Galaxy Project ETHZ CIL 2020

Philippe Blatter*, Lucas Brunner*, Alicja Chaszczewicz* and Simon Heinke*
* Group: 4stroneers
Department of Computer Science
ETH Zurich, Switzerland
Emails: pblatter@ethz.ch, brunnelu@ethz.ch,
achaszcze@ethz.ch, sheinke@ethz.ch

***Abstract*—In this work we use the given galaxy image dataset and train a Wasserstein GAN (WGAN) to generate images of galaxies which are indistinguishable for the human eye from the original image samples. Furthermore, we compare the performance of the trained WGAN model against the state-of-the-art StyleGAN2 pretrained with additional *Google Sky* data. Finally, we demonstrate how to finetune the WGAN critic in order to predict *galaxyness* scores. We achieve 0.10394 mean absolute error on the public Kaggle test dataset.**

***Index Terms*—Generative Adversarial Network, Galaxy Image Generation, Similarity Score Prediction**

I. Introduction

Training deep neural networks requires huge amounts of data, thus the demand for extensive datasets has recently surged. Due to the lack of adequate resources, generative models have experienced a considerable increase in relevance as the generation of realistic data enables the training of state-of-the-art deep neural networks in fields that suffer from data shortage. One type of generative models are Generative Adversarial Networks (GANs [1]), which are commonly used for synthesis of natural images [2], images of human faces [3] and various other types of image data. GANs have also been successfully used on astronomical data to advance research in astrophysics dark energy science [4] and to recover features in astrophysical images [5].

In our work, we train GANs to capture the concept of *galaxyness*. We use the given dataset consisting of reference cosmology images and their *galaxyness* scores, representing how similar an image is to a prototypical cosmology image. We demonstrate how to generate high-quality cosmology images based on the given dataset of cosmological reference images. Furthermore, we show how to use the discriminatory network in order to learn the similarity function (*galaxyness* scores), which can then be applied in a score prediction task to a set of unseen query images.

All of our code is available at: https://gitlab.ethz.ch/sheinke/cil_astroneers/.

II. Models and Methods

In the sections A, B and C we provide an analysis of the given dataset and describe the methods that were applied for preprocessing. Moreover, in the sections D-G we give an overview of the used models and optimization procedures, explaining the rationale behind them.

A. Data analysis

In our work we use a grayscale 1000x1000 pixels image dataset consisting of:

1) 1200 labeled images: Real cosmology images are labeled as class 1, whereas non-cosmology or corrupted images are labeled as class 0. The label distribution is imbalanced, with 200 0-labeled images and 1000 real cosmology images.

A representative example of a cosmology image can be seen in figure 1. Images are predominantly black and depict distant astrophysical objects. They have a complex underlying structure that is invisible to the human eye.

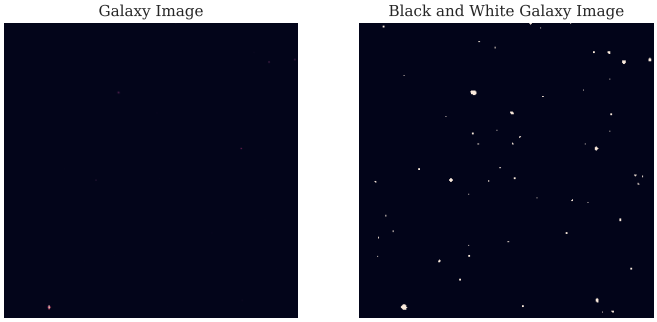


Fig. 1: A representative example of a cosmology image (left). Setting all pixels with higher-than-average brightness to white reveals a complex structure that is undetectable by means of visual inspection (right).

2) 9600 scored images: The scored images are drawn from the same distribution as the labeled images, but instead of labels they have real-valued scores in the range $[0.0, 8.0]$. Higher scores mean greater similarity to a "prototypically ideal" galaxy image. There is no overlap between labeled and scored images but a visual analysis of the dataset revealed that all images with scores above 0.01 are cosmology images (real or corrupted). Based on some basic assumptions on the process of image sampling for the scored and labeled images, we are confident with over 99% probability that the number of scored non-real images is lower than 2100. This implies that with high confidence we can expect that non-real images have scores of 0.86 at most. Please see details of this analysis in appendix B.

