ (±10.8%)	$1 \cdot 10^{1}$ (±2.0%)	$8 \cdot 10^{1}$ (±0.1%)
$\mathcal{D}$	1.001	1.094 (±0.3%)	**1.073** (±0.6%)	0.873 (±0.3%)	0.911 (±0.1%)
$\mathcal{S}$	$1 \cdot 10^{0}$	$7 \cdot 10^{-1}$ (±1.4%)	$\mathbf{9 \cdot 10^{-1}}$ (±2.0%)	$4 \cdot 10^{-1}$ (±1.4%)	$3 \cdot 10^{-1}$ (±0.1%)
CAS	$4 \cdot 10^{-4}$	$1 \cdot 10^{-2}$ (±4.2%)	$\mathbf{6 \cdot 10^{-3}}$ (±4.6%)	–	$5 \cdot 10^{-2}$ (±0.1%)
MID	$3 \cdot 10^{-4}$	$5 \cdot 10^{-3}$ (±9.9%)	$\mathbf{2 \cdot 10^{-3}}$ (±10.5%)	–	$3 \cdot 10^{-2}$ (±0.1%)
gini-m20	$2 \cdot 10^{-5}$	$5 \cdot 10^{-3}$ (±9.1%)	$\mathbf{4 \cdot 10^{-3}}$ (±9.4%)	–	$5 \cdot 10^{-2}$ (±0.1%)
ellipticity	$4 \cdot 10^{-7}$	$8 \cdot 10^{-4}$ (±7.6%)	$\mathbf{5 \cdot 10^{-4}}$ (±8.9%)	–	$4 \cdot 10^{-2}$ (±0.1%)
all morphs	$7 \cdot 10^{-3}$	$4 \cdot 10^{-2}$ (±2.4%)	$\mathbf{2 \cdot 10^{-2}}$ (±2.9%)	–	$2 \cdot 10^{-1}$ (±0.1%)

While we present the $\mathcal{D}$ and $\mathcal{S}$ metrics averaged over all clusters, results can be obtained for the individual clusters to show which object types need closer attention. In a future study, we will investigate this possibility as well as the application of these metrics to other datasets. Note that the number of clusters $k = 13$ used in this work is insufficient to verify that all rare types are produced in the right frequencies. A larger test set would be required to fully capture the diversity of the galaxy image dataset. Furthermore, the feature representation would need to be more informative. However, we know that none of the presented models generates all types well, e.g. spiral arm galaxies. Thus, at this point it is sufficient to probe only 10 clusters, which span the variety of well produced galaxy types in the sophisticated models. As the less sophisticated models lack this variety, the presented results well display the potential of the cluster based metrics. We confirmed the results by finding the same trends for different choices of $k < 20$. Still, in future studies, a suitable choice for k to test for the full variety in a larger target test set should be obtained via the ELBO-method (Thorndike, 1953).

Our dataset contains RGB images with labels for visual morphology features obtained via a citizen science project. To our knowledge, the Galaxy Zoo datasets are the only datasets available with labels regarding several aspects of visual galaxy morphology, such as ellipticity, spiral arms, bars and bulges. As the images are anonymized from the original SDSS data, it is not possible to trace back the original survey data. We thus have to use RGB images instead of raw survey data. However, from the computational point of view, this is an advantage, as required computational resources can be greatly reduced by the use of pre-processed data. Furthermore, the RGB images entail all information required for morphological classification. As the presented evaluation metrics – except for the morphological proxies – do not depend on the actual format of the dataset, no further advantage for this work is expected from using raw survey data instead of RGB images. Nonetheless, this work provides a proof of concept and all presented methods and findings can be applied to raw survey data as well.

For the training of all neural networks we use identical splitting into training, validation and test sets in order to exclude differences in sampling. This way, we can isolate and focus on the change in results due to different model architectures. Though a cross-validation approach would reduce bias induced by the splitting, it would also increase the required computation time. Still, the approach we choose allows for a systematic comparison of competing models. Their quality can be compared by the evaluation metrics presented in this paper.

This work focuses on investigation of several evaluation metrics for their ability to assess different quality aspects of galaxy image generators. It is advisable to compare models with obvious advantages and drawbacks in order to evaluate these metrics. We thus do not use fully optimized models in this work. Instead, optimizing the presented models will be part of an extensive ablation study we are currently working on.

A major drawback of CNNs is that, by construction, such networks focus on local features of the image instead of global ones. For example, local arc-shaped structures of high brightness can be generated well. However, generating a spiral instead of a ring is not a trivial task. We find that GANs based on CNNs tend to produce rings of high brightness in galaxies that should have spiral arms. We will further address this problem in a future study.

Morphological proxies from galaxy images have been computed using STATMORPH (see Section 3.2). This python package requires the subtraction of background noise and point spread function in pre-processing. This is usually done by using the entire image taken by the telescope (Blanton et al., 2011), where information – background signal or position within telescope beam – is available for a reasonable estimate. However, the generative models are built to produce images of individual galaxies without enough isolated background signal. Furthermore, for the training images, the position of the galaxy in the aperture of the telescope is unknown. This prevents us from subtracting the point spread function. In some cases, this may result in erroneous estimates of the morphological proxies. Ideally, galaxy image generators should use background subtracted images. However, original as well as generated images are equally affected by this caveat. We thus argue that this does not affect the validity of

a statistical comparison of the distribution of the morphological proxies.

We choose to compute the distribution scores for morphological parameters (Section 3.2) using the Wasserstein distance (Appendix B.6). We also tried to calculate the distance between distributions using the Chamfer distance (Appendix B.5). Both metrics consider the distance of nearest neighbors from two point clouds. However, while in the Chamfer distance two pairs may contain the same point, this is not the case for the Wasserstein distance. If the source distribution is concentrated on regions which are densely populated by the set of reference, the expected distance to the nearest neighbor is smaller than in more sparsely populated regions. Thus, Chamfer distance for the concentrated set is lower than for a point cloud that resembles the entire target distribution. Instead, the Wasserstein distance penalizes overcrowding if both clouds contain the same number of points. Though the Chamfer distance is computationally more efficient, the Wasserstein distance is a better metric for the target of recreating the reference distribution.

The relative standard deviation is significantly higher for Chamfer distance and the individual morphological proxy groups ($\sim$ 10%) than for the other metrics ($\sim$ 1%). Note that the Chamfer distance is computed on the same set of features as e. g. the Wasserstein distance, thus measuring the same underlying variation. However, since Chamfer distance is computed on 3D datapoints, the dimensionality of the feature space is further reduced using a T-SNE approach. The resulting lower dimensional representation puts a focus on the variation of the underlying data. This naturally results in a higher weight on the variation between different datasets, thus in higher deviation for the metric. The same is true for the morphology metrics, which use different approaches of reducing the features to three or less dimensions while focusing on the variation in the data.

7. Conclusions

In order to find suitable evaluation metrics to assess the quality of galaxy image generators, we investigate a number of evaluation metrics with the potential to measure different quality aspects. These metrics are probed with a small number of conditional generative models, some of which purposely are of worse quality than the others. We identified evaluation metrics that are good proxies for the quality of individual images and the resemblance of the target distribution. Our main results are:

- The newly introduced cluster-based metrics (Section 3.1) are a formidable new tool to assess the distribution of generated data.
- The KID metric on features of the pre-trained InceptionV3 model (Szegedy et al., 2015) provides a useful proxy on the overall quality of galaxy image generators.
- Using an ALReLU activation (Mastromichalakis, 2020) significantly enhances classification accuracy for rare object types. This is required to train conditional generators on the highly imbalanced dataset of galaxy images.

We introduce a new set of cluster-based metrics (Section 3.1), which are good proxies for the distribution quality. In particular, they allow for a qualitative interpretation regarding the sample diversity and the amount of samples generated for different types. Thus, they provide an intuitive new tool to assess the resemblance of the target distribution in a qualitative way, which so far has been a rather illusive task. In addition, one of the cluster metrics, the cluster error $\mathcal{E}$, provides a formidable tool to identify mode collapse. Moreover, the cluster metrics have the potential to be transformed into a loss function to train generative models to reproduce the distribution of training data exactly, or any other target distribution, e. g. balanced datasets. Together, the aforementioned metrics provide a good basis to assess the quality of trained image generators, which is not necessarily limited to the context of galaxy images. We are currently working on providing losses based on these cluster metrics, which have the potential to drastically improve training of generative models regarding distribution quality, which is especially valuable for GANs.

We find that the evaluated metrics commonly used to assess the quality of generated images are all sensitive to the resemblance of the target distribution as well, as they all basically compare the distribution of datasets in feature space. Of these metrics, the KID stands out in that it puts high penalty on the worse generators but at the same time leaves enough room for further improvement of the more sophisticated models, which are still far from ideal.