The score distribution is highly imbalanced, with peaks around the scores 0.0 and 1.5, as shown in figure 2. It is

Fig. 2: Highly imbalanced distribution of scores with major peaks around 0.0 and 1.5.

impossible for a layperson to accurately score the cosmological images by means of visual analysis. Therefore, we assume the similarity scores to be the output of an unknown function, that we will treat as a blackbox.

B. Data augmentation and Preprocessing

1) Data selection: In the training process of the generative model we make use of both the labeled and the scored images. We filter out all images from the scored set with scores lower than 1.0 so that our model learns to generate higher quality samples. Moreover, this threshold ensures filtering of all non-cosmology and corrupted images (see II-A2). From the labeled dataset we include all images labeled as real cosmology data.

For the task of scoring query images, we only use the scored images with their corresponding scores, which we normalize using the mean and the standard deviation of the training set. A training and a validation set is generated by a random split of data with 90% and 10% of samples respectively.

2) Image resolution: The given images have a very high resolution of 1000x1000. Working with such large images is computationally demanding and more training data is needed to accurately learn full scale data characteristics. One might consider cropping the training data into smaller images to simplify the training and to obtain more data samples. The astrophysical objects in the images are small compared to the image size, so the cropping should preserve their local structure. However, such an approach ignores global image geometry and might introduce artifacts if small images were to be combined to form a full-sized cosmology image. Therefore, we operate on full-size images exclusively, and perform data augmentation in order to increase the number of training samples.

3) Data augmentation: We assume that the *galaxyness* score of a galaxy is invariant to 90-degree rotations of the depicting image. Therefore we use all original images together with their 90-, 180- and 270-degrees rotations. Due to the discrete nature of image data, rotations by other angles introduce interpolation artifacts and our assumption of rotation-invariance no longer holds. To further augment the data, we additionally shift and randomly crop the images, adding black pixel padding where needed to ensure 1000x1000 size. The galaxy images have mostly low intensity pixels (see appendix A) so we consider padding to cause low alteration to galaxy image characteristics. Finally, we double the size of the augmented dataset by flipping the previously obtained images.

C. Additional Data Source

Despite the augmentation procedure the variability of the obtained training dataset samples is limited. In order to mitigate the effects of the limited dataset size on the generative model training, we perform transfer learning experiments using Google Sky [6] images of the night sky. Those images are different to the galaxy images in our goal dataset, nevertheless we expect pretraining on this additional data to help to gather statistics about distribution of star sizes, structure types of stars and their relative positions.

We use 10% of all the Google Sky pictures available with the highest zoom factor (highest zoom-factor is not available everywhere because of the telescope positions on Earth). Concatenating the collected images and using crops of this larger picture results in a theoretically unlimited number of pretraining data samples. For every epoch, we use a different set of 80.000 images. We further refer to this newly created pretraining dataset as the *Google Sky dataset* (example in appendix H).

D. Galaxy Generation Task: Baseline Models

For the galaxy image generation task we experiment with two basic generative models.

1) Convolutional Variational Autoencoder: A variational autoencoder (VAE) [7] consists of an encoder network that maps high-dimensional inputs into a latent space, and a decoder network that reconstructs the original input from this latent representation. By introducing a density model over the latent space the decoder network can be used to generate new data. VAEs are known to be easier to train than adversarial architectures, thus in our experiments we first considered a convolutional VAE model (CVAE). We tested various combinations of hyperparameters for the dimensionality of the latent space and the depth and size of the encoder and decoder networks. Irrespective of any particular experimental setting, the CVAE training never produced anything else than completely black images.

In the full-size 1000x1000 pixel images individual stars and galaxies each only consist of a dozen pixels at most. Vanilla CVAEs have a tendency to produce blurry images [8], thus we conclude that our basic CVAE implementation may be incapable of learning such finegrained features.

2) Deep Convolutional Generative Adversarial Network: Deep convolutional generative adversarial networks (DCGANs) are based on regular GANs [1], but use deep convolutional networks for the generator and the discriminator, leading to an increased training stability [9].

In spite of the architectural improvements of the normal GAN, the training of the DCGAN model on the galaxy dataset yielded poor results as the generated images turned out to be predominantly black. One possible explanation for the bad training performance of the DCGANs is the vanishing gradient problem. If the discriminator is able to correctly distinguish between generated and real images, the gradient used in the update step of the generator vanishes.