In addition, we also probed metrics tailored to galaxy images that are based on morphological statistics widely used in astrophysics. These can be used to assess the physical soundness of the generated sample. Furthermore, they are required to tune the generated sample to specific distributions, e. g. competing physical models or balanced datasets.

We identify useful evaluation metrics that assess different quality aspects of galaxy image generators. Thus, the results of our work allow for ablation studies to find which architecture is best suited to generate galaxy images for scientific purposes. This is a prerequisite task for testing inference methods as well as training of deep learning models. Both are required to learn from coming humongous datasets in a reasonable amount of time. We are in the process of preparing such an ablation study.

CRediT authorship contribution statement

S. Hackstein: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data curation, Writing – original draft, Writing – review & editing, Visualization, Supervision, Project administration. **V. Kinakh:** Software, Validation, Formal analysis, Investigation, Data curation, Writing – original draft, Writing – review & editing, Visualization. **C. Bailer:** Conceptualization, Methodology, Software, Investigation. **M. Melchior:** Conceptualization, Methodology, Supervision, Project administration, Funding acquisition.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: All authors report that financial support was provided by the Swiss National Science Foundation (SNF).

Data availability

The data underlying this article are available in the article and in its online supplementary material. The dataset is available in https://www.kaggle.com/c/galaxy-zoo-the-galaxy-challenge. The models and training loops can be found in https://github.com/shackste/galaxy-generator and https://github.com/vkinakh/galaxy-zoo-generation. Galaxy images can be generated interactively in https://tinyurl.com/generategalaxies.

Acknowledgments

This work has been funded by the SNF via the Euclid Sinergia project under project numbers CRSII5_173716 and CRSII5_198674 as well as the RODEM sinergia project under project number CRSII5_193716/1. Computations have been performed on CSCS (project ID sm63), HSS and HPC at University of Geneva using deepo dockers (Yang, 2017) and conda environment.

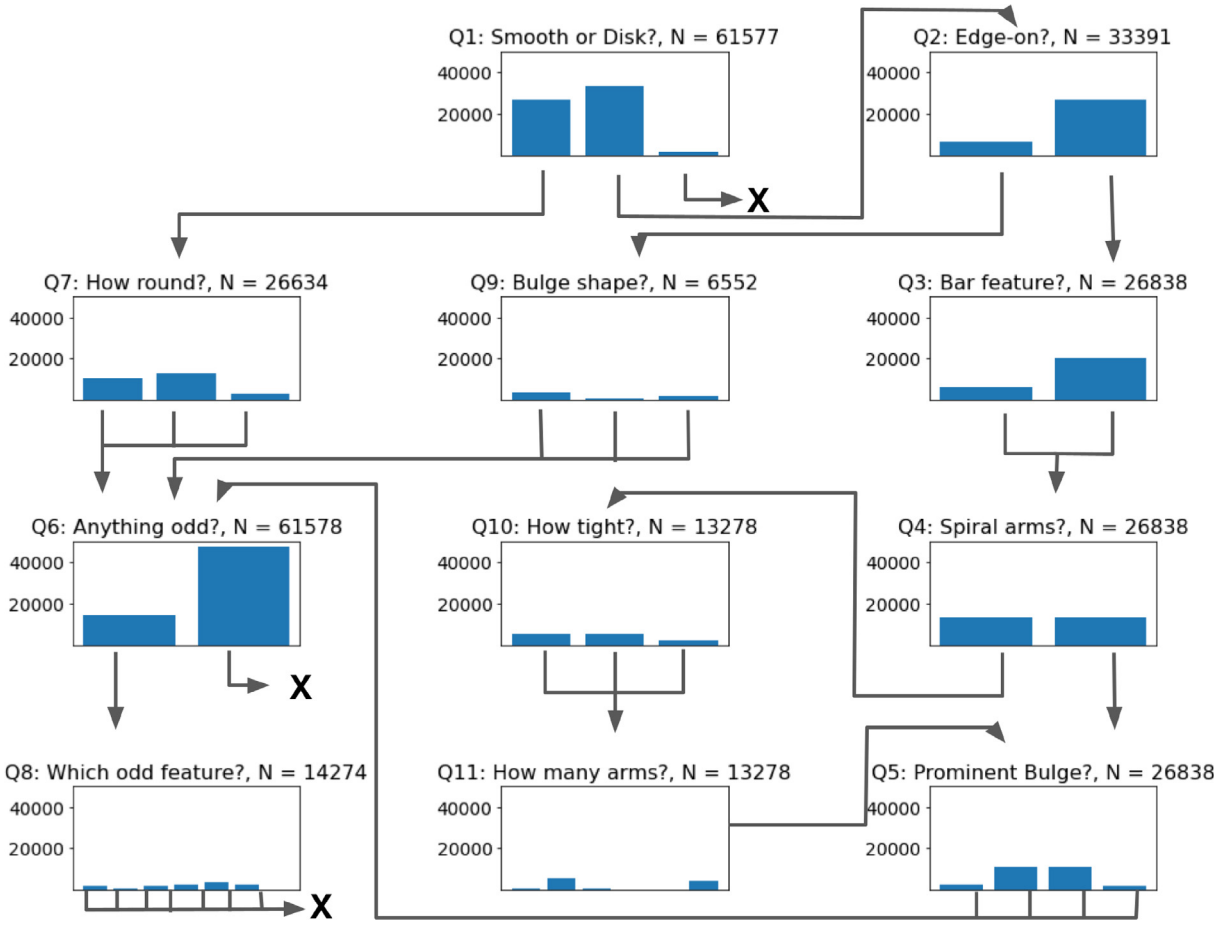

Fig. A.7. Sketch of the imbalance in the hierarchical label space. Each of the plots shows the distribution of answers to a given question. The title indicates the question and the total number of participants that were asked this question. The arrows indicate which questions follow after the corresponding answer. For more details on questions and answers, see https://www.kaggle.com/c/galaxy-zoo-the-galaxy-challenge/overview/the-galaxy-zoo-decision-tree.

Appendix A. Galaxy Zoo — class imbalance

In Fig. A.7 we show a schematic overview of the hierarchical structure of labels. Participants are shown a picture and start with question Q1. Depending on their answer, they proceed to one of the next questions, Q2 or Q7, or end the questionnaire (Willett et al., 2013). The most abundant galaxy types with clear labels (i. e. label score > 0.9) are spiral galaxies, predominantly with two arms, smooth galaxies, especially completely round ones, edge-on galaxies as well as barred galaxies. In contrast, there are no samples with clear labels regarding bulges, cigar-shapes or odd features, except for rings.

Appendix B. Common evaluation metrics

In the following, we give a short overview on metrics commonly used to assess the image quality of data generated by deep neural networks. While they all assess the distribution of feature vectors of generated sets, they have been shown to be good metrics for the quality of images in the sense that for good scores, the generated images are indiscernible from real images.

B.1. Inception score

The Inception Score (IS) (Salimans et al., 2016) evaluates the quality of generated images by measuring the variety and unambiguity of their labels, predicted by a suitably trained classifier. We use the IV3 model, which is common practice. Higher values for IS indicate better image quality.

B.2. FID score

The Fréchet inception distance (FID) (Heusel et al., 2017) is a metric used to assess the quality of generated images as well as their distribution. Using the features in a deep layer of a recognition model, it compares the distribution of a set of generated images with the distribution of a set of real images. A lower value indicates a better match, which correlates with better image quality and resemblance of the target distribution. We use the FID score with features from the IV3 (FID IV3), SSL (FID SSL) and VAE (FID VAE) models.

B.3. KID score

Kernel Inception Distance (KID) (Bińkowski et al., 2018) is another metric for the evaluation of image quality that extends FID. It uses a polynomial kernel to eliminate the potential bias of FID. We use the KID score with features from the IV3 (KID IV3), SSL (KID SSL) or VAE (KID VAE) models.

B.4. Perceptual path length

The perceptual path length (PPL) (Karras et al., 2021) measures the entanglement of features in the latent space of a generator. It is computed as the empirical mean of the perceptual difference between images generated from similar latent vectors. In practice, two images are generated using the same labels but slightly different latent vectors. These images are encoded using one of

the reduction models listed above. The PPL is the L2 distance of the encoded images. Averaging PPL over many pairs of images allows us to measure the smoothness of latent space. Small values indicate that generated images change smoothly with changing latent input, suggesting a stable generator. We use PPL with features from VGG, SSL or VAE.

B.5. Chamfer distance

Chamfer distance (Ravi et al., 2020) measures the similarity of two point clouds as the average distance of nearest neighbors. It can be used to assess the image quality by computing the distance between the projected features of the real and generated images. We use the Chamfer distance with features from SSL (Chamfer SSL) or VAE (Chamfer VAE).