We found the performance of the analysed basic models unsatisfactory, hence they are excluded from further analysis. We achieve good results by using models that do not suffer from the previously mentioned limitations of CVAEs or DCGANs.

E. Galaxy Generation Task: Wasserstein GAN

One of the main weaknesses of regular GANs is their instability during training. They can either suffer from vanishing gradients or mode collapse. The authors of the original GAN paper [1] directly proposed an alternative cost function to tackle the vanishing gradient problem. Arjovsky et al. [10] showed that the alternative cost function has a high variance. The solution proposed by [10] is the Wasserstein GAN (WGAN), which is optimized by a new cost function using the Earth-Mover (EM) or Wasserstein-1 distance. The main benefit is that the Wasserstein distance has a smoother gradient. Therefore, one can train the WGAN critic (discriminator) until optimality without the vanishing gradient.

The WGAN successfully learns to generate images that closely resemble the original samples after 4 epochs of training, please see the results analysis in the section III-A.

For the WGAN model details, please refer to appendix C.

F. Galaxy Generation Task: StyleGAN2

To ensure high quality of generated images (potentially higher than the ones generated by the WGAN) we decided to additionally train some state-of-the-art GAN architecture using the *Google Sky dataset* for pretraining. We analyzed the potential of multiple state-of-the-art GANs. However, many of them either required a huge amount of training data or an infeasible training time. StyleGAN [3] performs exceptionally well with an acceptable training time and relatively few model parameters. The first version of this model suffers from artifacts like raindrops on the generated images, therefore we use the improved version StyleGAN2 [11] that is able to remove many of those artifacts.

The original version of the StyleGAN2 was trained on RGB-images, thus we slightly change the model architecture to handle the given grayscale images. We also reduce the overall number of parameters by a factor of three by scaling the number of convolutional filters, such that the ratio of trainable parameters and input size stays approximately the same.

The number of our training images after data augmentation [1] is comparable to the number of face-images used in the original StyleGAN2. After about 75 epochs we start to get images which are hard to distinguish from the real cosmology images. Pretraining on the *GoogleSky dataset* helps even further, as after 3 epochs after pretraining the generated images look very similar to samples from the provided dataset.

For the result analysis please see the section III-A. For architectural and optimisation details we refer the reader to [11].

G. Score Prediction Task

The trained discriminator is able to differentiate between real and false image samples and thus has learned something about what constitutes a realistic cosmology image. Therefore, to predict scores for the given set of query images, we make use of transfer learning, using the trained WGAN discriminator. We apply the following training procedure:

1) Load the pretrained discriminator / critic and exclude its weights from further training (*layer freezing*).
2) Change the model output by adding new (untrained) layers to the second-to-last layer of the old model, for details please see appendix 5.
3) Train only the newly added layers until the validation error does not improve anymore.
4) Unfreeze the base-layers of the pretrained critic and train them jointly, saving the model with the best validation score.

As the base model we use the trained WGAN critic, which has more parameters than the StyleGAN2 discriminator and yielded better performance in early stages of the finetuning experiments.

To further improve the score prediction accuracy, we create a new model formed by an ensemble of five finetuned WGAN critics, which were trained with different validation sets. The similarity score is then computed as the average of all networks' outputs.

To judge the performance of our approach, we compare it against three baselines:

- Average score prediction.
- Standard convolutional neural network (CNN) network (details in appendix D).
- Feature based approach using gradient boosting regression trees on Coiflet wavelet coefficients energy proportion features (please see appendix E).

We also examine the impact of pretraining the critic as part of the GAN, by comparing the finetuned critic against an identical model with randomly initialized weights (note that in this experiment we directly train all layers jointly, since early frozen layers with random weights would only leave noise to the trainable layers at the end of the model). Please find the results analysis in the section III-B.

[1] For StyleGAN2 data augmentation we pad images to 1024x1024 due to architectural reasons, instead of 1000x1000. We also increase the score threshold to 1.25 as after pretraining on *Google Sky dataset* we prefer less data but of higher quality.

III. Results

A. Galaxy Image Generation

We do not have access to the ground truth similarity function, therefore we cannot judge the *galaxyness* of generated images directly. By means of visual inspection we can conclude that both the WGAN and the StyleGAN2 models produce full-size images that to the human eye look very similar to the original data samples (see appendix 9 and 10 for samples of generated images and appendix H for YouTube videos of the StyleGAN2 training process). We notice that the StyleGAN2 stars look more blurry than the ones from the original dataset or generated by the WGAN model.