B.6. Wasserstein distance

The unweighted Wasserstein distance (Villani, 2008) (also called optimal transport) measures the similarity of two point clouds as the smallest possible sum of distances for a bijective connection of points. In contrast, the Chamfer distance allows us to compute the distance of multiple points to the same nearest neighbor. The simplest computation creates a distance matrix between all pairs of points and sums up the smallest numbers, using only one number in each row and each column. As it is a computationally expensive problem, it needs to be approximated. We use the python package Geomloss (Feydy et al., 2019), which approximates the Wasserstein distance with an unbiased Sinkhorn divergence. The distance is computed on features obtained from SSL (Wasserstein SSL) or VAE (Wasserstein VAE).

B.7. Aggregated label control accuracy

The aggregated label control accuracy (ALCA) measures how well images produced by a conditional generator resemble the input labels. Generated images are processed by the classifier (Appendix D.1). The ALCA is the root-mean-square (RMS) deviation between input labels and predictions.

Appendix C. Morphological proxies

Here we give a short overview on the morphological proxies used for physically motivated metrics for the quality of generators. All these proxies are computed directly on the image. As the physical soundness of the generated images can be regarded as image quality, the physically motivated metrics are an efficient way to assess the image quality in pixel space.

C.1. Ellipticity

The *ellipticity* measures the roundness of a galaxy. It is proportional to the ratio of semi-minor-axis to semi-major-axis. These are the shortest and longest straight lines, respectively, through the center of the observed ellipse. Higher values indicate a more elliptical shape.

C.2. Gini-m20

The Gini-M20 classification system (Lotz et al., 2004) has been used to distinguish early-type, late-type and merging galaxies.

The *Gini* coefficient (G) is a traditional statistic that measures distribution inequality. When used for galaxy images, it is a measure for the homogeneity of the brightness distribution within the galaxy, where $G = 1$ for all of the flux concentrated in a single pixel and $G = 0$ for homogeneous brightness.

The *M20* light statistic (M_{20}) measures the distribution of brightness across the galaxy. Higher values indicate that bright pixels are farther from the center.

C.3. CAS

The non-parametric CAS statistics (Conselice, 2003) are a widely used set of morphology indicators. They reflect the merger history, mass and environment of the galaxy.

The *concentration* index (C) measures how strongly the brightness is concentrated on the center of the galaxy. A higher value of C indicates a more extended bright center.

The *asymmetry* index (A) is obtained by subtracting the galaxy image rotated by 180° from the original image. Higher values of A indicate that the galaxy is less symmetric.

The *smoothness* or "clumpiness" index (S) measures how smoothly the light is distributed among the galaxy. It is obtained by subtracting the galaxy image smoothed with a boxcar filter from the original image. Larger values of S correspond to galaxies that are less smooth, or more "clumpy".

C.4. MID

The MID statistics (Freeman et al., 2013; Peth et al., 2016) are an alternative to the aforementioned statistics, intended to be more sensitive to recent galaxy mergers. The MID is calculated on a segmentation map, which is obtained by defining a surface brightness threshold. The main source is the set of connected pixels containing the brightest pixel. This segmentation map is further regularized with a 3×3 boxcar filter.

The *multimode* statistic (M) is intended to search for a second prominent clump, which could indicate a recent merger. It is computed as the ratio between the areas covered by the two most prominent clumps within a galaxy. High values of M indicate the presence of a second prominent clump.

The *intensity* statistic (I) is intended to find a second subregion with brightness comparable to the center of a galaxy. It measures the ratio of total brightness between the two brightest subregions of a galaxy. High values of I indicate a second subregion with brightness comparable to the center. This could be the remnant of a recent merger.

The *deviation* statistic (D) measures the distance between the centroid and the brightest peak found during computation of I. High values of D indicate a strong separation of the two brightest regions. This could signal a recent merger.

Appendix D. Model architectures and training pipelines

D.1. Classifier

D.1.1. Architecture

Classification of visual morphology of galaxies from RGB images is not straightforward. The simple application of state-of-the-art image classifier neural networks, such as ResNet (He et al., 2016), does not deliver satisfactory results, even when using data augmentation. This is because CNNs are not equivariant to rotation, whereas galaxies have no preferred orientation. Therefore, we make use of the pipeline architecture proposed by the winning solution (Dieleman et al., 2015) of the GalaxyZoo challenge on kaggle[7]: In order to grasp the rotational invariance, 16 different views of an image are constructed and fed into the same CNN. The results are combined via a set of three dense Maxout layers (Goodfellow et al., 2013) with two linear filters each. The use of a Maxout layer makes the representation more efficient, such that fewer parameters are used. The output is the predicted morphological labels. Finally, these are renormalized to fit the hierarchical structure of labels (see Section 2).

[7] https://www.kaggle.com/c/galaxy-zoo-the-galaxy-challenge

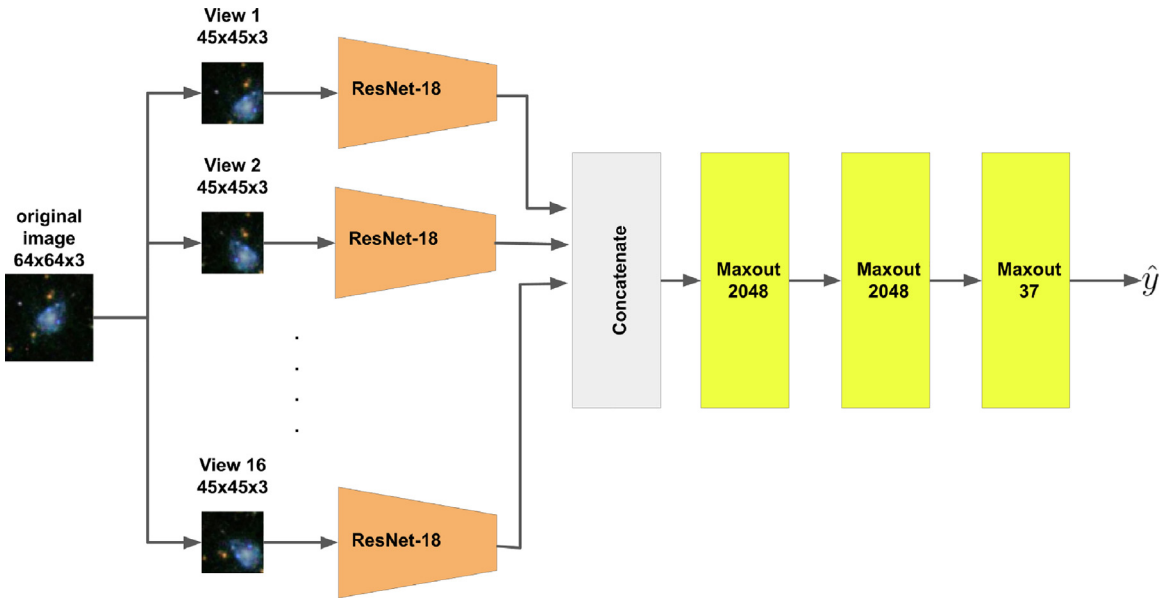

Fig. D.8. Schematic overview of the image classifier architecture. For the input RGB image of 64 × 64 pixels, 16 views are constructed by flipping the image, rotating by 45° and finally dividing into four overlapping crops of 45 × 45 pixels of each corner. Each of these views is fed to the same ResNet-18 convolutional network. The output is then concatenated and fed to a sequence of Maxout layers with the presented number of nodes. The final output is the predicted labels for the galaxy in the original image.

The 16 views of an image are constructed using three transformations. First, a second view is created by flipping the image horizontally. Second, two additional views are created by rotating the original and the flipped views by 45°. Third, each of the resulting four views is divided into four overlapping crops of 45 × 45 pixels, one for each corner. All of these are rotated such that the center of the galaxy is in the lower right corner.

An overview of the classifier pipeline is shown in Fig. D.8. We setup the classifier as a PYTORCH (Paszke et al., 2019) module. Instead of the convolutional network used by Dieleman et al. (2015), we make use of the state-of-the-art ResNet-18 architecture (He et al., 2016), provided with the TORCHVISION python package (Marcel and Rodriguez, 2010). The 512 output values of the last hidden layer of ResNet-18 for each of the 16 views are concatenated to a single vector. They are then fed into a sequence of two maxout layers with 2048 nodes and a final maxout layer with 37 nodes and ALReLU activation function.

The labels are not one-hot encoded and instead, a distribution of votes for a label is given per image (see Section 2). This label uncertainty holds some information on its own, since the morphology seen in an galaxy image can be ambiguous. Hence, we train the model to match the label distribution instead of simply matching the label with the highest score. Thus, the classification task becomes rather a regression problem, which is trained by using an MSE loss function. Furthermore, we fit the output layer to the hierarchical structure of labels. Though Softmax layers would provide valid probability distributions, Dieleman et al. (2015) report that they decrease the overall performance. Hence, we follow an alternative approach. The output nodes use an ALReLU (Mastromichalakis, 2020) activation and are renormalized according to the label hierarchy. For rare labels (see Appendix A), the use of ALReLU forces the classifier to predictions close to zero instead of drifting to high negative values. Positive predictions can thus be learned much easier, which helps to correctly classify rare galaxy types.

D.1.2. Training

We updated the model parameters using the Adam optimizer (Kingma and Ba, 2015). By using a larger batch size than Dieleman et al. (2015), using 512 instead of 16, better gradient information is provided to the models thus resulting in better image quality. For training optimization, we do not use the hierarchical normalization in the first epoch. Instead, we let the network train to reproduce the label distribution in the final layer. Furthermore, during training, dropout (Srivastava et al., 2014) was used previous to all three Maxout layers, hiding each input node with 50% probability, in order to improve model robustness and reduce overfitting.

The dataset is rather small for training a deep neural network. Thus, a key ingredient for good performance and extrapolation of results is the use of data augmentation. We use random rotation, translation and flipping, as explained in Section 2. Then the image is cropped to 64 × 64 pixels, used to construct the 16 views mentioned above.

For training, we use an initial learning rate of 0.04 and reduce it by factor 0.1 after 292 and 373 epochs. The performance is monitored using (Biewald, 2020). The training wall time for this model is 24 node hours.

D.1.3. Validation of results

Measuring the performance of a classifier on uncertain target labels is not straight forward. The evaluation of ambiguous samples is unclear. Certainly, it is important that labels with high scores are estimated correctly whereas an exact fit of low scores is not mandatory. Hence, the distribution within a particular question should be evaluated only if their total score is high enough. To estimate the accuracy regarding a question we thus only consider images where more than 50 % of participants answered this question. For these we check whether the label with the highest score in the prediction also has the highest score in the target. The accuracy for a particular question is given by the percentage of considered samples where this is true. We find that in the test set the accuracy of our classifier exceeds 90% for all questions. We thus outperform the model in Dieleman et al. (2015), which leads to question accuracy for rare types of about 70%. Note, however, that our classifier is significantly outperformed by the ZOOBOT model (Walmsley et al., 2022), which was published during the review process of this paper.

D.2. Conditional BigGAN

D.2.1. GAN

The general GAN architecture consists of two neural networks, the generator and the discriminator. These play an adversarial game (Goodfellow et al., 2014b). The discriminator is trained on images to provide binary output equal to one for images in the training set and zero for generated images. The generator transforms a random latent vector into an image. It is trained on the same loss function as the discriminator but with opposite target labels. Thus, the generator learns to fool the discriminator to identify the generated image as real. The two networks are trained alternatingly until they reach a Nash equilibrium and cannot improve any further. The intended final result is that the generated images are novel and yet no longer discernible from original images.

GANs have proven to be a powerful tool to generate unseen images of high quality in several contexts (Radford et al., 2015; Reed et al., 2016; Huang et al., 2017; Isola et al., 2017; Jin et al., 2017; Karras et al., 2017; Zhang et al., 2017; Zhu et al., 2017b; Brock et al., 2018; Wang et al., 2018; Wu et al., 2019). However, it has been shown that the relative diversity of the generated sample lags behind other generative models, especially autoregressive models and normalizing flows (Bond-Taylor et al., 2021). In turn, these generally show worse sample quality, at least when using identical one-hot encoded labels for the same object type. However, the uncertain nature of class labels in the galaxy dataset (see Section 2) allows to increase the variation within a certain type. This can be exploited to increase the diversity of samples generated with GANs.

D.2.2. BigGAN

We make use of the conditional BigGAN.[8] (Brock et al., 2018) It has been shown to consistently generate large images of high quality. The major improvement of BigGAN is obtained by using more computational resources, i. e. more model parameters and larger batches. By training more parameters, the capacity of the model is increased relative to the complexity of the dataset. By training on larger batches, more modes provide increased convergence of the gradient information. In addition, they further improve the architecture and training of generator and discriminator with a number of further adjustments.

Hinge loss Instead of adversarial loss, BigGAN is trained using hinge loss, which results in better performance.

Truncation trick During training, Gaussian random vectors are used. Better image quality can be achieved by using truncated Gaussian vectors. This comes as a trade-off between image quality and sample variety and therefore has to be tuned carefully.

Attention modules Attention layers are included in the BigGAN architecture. These are designed to attend to important features (Zhang et al., 2019) and thus reduce the required image processing power.

Conditional instance normalization The generator is provided with class information via class-conditional batch normalization (Dumoulin et al., 2016). An embedding layer predicts the parameters for batch normalization and thus allows to scale feature maps as required for the input class. The discriminator obtains the class information via projection, i. e. concatenated into intermediate layers, using a shared embedding for all class labels (Miyato et al., 2018).

Spectral normalization The weight matrices are renormalized before each training step, such that the spectral norm satisfies the Lipschitz constraint, i. e. none of the eigenvalues exceeds unity (Miyato et al., 2018). This stabilizes the discriminator during training.

Double discriminator training The discriminator is trained twice before the generator is trained once. This ensures that the discriminator stays ahead of the generator.

Moving average weights are updated using a moving average (Karras et al., 2017). This allows for a more accurate determination of the trend direction.

Orthogonal regularization The weight matrices are initialized as orthogonal matrices and regularized to stay close to orthogonal (Saxe et al., 2013). Orthogonal weights are desirable, as weight multiplication conserves the norm of the original matrix.

Skip-z connections The latent input vector is also fed into deeper layers of the network. This allows the class information to directly influence features at several resolutions.

D.2.3. Training

A schematic of the training pipeline for BigGAN is shown in Fig. D.9. The generator processes label and randomly generated latent vectors of 128 dimensions. A hinge loss is used to train the discriminator together with the generator. As a first test, we use constant class information input to verify that this architecture is capable of producing galaxy images of high quality. We train the generator with the same number of labels from the training set and randomly generated ones. The former focus the training on the regions of interest whereas the latter increase diversity in the label space. Furthermore, generated images are processed by the pre-trained classifier to force conditional image generation using an MSE class loss. The training wall time for this model is 24 node hours.

D.3. InfoSCC-GAN

D.3.1. InfoSCC-GAN

We use the information-theoretic stochastic contrastive conditional generative adversarial network (InfoSCC-GAN),[9] introduced in Kinakh et al. (2021). In addition to the generator and discriminator, this pipeline includes an encoder and classifier pair to enforce conditional image generation. Their interplay can be seen in Fig. D.10, stage 3. The generator is based on the stochastic generative model EigenGAN (He et al., 2021) with explorable latent space. We further use the Patch discriminator from Zhu et al. (2017a). The independent contrastive encoder is based on SimCLR (Chen et al., 2020). It has shown state-of-the-art performance in unsupervised learning on diverse datasets. As base model, we use ResNet-50 (He et al., 2016). The encoder can be used for: (i) internal latent exploration, (ii) feature metrics like VGG-loss (Ledig et al., 2017), (iii) feature extraction for the classification of the generated samples. The encoder provides the input for the classifier, which enables training of the generator on class loss. This classifier is a single MaxOut layer with AL-ReLU (Mastromichalakis, 2020) activation (cf. Appendix D.1). The output is further renormalized to follow the hierarchical label structure (see Section 2). More information on the architectures and their interplay can be found in Kinakh et al. (2021).

Both the encoder and classifier are trained separately before the generator. This is done in order to avoid training on

[8] https://github.com/shackste/galaxy-generator

[9] https://github.com/vkinakh/galaxy-zoo-generation

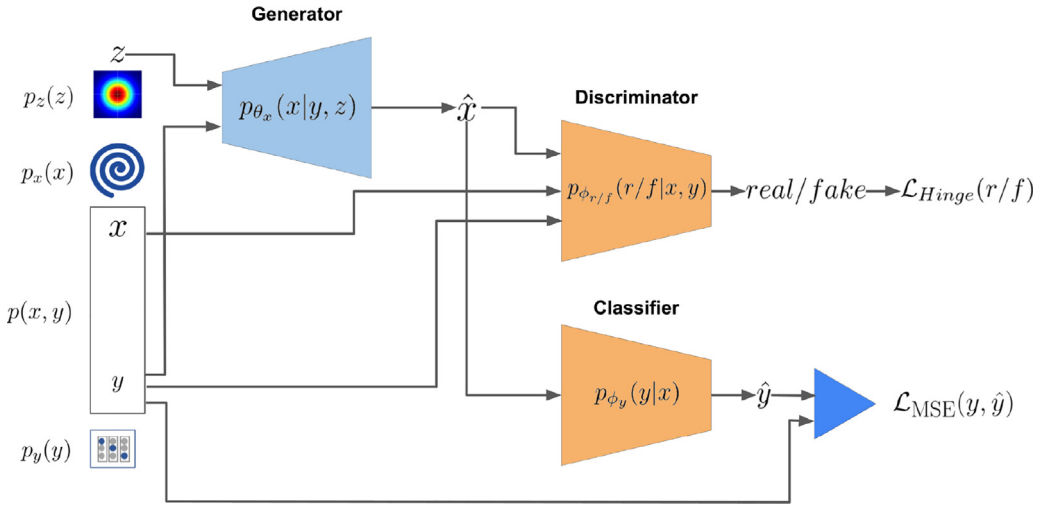

Fig. D.9. Training pipeline of BigGAN. The generator processes label and random latent vectors to generate images. The generator is trained on hinge loss to have the discriminator identify the generated image as real. The discriminator in turn is trained to identify the generated images as fake and original images as real. Furthermore, the generator is trained on MSE class loss, such that classifier returns the correct labels.