To gain some additional insights about the quality of the images, we also score a large set of generated images using our WGAN critic score prediction model. We obtain score distributions for image samples generated by the two models (see figure 3).

Therefore, based on the assumption that our score prediction model approximates the unknown similarity function well, we observe that the WGAN images achieve mean score of 2.07 with 0.91 standard deviation, whereas the StyleGAN2 produces images with mean score of 1.73 and 0.46 standard deviation. The WGAN samples resemble the original distribution very well, just without the spike around 0, which is explained by the fact that the samples with low scores were filtered out before training. The StyleGAN2 generates much less diverse samples with less quality on average. However, the worst similarity score for StyleGAN2 images is 0.95, which is much higher than 0.25 in case of WGAN samples.

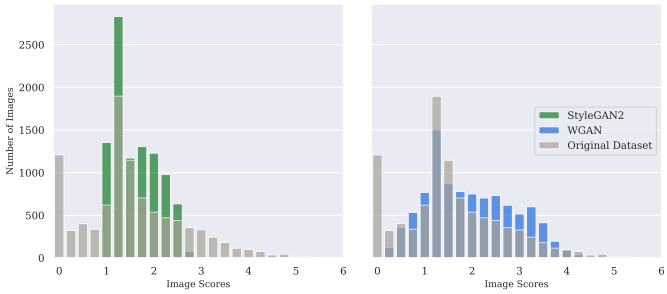


Fig. 3: Original score distribution (transparent grey) and predicted scores for StyleGAN2 images (green) and WGAN images (blue).

B. Accurate Score Prediction

We evaluate the models for the score prediction task using mean absolute error on the Kaggle public test set (from now on abbreviated as MAE_{KP}).

Model	MAE_{KP}
Average Score Prediction	0.885
Standard CNN	0.772
Feature-based Approach	0.289
Finetuned WGAN Critic	0.126
Ensemble of 5 WGAN Critics	**0.104**

Table 1: Mean absolute error on the Kaggle public set.

Constant prediction of the average score results in 0.885 MAE_{KP}. The standard CNN improves the result only by about 0.1, while the feature based model achieves MAE_{KP} of 0.289. The finetuned WGAN critic performs significantly better than all the baseline models with 0.126 MAE_{KP} score. The ensemble model achieves an improvement over that with the best 0.104 MAE_{KP} score [2].

We observe that transfer learning with the trained critic for the score prediction task is beneficial in several ways. While we keep the base-layers frozen, the correlation between train error and validation error is remarkably high, whereas when we use randomly initialized weights and train the model without any transfer learning, the correlation between train error and validation error seems almost absent. Moreover, the randomly initialized model performs much worse on average [3].

IV. Discussion

A. Image Generation

The WGAN models seems to mimic the image distribution much better than the StyleGAN2 model with the *Galaxy Sky Dataset* pretraining. We hypothesize that this might be an effect of:

- Image characteristics bias (e.g. blurry stars) introduced in the *Galaxy Sky Dataset* pretraining phase and not fully removed during the finetuning.
- Imperfect model evaluation method as the score prediction is based on the finetuned WGAN critic that might be biased towards WGAN-generated images.

B. Accurate Score Prediction

The final ensemble of 5 WGAN critics results in 0.10394 MAE_{KP}. We observe that for the training set our predictions have greater variance for high similarity scores, which is an effect of less data in the higher range of score values. To minimize the negative impact of this dataset imbalance, we experimented with rebalancing procedures. Our approach did not yield satisfactory results, however, we expect that more sophisticated methods of rebalancing might be beneficial. Adding to the ensemble a model tailored towards high score range predictions might also improve the prediction bias in this highest score range (see appendix F).

V. Summary

We presented an approach to generate full-size realistic cosmology images. Furthermore, we showed how to finetune the pretrained discriminator to the *galaxyness* score prediction task. Our end-to-end approach is capable of capturing the concept of *galaxyness*, both in terms of a generative approach as well as in a discriminatory sense.

[2] The experiments were run three times, reported results are averaged.

[3] In a few experiments the randomly initialized model achieved comparable validation errors, but it was impossible to systematically reproduce this due to training instability.

REFERENCES

[1] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[2] Andrew Brock, Jeff Donahue, and Karen Simonyan, "Large scale gan training for high fidelity natural image synthesis," *International Conference on Learning Representations*, 2018.

[3] Tero Karras, Samuli Laine, and Timo Aila, "A style-based generator architecture for generative adversarial networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[4] Siamak Ravanbakhsh, Francois Lanusse, Rachel Mandelbaum, Jeff Schneider, and Barnabas Poczos, "Enabling dark energy science with deep generative models of galaxy images," in *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[5] Kevin Schawinski, Ce Zhang, Hantian Zhang, Lucas Fowler, and Gokula Krishnan Santhanam, "Generative adversarial networks recover features in astrophysical images of galaxies beyond the deconvolution limit," *Monthly Notices of the Royal Astronomical Society: Letters*, vol. 467, no. 1, pp. L110–L114, 2017.

[6] Google, "Google sky," https://www.google.com/sky/, Accessed: 10.04.2020.

[7] Diederik P Kingma and Max Welling, "Auto-encoding variational bayes," *International Conference on Learning Representations*, 2014.

[8] Jianmin Bao, Dong Chen, Fang Wen, Houqiang Li, and Gang Hua, "Cvae-gan: Fine-grained image generation through asymmetric training," *Proceedings of the IEEE international conference on computer vision*, 2017.

[9] Alec Radford, Luke Metz, and Soumith Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv:1511.06434*, 2015.

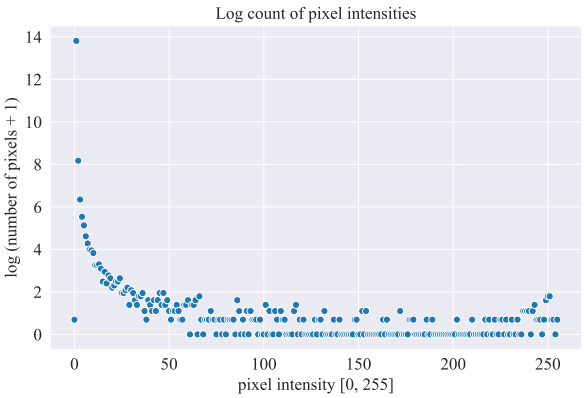
[10] Martin Arjovsky, Soumith Chintala, and Léon Bottou, "Wasserstein gan," *International Conference on Machine Learning*, 2017.

[11] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila, "Analyzing and improving the image quality of stylegan," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 8110–8119.

APPENDIX

A. Data Analysis Pixel Counts

We observe that the vast majority of galaxy image pixels are black (0-3 intensity in [0,255] range).



Appendix 1: Distribution of pixel intensities for a representative cosmology image. We observed that the distributions of pixel intensities for cosmological images look like the one depicted above, sometimes shifted slightly to the left or right, with the average intensity ranging from 0 to 3.

B. Score Threshold Analysis

We know that the labeled images are drawn from the same distribution as the scored images. We make an additional natural assumption that each image from the scored and labeled set was sampled independently with probability p of being a non-real image. Based on this assumption, we can estimate with high confidence the number of non-real images in the scored dataset and a score threshold that should filter them out. Please see R code below.

```
# use package binom
require(binom)

# load scored and labeled datasets
scored <- read.csv("scored.csv")
labeled <- read.csv("labeled.csv")

# count images
labeled.total <- nrow(labeled)
labeled.nonreal <- nrow(labeled[labeled$Actual == 0,])
scored.total <- nrow(scored)

# compute confidence interval for p - probability of nonreal image
# for binomial distribution with labeled.total trials
conf.interval <- binom.confint(labeled.nonreal, labeled.total, conf.level = 0.999)

# choose the most restrictive bound (the highest upper bound for p)
maxi.ind <- which.max(conf.interval$upper)
highest.p <- conf.interval[maxi.ind, "upper"]

# find 0.999 quantile of the Binomial distribution
# with p = highest.p and number of trails = scored.total
quant <- qbinom(0.999, scored.total, highest.p)
print(quant)
# output 2092

# find corresponding score threshold
scores.sorted <- sort(scored$Actual)
print(scores.sorted[quant])
# output 0.8503561
```