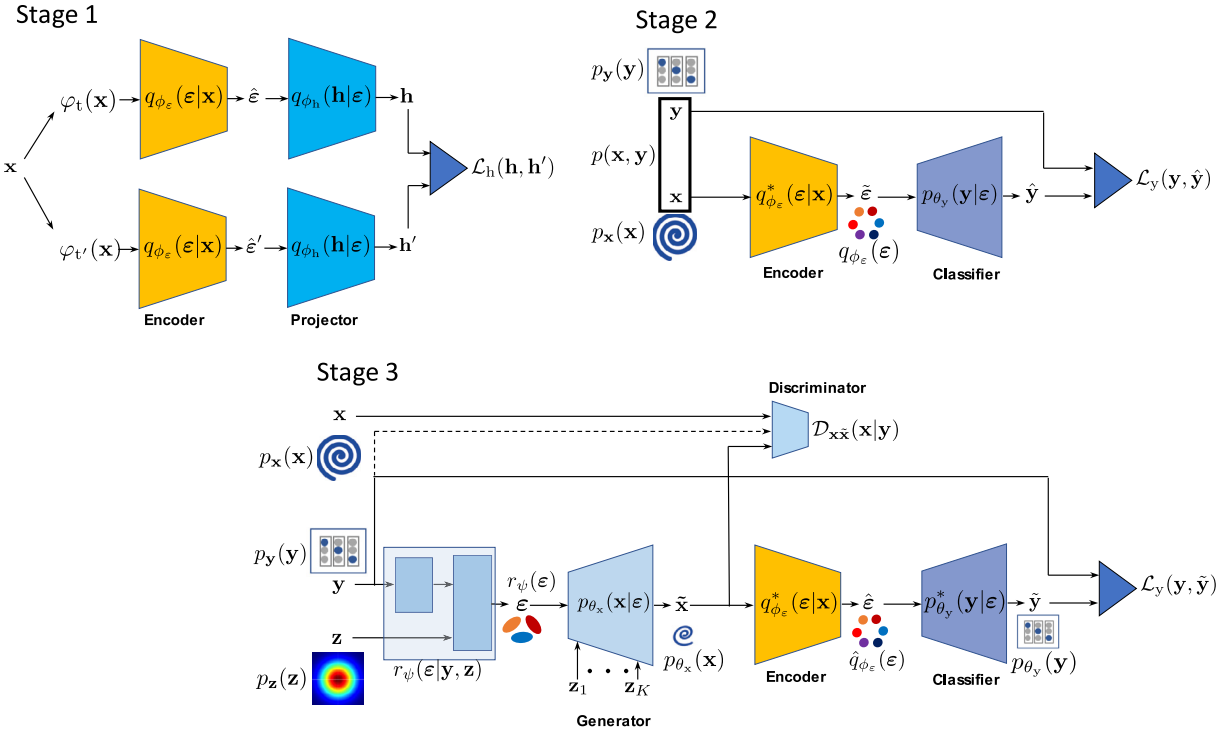

Fig. D.10. Training pipeline of InfoSCC-GAN. Stage 1. Contrastive encoder. Stage 2. Classifier. Stage 3. Contrastive generator. In stage 1, the encoder processes two augmentations of the same image and is trained on mutual information loss. In stage 2, the feature classifier processes encoded images and is trained on MSE class loss. In stage 3, the generator processes label and random latent vector to generate images. The generator is trained an adversarial loss to have the discriminator identify generated images as real. The discriminator in turn is trained to identify generated images as fake and original images as real. In addition, the generator is trained on MSE class loss, such that the encoder–classifier pair returns the correct labels.

poorly generated data in the early steps of training, where the model does not produce realistic images. The training of the InfoSCC-GAN thus includes 3 stages: stage (1) training of the encoder; stage (2) training of the classifier; stage (3) training of the conditional generator.

D.3.2. Training

Stage 1: encoder. A schematic of the training pipeline is shown in Fig. D.10, stage 1. The ResNet-50 encoder is fed with two augmentations of the same image. The output is processed by a projector model, which consists of three linear layers connected with ReLU activation. The results are used to compute the contrastive loss. By training the encoder in an unsupervised way, it learns the inner data distribution, which is then used to compare real and generated data. The SimCLR encoder is trained on contrastive NT-Xent loss (Sohn, 2016). In addition to the augmentations mentioned in Section 2, we use random color augmentation, random grayscale, random affine transformation and random gaussian blur. The encoder is trained for 200 epochs.

Stage 2: classifier. A schematic of the training pipeline is shown in Fig. D.10, stage 2. In contrast to the model introduced in

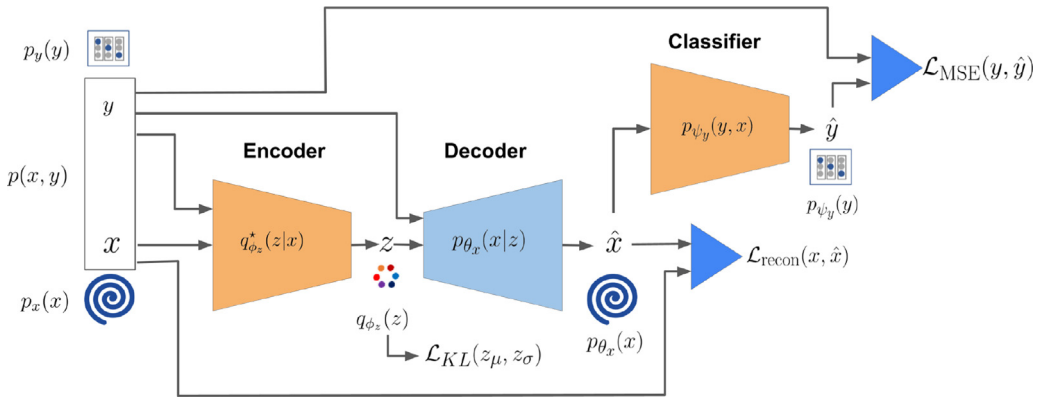

Fig. D.11. Training pipeline of cVAE.

Appendix D.1, the input of this classifier is the output of the encoder trained in stage 1. The classifier is trained on an MSE class loss for 200 epochs.

Stage 3: conditional generator. The training pipeline is schematically shown in Fig. D.10, stage 3. The label vector and random noise are prepared as generator input using an embedding layer. The discriminator uses the original labels to assess whether an image is real or fake. The adversarial loss is used to train both the generator and the discriminator. Furthermore, the generated image is processed by the pre-trained encoder and classifier. Conditional image generation is trained using an MSE class loss. Generator weights are updated such that labels predicted by the classifier are the same as the input labels. Usually, classification regularization is applied at each iteration. However, here we apply it at every 8-th iteration in order to balance between the generation of diverse samples and those with specified labels. When skipping more iterations the generated images do not represent the intended class, while skipping less leads to mode collapse. Since the adversarial loss varies strongly, it can saturate the class loss, preventing efficient conditional training. To avoid this, we update the generator on each loss separately. The training wall time for this model is 45 node hours.

D.4. Variational autoencoder

In order to test whether the evaluation metrics can identify better generators, we need to compare strong and weak models. VAEs are known to generate blurrier images than GANs. We hence make use of a rather simple conditional VAE[10] (cVAE) with a latent dimension size of 128. This model produces worse image quality and worse distribution quality than the more sophisticated models in the previous sections. We can therefore use it as a reference to assess the different metrics regarding these quality aspects.

The training procedure for the cVAE is shown in Fig. D.11. The encoder is a CNN with kernel size 3, where the second to last layer is concatenated with a label vector, suitably transformed to the same dimension as the latent vector. Each layer includes a BatchNorm with momentum of 0.99 and a LeakyReLU with a negative slope of 0.2. The decoder concatenates the latent vector with an embedded label vector of same size and uses bilinear upsampling to generate an image. We then use the classifier from Appendix D.1 to predict labels for the generated images. This cVAE is trained on an L1 reconstruction loss, a KL loss and an MSE class loss for 250 epochs. The training wall time for this model is 2 node hours.