C. Network Architectures and Hyperparameters: WGAN

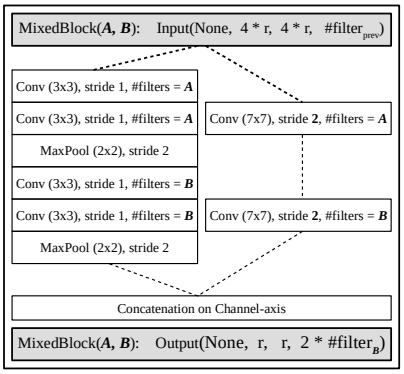
The following explains the network architectures of the generator and the critic of the Wasserstein GAN and provides a deeper insight into the hyperparameters and other implementation details.

Generator: Appendix 2 illustrates the network architecture of the generator network used in the WGAN.

Layer Type	Output Shape	Param #
InputLayer	[(None, 200)]	0
Dense	(None, 8192)	1638400
Reshape	(None, 8, 8, 128)	0
Conv2DTranspose	(None, 16, 16, 128)	409728
Conv2DTranspose	(None, 32, 32, 64)	204864
Conv2DTranspose	(None, 64, 64, 64)	102464
Conv2DTranspose	(None, 128, 128, 32)	100384
Cropping2D	(None, 125, 125, 32)	0
Conv2DTranspose	(None, 250, 250, 16)	25104
Conv2DTranspose	(None, 500, 500, 8)	6280
Conv2DTranspose	(None, 1000, 1000, 1)	393
Total params: 2,487,617		

Appendix 2: Network architecture of the generator network with a total of 2,487,617 parameters.

Critic: Appendix 4 illustrates the network architecture of the critic network used in the WGAN. To simplify the explanation, we first define a building block of the discriminator (see appendix 3).



Appendix 3: Building block of the discriminator: A "MixedBlock" combines small, pooled convolutions with large, strided convolutions.

Layer Type	Output Shape	Param #
InputLayer	(None, 1000, 1000, 1)	0
Conv2D	(None, 1000, 1000, 8)	80
Conv2D	(None, 1000, 1000, 8)	584
MaxPooling2D	(None, 500, 500, 8)	0
MixedBlock(A=16, B=32)	(None, 125, 125, 64)	48784
MixedBlock(A=32, B=64)	(None, 31, 31, 128)	283936
MixedBlock(A=64, B=128)	(None, 7, 7, 256)	1938048
Flatten	(None, 12544)	0
Dense	(None, 1)	12545
Total params: 2,283,977		

Appendix 4: Network architecture of the critic network with a total of 2,283,977 parameters.

We use Adam as an optimizer with a learning rate of $1e-4$. We set beta_1 to 0.5 and beta_2 to 0.9, in order to mitigate the momentum-effect, as suggested in [10]

Layer Type	Output Shape	Param #
InputLayer	(None, 1000, 1000, 1)	0
Conv2D	(None, 1000, 1000, 8)	80
Conv2D	(None, 1000, 1000, 8)	584
MaxPooling2D	(None, 500, 500, 8)	0
MixedBlock(A=16, B=32)	(None, 125, 125, 64)	48784
MixedBlock(A=32, B=64)	(None, 31, 31, 128)	283936
MixedBlock(A=64, B=128)	(None, 7, 7, 256)	1938048
Conv2D	(None, 5, 5, 256)	590080
Conv2D	(None, 3, 3, 512)	1180160
Flatten	(None, 4608)	0
Dense	(None, 512)	2359808
Dense	(None, 1024)	525312
Dense	(None, 1)	1025
Total params: 6,937,033		

Appendix 5: Network architecture of the score prediction model with a total of 6,937,033 parameters, of which 4,660,993 were newly added to the pretrained WGAN critic.

We use Adam as an optimizer with a learning rate of $2e-3$ when training with frozen baselayers and $3e-4$ when training all layers jointly. We reduce the learning rate when the validation error stagnates.

D. Network Architecture and Hyperparameters: CNN baseline

Appendix 6 explains the network architecture of the baseline CNN.