[10] https://github.com/shackste/galaxy-generator

In addition to the cVAE we use the same encoder and decoder architectures, but without embedding labels. We use this VAE to reduce the dimensions of images to a feature space with 16 dimensions, which is the smallest latent space to generate sufficiently diverse images. While the cVAE is trained for the same number of epochs as the other generators for good comparison, the VAE is trained much longer to ensure convergence, required to obtain a meaningful lower-dimensional representation. It is trained on L1 reconstruction loss and a KL loss for 2000 epochs. The latent vector obtained from this encoder is then used for the evaluation metrics listed in Section 3.

D.5. Collapsed

A major problem with GANs is mode collapse. Instead of generating all types found in the training set, a collapsed model can result in good training scores by only providing a subset of these types with high quality. So far, mode collapsed GANs are mostly recognized by human inspection, which is a tough task in the context of galaxy images. We build an archetypal dummy model that always returns a single image taken from the dataset using the augmentations described in Section 2. By choosing a galaxy image of high quality, this model produces ideal image quality with worst possible distribution quality. It can thus be used as a reference to identify metrics that assess distribution quality and mode collapse.

Appendix E. Number of clusters

For computation of the cluster metrics (Section 3.1) we are using a pre-trained VAE. In order to determine the optimal number of clusters $k^\star$ we explore the dataset in the latent space of this VAE using two commonly used methods:

Elbow method. The distortion, i. e. the sum of squared distances of each point to its assigned cluster center, is plotted as a function of k, see Fig. E.12. The optimal choice for k is found where the distortion starts to decrease linearly, i. e. $k^\star = 13$.

Gap statistics. The gap $G(k)$ for k clusters is the difference in compactness, given by the intra-cluster distances, between the test set and a uniformly distributed set representing the null-hypothesis. The latter is obtained as the average of 10 realizations of the null-hypothesis, thus provides an estimate of the standard deviation $\sigma_G(k)$. The gap $G(k)$ increases monotonically with k thus does not provide a reasonable estimate for $k^\star$. Instead, we obtain $k^\star$ as the smallest k where $G(k) \geq G(k+1) - \sigma_G(k+1)$, which is $k^\star = 13$ (see Fig. E.13).

Both methods agree that $k^\star = 13$. We verify that the dataset is well represented by this $k^\star$ by computing the cluster metrics for

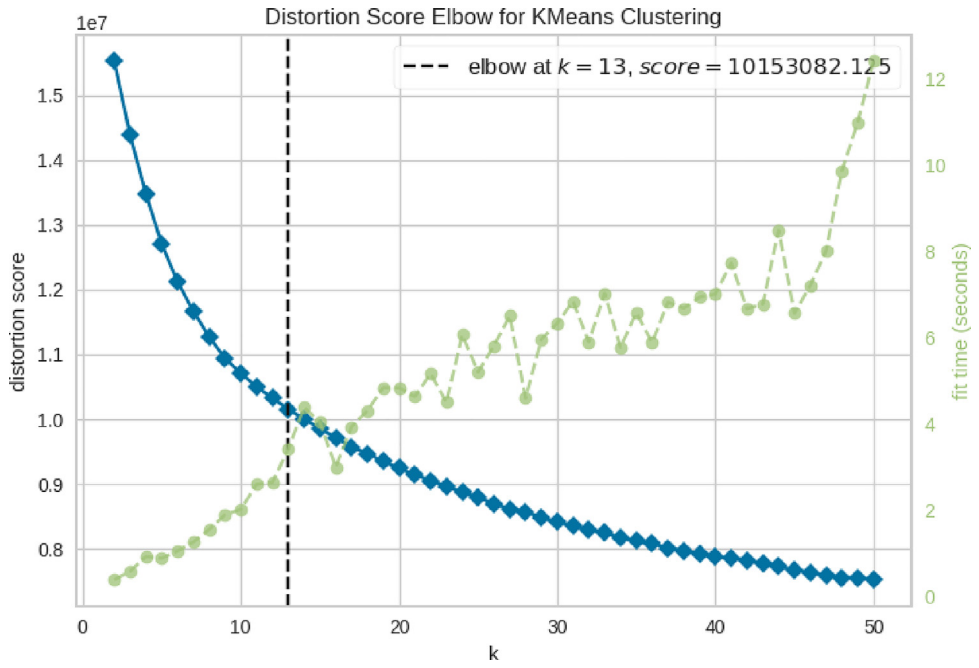

Fig. E.12. Elbow method for selecting the optimal number of clusters. The optimal number of clusters is 13 and is highlighted with a dashed line.

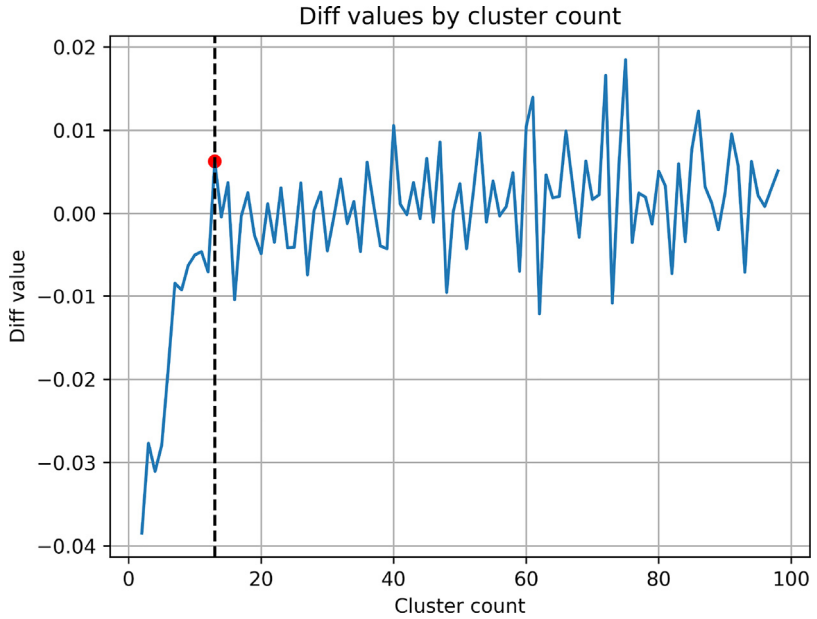

Fig. E.13. Gap statistics method for selecting the optimal number of clusters. Shown is the $G(k) - (G(k+1) - \sigma_G(k+1))$. The optimal number of clusters is 13 and is highlighted with a dashed line.

the same generated set for 10 different initializations of k-means. The resulting standard deviation is roughly 10 times lower than those reported in Table 2, which show the variation for different generated sets. This signals that the dataset is well described by $k^\star = 13$ clusters.

References

Abell, P.A., Allison, J., Anderson, S.F., Andrew, J.R., Angel, J.R.P., Armus, L., Arnett, D., Asztalos, S., Axelrod, T.S., Bailey, S., et al., 2009. Lsst science book, version 2.0. arXiv preprint arxiv:0912.0201.

Aggarwal, C.C., Hinneburg, A., Keim, D.A., 2001. On the surprising behavior of distance metrics in high dimensional space. In: International Conference on Database Theory. Springer, pp. 420–434.

Barchi, P., da Costa, F., Sautter, R., Moura, T., Stalder, D., Rosa, R., de Carvalho, R., 2017. Improving galaxy morphology with machine learning. arXiv preprint arXiv:1705.06818.

Biewald, L., 2020. Experiment tracking with weights and biases. https://www.wandb.com/.

Bińkowski, M., Sutherland, D.J., Arbel, M., Gretton, A., 2018. Demystifying MMD GANs. In: International Conference on Learning Representations. https://openreview.net/forum?id=r1lUOzWCW.

Blanton, M.R., Kazin, E., Muna, D., Weaver, B.A., Price-Whelan, A., 2011. Improved background subtraction for the sloan digital sky survey images. Astron. J. 142 (1), 31.

Bond-Taylor, S., Leach, A., Long, Y., Willcocks, C.G., 2021. Deep generative modelling: A comparative review of vaes, gans, normalizing flows, energy-based and autoregressive models. arXiv preprint arXiv:2103.04922.

Boquien, M., Burgarella, D., Roehlly, Y., Buat, V., Ciesla, L., Corre, D., Inoue, A., Salas, H., 2019. CIGALE: a python code investigating galaxy emission. Astron. Astrophys. 622, A103.

Bretonnière, H., Boucaud, A., Lanusse, F., Jullo, E., Merlin, E., Tuccillo, D., Castellano, M., Brinchmann, J., Conselice, C., Dole, H., et al., 2022. Euclid preparation-xIII. Forecasts for galaxy morphology with the euclid survey using deep generative models. Astron. Astrophys. 657, A90.

Brock, A., Donahue, J., Simonyan, K., 2018. Large scale GAN training for high fidelity natural image synthesis. arXiv preprint arXiv:1809.11096.

Castelvecchi, D., et al., 2017. Astronomers explore uses for AI-generated images. Nat. 542 (7639), 16–17.

Chen, T., Kornblith, S., Norouzi, M., Hinton, G., 2020. A simple framework for contrastive learning of visual representations. arXiv preprint arXiv:2002.05709.