Layer Type	Output Shape	Param #
InputLayer	(None, 1000, 1000, 1)	0
Conv2D	(None, 498, 498, 4)	104
Conv2D	(None, 247, 247, 8)	808
Conv2D	(None, 122, 122, 16)	3216
Conv2D	(None, 59, 59, 32)	12832
Conv2D	(None, 28, 28, 64)	51264
Conv2D	(None, 12, 12, 128)	204928
Conv2D	(None, 4, 4, 256)	819456
Flatten	(None, 4096)	0
Dense	(None, 1)	4097
Total params: 1,098,737		

Appendix 6: Network architecture of the basic CNN baseline network with a total of 1,098,737 parameters. BatchNorm and LeakyReLU were applied after every Conv-layer.

We use Adam as an optimizer with a learning rate of $1e-3$. We reduce the learning rate when the validation error stagnates.

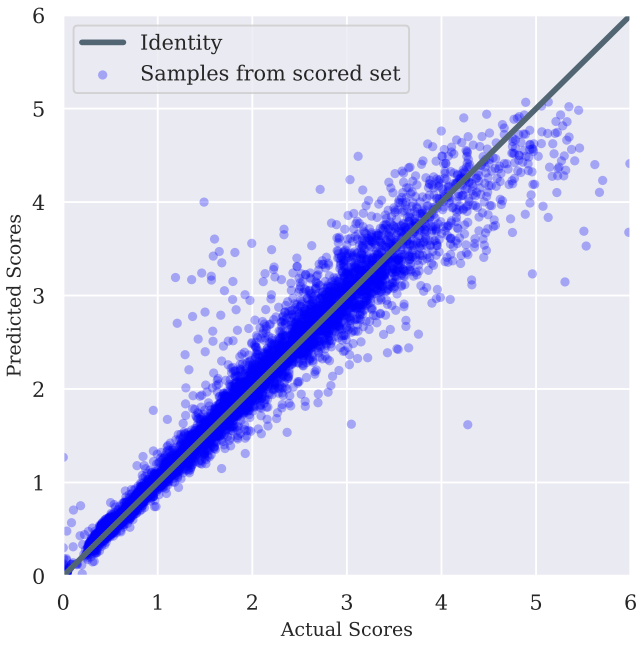
E. Model Details: Feature based baseline

Since we observed a correlation between some basic features and the similarity scores (see appendix 7), we consider as an additional baseline a model based on extracted features. Since signal is very localised in the galaxy images, we choose as our baseline a model based on the Discrete Wavelet Transform (DWT). Specifically, we train a histogram-based gradient boosting regression tree on features of proportions of energy in coefficients of Coiflet DWT with a maximal refinement level of 10. The histogram-based gradient boosting regression was trained with the least absolute deviation loss, maximal number of iterations 500 and maximal number of leaf nodes of 100 (other parameters were left as default in the scikit-learn implementation).

Appendix 7: Observed relation between image scores and the percentage of pixels with intensities above average. For all images with high scores (i.e. above 2.0), there are very few (i.e. less than 20 percent) pixels with above average intensity. The opposite is true for low-score images. For majority of scores around 0.0, there are over 60 percent of pixels with above average intensity.

F. Error Analysis: Score Prediction

Appendix 8 illustrates the performance of our approach for the task of assigning similarity scores on the training dataset.

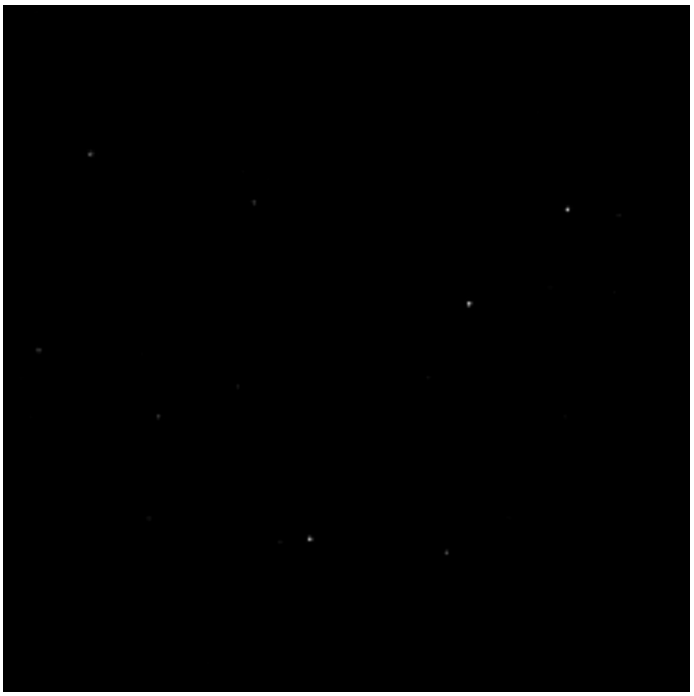


Appendix 8: Score predictions of the finetuned discriminator for the train set (i.e. the scored subset).