Conselice, C.J., 2003. The relationship between stellar light distributions of galaxies and their formation histories. Astrophys. J. Suppl. Ser. 147 (1), 1–28. doi:10.1086/375001.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., Fei-Fei, L., 2009. Imagenet: A large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition. Ieee, pp. 248–255.

Dia, M., Savary, E., Melchior, M., Courbin, F., 2019. Galaxy image simulation using progressive GANs. arXiv preprint arXiv:1909.12160.

Dieleman, S., Willett, K.W., Dambre, J., 2015. Rotation-invariant convolutional neural networks for galaxy morphology prediction. Mon. Not. R. Astron. Soc. 450 (2), 1441–1459.

Domínguez Sánchez, H., Huertas-Company, M., Bernardi, M., Tuccillo, D., Fischer, J., 2018. Improving galaxy morphologies for SDSS with deep learning. Mon. Not. R. Astron. Soc. 476 (3), 3661–3676.

Dumoulin, V., Shlens, J., Kudlur, M., 2016. A learned representation for artistic style. arXiv preprint arXiv:1610.07629.

Ferreira, L., Conselice, C.J., Duncan, K., Cheng, T.-Y., Griffiths, A., Whitney, A., 2020. Galaxy merger rates up to z 3 using a Bayesian deep learning model: A major-merger classifier using illustristng simulation data. Astrophys. J. 895 (2), 115.

Feydy, J., Séjourné, T., Vialard, F.-X., Amari, S.-i., Trouve, A., Peyré, G., 2019. Interpolating between optimal transport and MMD using sinkhorn divergences. In: The 22nd International Conference on Artificial Intelligence and Statistics. pp. 2681–2690.

Fortson, L., Masters, K., Nichol, R., Edmondson, E., Lintott, C., Raddick, J., Wallin, J., 2012. Galaxy zoo. Adv. Mach Learn Data Mining Astronomy 2012, 213–236.

Freeman, P.E., Izbicki, R., Lee, A.B., Newman, J.A., Conselice, C.J., Koekemoer, A.M., Lotz, J.M., Mozena, M., 2013. New image statistics for detecting disturbed galaxy morphologies at high redshift. Mon. Not. R. Astron. Soc. 434 (1), 282–295. doi:10.1093/mnras/stt1016.

Fussell, L., Moews, B., 2019. Forging new worlds: high-resolution synthetic galaxies with chained generative adversarial networks. Mon. Not. R. Astron. Soc. 485 (3), 3203–3214.

Goodfellow, I.J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y., 2014a. Generative adversarial nets. In: Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2. NIPS '14, MIT Press, Cambridge, MA, USA, pp. 2672–2680.

Goodfellow, I.J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y., 2014b. Generative adversarial networks. arXiv preprint arXiv:1406.2661.

Goodfellow, I.J., Warde-Farley, D., Mirza, M., Courville, A., Bengio, Y., 2013. Maxout networks. In: Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28. ICML '13, JMLR.org, pp. III–1319–III–1327.

Hartigan, J.A., Wong, M.A., 1979. Algorithm AS 136: A k-means clustering algorithm. J. R. Statist Soc. Ser C (Appl. Statist) 28 (1), 100–108.

He, Z., Kan, M., Shan, S., 2021. Eigengan: Layer-wise eigen-learning for GANs. In: International Conference on Computer Vision. ICCV.

He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 770–778.

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S., 2017. GANs trained by a two time-scale update rule converge to a local nash equilibrium. Adv. Neural Inf. Process. Syst. 30.

Ho, J., Jain, A., Abbeel, P., 2020. Denoising diffusion probabilistic models. Adv. Neural Inf. Process. Syst. 33, 6840–6851.

Holzschuh, B.J., O'Riordan, C.M., Vegetti, S., Rodriguez-Gomez, V., Thuerey, N., 2022. Realistic galaxy images and improved robustness in machine learning tasks from generative modelling. Mon. Not. R. Astron. Soc..

Huang, R., Zhang, S., Li, T., He, R., 2017. Beyond face rotation: Global and local perception gan for photorealistic and identity preserving frontal view synthesis. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 2439–2448.

Isola, P., Zhu, J.-Y., Zhou, T., Efros, A.A., 2017. Image-to-image translation with conditional adversarial networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1125–1134.

Jin, Y., Zhang, J., Li, M., Tian, Y., Zhu, H., Fang, Z., 2017. Towards the automatic anime characters creation with generative adversarial networks. arXiv preprint arXiv:1708.05509.

Karras, T., Aila, T., Laine, S., Lehtinen, J., 2017. Progressive growing of gans for improved quality, stability, and variation. arXiv preprint arXiv:1710.10196.

Karras, T., Aittala, M., Laine, S., Härkönen, E., Hellsten, J., Lehtinen, J., Aila, T., 2021. Alias-free generative adversarial networks. In: Beygelzimer, A., Dauphin, Y., Liang, P., Vaughan, J.W. (Eds.), Advances in Neural Information Processing Systems. https://openreview.net/forum?id=Owggnutk6lE.

Khalifa, N.E., Taha, M.H., Hassanien, A.E., Selim, I., 2018. Deep galaxy V2: Robust deep convolutional neural networks for galaxy morphology classifications. In: 2018 International Conference on Computing Sciences and Engineering. ICCSE, IEEE, pp. 1–6.

Kinakh, V., Drozdova, M., Quétant, G., Golling, T., Voloshynovskiy, S., 2021. Information-theoretic stochastic contrastive conditional GAN: InfoSCC-GAN. In: Bayesian Deep Learning NeurIPS Workshop.

Kingma, D.P., Ba, J., 2015. Adam: A method for stochastic optimization. In: Bengio, Y., LeCun, Y. (Eds.), 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings. http://arxiv.org/abs/1412.6980.

Kingma, D.P., Welling, M., 2014. Auto-Encoding Variational Bayes. In: 2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings. arXiv:http://arxiv.org/abs/1312.6114v10.

Kynkäänniemi, T., Karras, T., Laine, S., Lehtinen, J., Aila, T., 2019. Improved precision and recall metric for assessing generative models. Adv. Neural Inf. Process. Syst. 32.

Lacey, C.G., Baugh, C.M., Frenk, C.S., Benson, A.J., Bower, R.G., Cole, S., Gonzalez-Perez, V., Helly, J.C., Lagos, C.D., Mitchell, P.D., 2016. A unified multiwavelength model of galaxy formation. Mon. Not. R. Astron. Soc. 462 (4), 3854–3911.

Lanusse, F., Mandelbaum, R., Ravanbakhsh, S., Li, C.-L., Freeman, P., Póczos, B., 2021. Deep generative models for galaxy image simulations. Mon. Not. R. Astron. Soc. 504 (4), 5543–5555.

Laureijs, R., Amiaux, J., Arduini, S., Augueres, J.-L., Brinchmann, J., Cole, R., Cropper, M., Dabin, C., Duvet, L., Ealet, A., et al., 2011. Euclid definition study report. arXiv preprint arXiv:1110.3193.

Laureijs, R., Gondoin, P., Duvet, L., Criado, G.S., Hoar, J., Amiaux, J., Auguères, J.-L., Cole, R., Cropper, M., Ealet, A., et al., 2012. Euclid: Esa's mission to map the geometry of the dark universe. In: Space Telescopes and Instrumentation 2012: Optical, Infrared, and Millimeter Wave. 8442, International Society for Optics and Photonics, p. 84420T.

Ledig, C., Theis, L., Huszár, F., Caballero, J., Aitken, A.P., Tejani, A., Totz, J., Wang, Z., Shi, W., 2017. Photo-realistic single image super-resolution using a generative adversarial network. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 105–114.

Lin, Y., Wahba, G., Zhang, H., Lee, Y., 2002. Statistical properties and adaptive tuning of support vector machines. Mach. Learn. 48 (1), 115–136.

Lintott, C.J., Schawinski, K., Slosar, A., Land, K., Bamford, S., Thomas, D., Raddick, M.J., Nichol, R.C., Szalay, A., Andreescu, D., et al., 2008. Galaxy zoo: morphologies derived from visual inspection of galaxies from the sloan digital sky survey. Mon. Not. R. Astron. Soc. 389 (3), 1179–1189.

Lotz, J.M., Primack, J., Madau, P., 2004. A new nonparametric approach to galaxy morphological classification. Astron. J. 128 (1), 163–182. doi:10.1086/421849.

Lovell, C.C., Acquaviva, V., Thomas, P.A., Iyer, K.G., Gawiser, E., Wilkins, S.M., 2019. Learning the relationship between galaxies spectra and their star formation histories using convolutional neural networks and cosmological simulations. Mon. Not. R. Astron. Soc. 490 (4), 5503–5520.