G. Galaxy Image Generation Results

The following two figures show some generated images. The cosmology image in figure appendix 9 has been generated with the WGAN, whereas the cosmology image in figure appendix 10 has been generated with the StyleGAN2.

Appendix 9: Full-size image, generated by the trained WGAN.

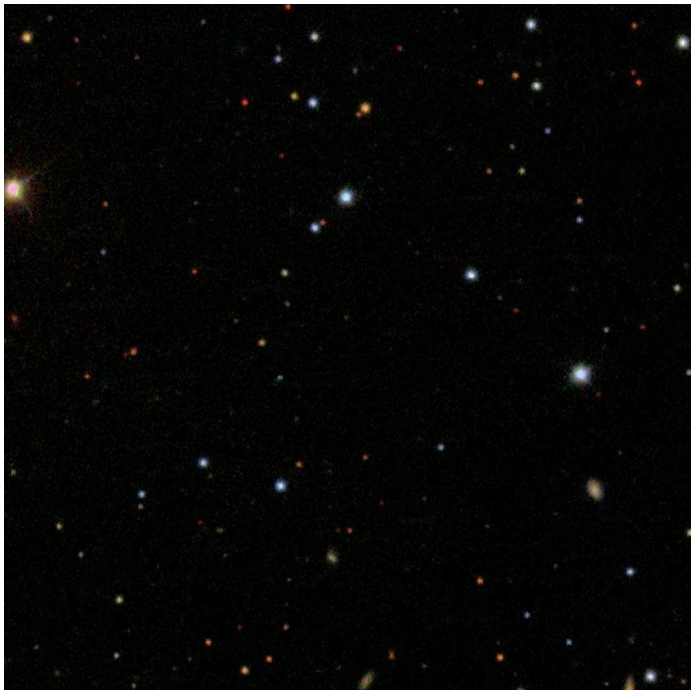


Appendix 10: Full-size image, generated by the StyleGAN2 after finetuning on the given cosmology dataset.

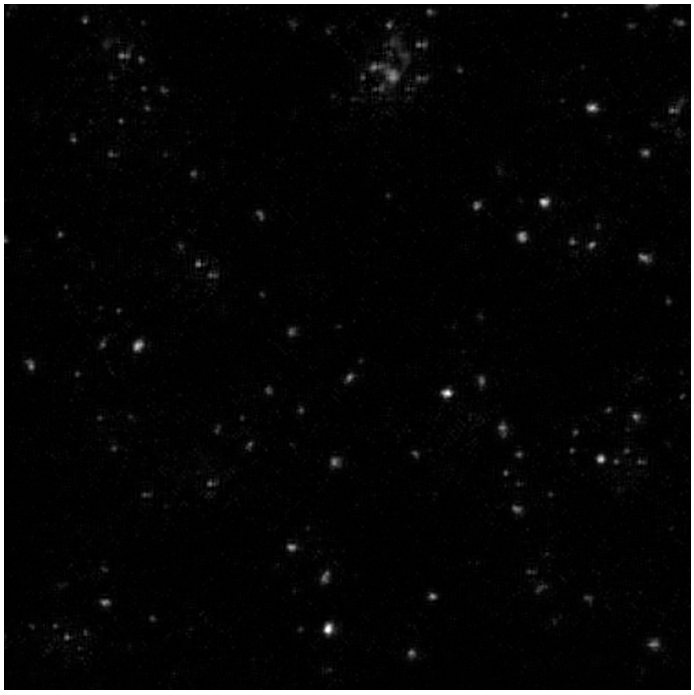
H. Google Sky and StyleGAN2

Pretraining process of the StyleGAN2 with *Google Sky dataset*: https://youtu.be/ojSu4vEa7tA

Finetuning process of the StyleGAN2 with the original galaxy dataset: https://youtu.be/eAKhglFyjpo



Appendix 11: Example Image from Google Sky.



Appendix 12: Full-size image, generated by our StyleGAN2 after pretraining on *Google Sky*.