Marcel, S., Rodriguez, Y., 2010. Torchvision the machine-vision package of torch. In: Proceedings of the 18th ACM International Conference on Multimedia. MM '10, Association for Computing Machinery, New York, NY, USA, pp. 1485–1488. doi:10.1145/1873951.1874254.

Mastromichalakis, S., 2020. Alrelu: A different approach on leaky ReLU activation function to improve neural networks performance. CoRR https://arxiv.org/abs/2012.07564.

Miyato, T., Kataoka, T., Koyama, M., Yoshida, Y., 2018. Spectral normalization for generative adversarial networks. arXiv preprint arXiv:1802.05957.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala, S., 2019. Pytorch: An imperative style, high-performance deep learning library. In: Wallach, H., Larochelle, H., Beygelzimer, A., d' Alché-Buc, F., Fox, E., Garnett, R. (Eds.), Advances in Neural Information Processing Systems 32. Curran Associates, Inc., pp. 8024–8035.

Peth, M.A., Lotz, J.M., Freeman, P.E., McPartland, C., Mortazavi, S.A., Snyder, G.F., Barro, G., Grogin, N.A., Guo, Y., Hemmati, S., Kartaltepe, J.S., Kocevski, D.D., Koekemoer, A.M., McIntosh, D.H., Nayyeri, H., Papovich, C., Primack, J.R., Simons, R.C., 2016. Beyond spheroids and discs: classifications of CANDELS galaxy structure at 1.4 2 via principal component analysis. Mon. Not. R. Astron. Soc. 458 (1), 963–987. doi:10.1093/mnras/stw252.

Pillepich, A., Springel, V., Nelson, D., Genel, S., Naiman, J., Pakmor, R., Hernquist, L., Torrey, P., Vogelsberger, M., Weinberger, R., et al., 2018. Simulating galaxy formation with the illustristng model. Mon. Not. R. Astron. Soc. 473 (3), 4077–4106.

Primack, J., Dekel, A., Koo, D., Lapiner, S., Ceverino, D., Simons, R., Snyder, G., Bernardi, M., Chen, Z., Domínguez-Sánchez, H., et al., 2018. Deep learning identifies high-z galaxies in a central blue nugget phase in a characteristic mass range. Astrophys. J. 858 (2), 114.

Radford, A., Metz, L., Chintala, S., 2015. Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint arXiv:1511.06434.

Ravanbakhsh, S., Lanusse, F., Mandelbaum, R., Schneider, J., Poczos, B., 2017. Enabling dark energy science with deep generative models of galaxy images. In: Proceedings of the AAAI Conference on Artificial Intelligence. 31, (1).

Ravi, N., Reizenstein, J., Novotny, D., Gordon, T., Lo, W.-Y., Johnson, J., Gkioxari, G., 2020. Accelerating 3D deep learning with PyTorch3D.

Reed, S., Akata, Z., Yan, X., Logeswaran, L., Schiele, B., Lee, H., 2016. Generative adversarial text to image synthesis. In: International Conference on Machine Learning. PMLR, pp. 1060–1069.

Regier, J., McAuliffe, J., Prabhat, M., 2015. A deep generative model for astronomical images of galaxies. In: NIPS Workshop: Advances in Approximate Bayesian Inference.

Rodriguez-Gomez, V., Snyder, G.F., Lotz, J.M., Nelson, D., Pillepich, A., Springel, V., Genel, S., Weinberger, R., Tacchella, S., Pakmor, R., Torrey, P., Marinacci, F., Vogelsberger, M., Hernquist, L., Thilker, D.A., 2019. The optical morphologies of galaxies in the IllustrisTNG simulation: a comparison to Pan-STARRS observations. Mon. Not. R. Astron. Soc. 483 (3), 4140–4159. doi:10.1093/mnras/sty3345.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B., 2022. High-resolution image synthesis with latent diffusion models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10684–10695.

Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., Chen, X., 2016. Improved techniques for training GANs. In: Proceedings of the 30th International Conference on Neural Information Processing Systems. NIPS '16, Curran Associates Inc., Red Hook, NY, USA, pp. 2234–2242.

Saxe, A.M., McClelland, J.L., Ganguli, S., 2013. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. arXiv preprint arXiv:1312.6120.

Shamir, L., 2011. Ganalyzer: A tool for automatic galaxy image analysis. Astrophys. J. 736 (2), 141.

Simonyan, K., Zisserman, A., 2015. Very deep convolutional networks for large-scale image recognition. In: Bengio, Y., LeCun, Y. (Eds.), 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings. http://arxiv.org/abs/1409.1556.

Smith, M.J., Geach, J.E., 2019. Generative deep fields: arbitrarily sized, random synthetic astronomical images through deep learning. Mon. Not. R. Astron. Soc. 490 (4), 4985–4990.

Smith, M.J., Geach, J.E., Jackson, R.A., Arora, N., Stone, C., Courteau, S., 2022. Realistic galaxy image simulation via score-based generative models. Mon. Not. R. Astron. Soc. 511 (2), 1808–1818.

Sohn, K., 2016. Improved deep metric learning with multi-class n-pair loss objective. Adv. Neural Inf. Process. Syst. 29.

Somerville, R.S., Davé, R., 2015. Physical models of galaxy formation in a cosmological framework. Annu. Rev. Astron. Astrophys. 53, 51–113.

Song, J., Meng, C., Ermon, S., 2020. Denoising diffusion implicit models. arXiv preprint arXiv:2010.02502.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R., 2014. Dropout: a simple way to prevent neural networks from overfitting. J. Mach. Learn. Res. 15 (1), 1929–1958.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z., 2015. Rethinking the inception architecture for computer vision. CoRR http://arxiv.org/abs/1512.00567.

Thorndike, R.L., 1953. Who belongs in the family. In: Psychometrika. Citeseer.

Tibshirani, R., Walther, G., Hastie, T., 2001. Estimating the number of clusters in a data set via the gap statistic. J. R. Stat. Soc. Ser. B Stat. Methodol. 63 (2), 411–423.

Villani, C., 2008. Optimal Transport: Old and New. In: Grundlehren der mathematischen Wissenschaften, Springer Berlin Heidelberg, https://books.google.de/books?id=hV8o5R7_5tkC.

Wallace, G.K., 1991. The JPEG still picture compression standard. Commun. ACM 34 (4), 30–44.

Walmsley, M., Lintott, C., Géron, T., Kruk, S., Krawczyk, C., Willett, K.W., Bamford, S., Kelvin, L.S., Fortson, L., Gal, Y., et al., 2022. Zoobot: Deep learning galaxy morphology classifier. Astrophysics Source Code Library ascl–2203.

Walmsley, M., Smith, L., Lintott, C., Gal, Y., Bamford, S., Dickinson, H., Fortson, L., Kruk, S., Masters, K., Scarlata, C., et al., 2020. Galaxy zoo: probabilistic morphology through Bayesian CNNs and active learning. Mon. Not. R. Astron. Soc. 491 (2), 1554–1574.

Wang, T.-C., Liu, M.-Y., Zhu, J.-Y., Tao, A., Kautz, J., Catanzaro, B., 2018. High-resolution image synthesis and semantic manipulation with conditional gans. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 8798–8807.

Willett, K.W., Lintott, C.J., Bamford, S.P., Masters, K.L., Simmons, B.D., Casteels, K.R., Edmondson, E.M., Fortson, L.F., Kaviraj, S., Keel, W.C., et al., 2013. Galaxy zoo 2: detailed morphological classifications for 304 122 galaxies from the sloan digital sky survey. Mon. Not. R. Astron. Soc. 435 (4), 2835–2860.

Wu, H., Zheng, S., Zhang, J., Huang, K., 2019. Gp-gan: Towards realistic high-resolution image blending. In: Proceedings of the 27th ACM International Conference on Multimedia. pp. 2487–2495.

Yang, M., 2017. Deepo: set up deep learning environment in a single command line.. In: GitHub Repository. GitHub, https://github.com/ufoym/deepo.

York, D.G., Adelman, J., Anderson Jr., J.E., Anderson, S.F., Annis, J., Bahcall, N.A., Bakken, J., Barkhouser, R., Bastian, S., Berman, E., et al., 2000. The sloan digital sky survey: Technical summary. Astron. J. 120 (3), 1579.

Zhang, H., Goodfellow, I., Metaxas, D., Odena, A., 2019. Self-attention generative adversarial networks. In: International Conference on Machine Learning. PMLR, pp. 7354–7363.

Zhang, H., Xu, T., Li, H., Zhang, S., Wang, X., Huang, X., Metaxas, D.N., 2017. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 5907–5915.

Zhu, J.-Y., Park, T., Isola, P., Efros, A.A., 2017a. Unpaired image-to-image translation using cycle-consistent adversarial networks. In: Computer Vision (ICCV), 2017 IEEE International Conference on.

Zhu, J.-Y., Park, T., Isola, P., Efros, A.A., 2017b. Unpaired image-to-image translation using cycle-consistent adversarial networks. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 2223–2232.