Deep Generative Models for High-Resolution Cosmology Images

Till Schnabel, Sven Kellenberger, Hannes Pfammatter, Michelle Woon
Group: Galaxy Crusaders, Department of Computer Science, ETH Zurich, Switzerland

***Abstract*—This paper explores various machine learning models for generating sparse, high-resolution cosmology images. The contribution of this paper is twofold. Besides the design of models for cosmology image generation, a similarity function, which is part of our data, is learned.**

The generative models are compared to a simple baseline model and evaluated by estimating the values of the similarity function. We found that extracting the most important features from the images and only designing generative models for these features produced the best results. Our approximation of the similarity function achieved second place on both the public and private test data set on Kaggle [1].

I. Introduction

Generative modeling is not a new area of research. However, with the recent advent of deep generative models the research interest in this area has increased drastically.

In this project, we study the generation of cosmology images. Our cosmology images are sparse, high-dimensional data, which has proven to be an interesting problem suitable for the application of mathematical methods and machine learning models.

We split our project into two parts. First, we designed multiple machine learning models for cosmology image generation. Our starting point was a deep convolutional generative adversarial network (DCGAN) capable of generating cosmology images with high resolution. As this model was not able to capture important features such as stars with enough detail, we decided to focus the remaining models on the generation of these critical features only. For the generation of stars only, we designed a variational autoencoder (VAE) and a DCGAN. Our final solution consists of a conditional DCGAN (cDCGAN) that is able to generate several classes of stars that were found by clustering.

Second, we designed a convolutional neural network (CNN) and used an off-the-shelf random forest (RF) regression model to learn the similarity function which is part of our data. Finally, we evaluated our generative models by estimating the similarity scores of the generated images.

II. Models and Methods

The data set used for this paper was obtained from Kaggle. It consists of labeled images (where label $\in \{0, 1\}$ depending on if it is a cosmology image or not), scored images (where score $\in [0\ , 8]$ based on how much the image looks like a galaxy image) and query images (to be scored and uploaded to Kaggle).

We built all our neural networks using the Keras (Chollet et al. [2]) API on the TensorFlow (Abadi et al. [3]) backend and trained them on a single NVIDIA Tesla V100-SXM2 32 GB GPU on the Leonhard Cluster. The RF models were run locally.

A. Approximation of the Similarity Function

1) Random forest regression baseline: Random forests were introduced in 1995 by Ho [4] and further developed in 1999 by Breiman [5]. In this work, they were used as an off-the-shelf model for regression. By improving on this baseline we also achieved our best result for approximating the similarity function.

For training, we used histograms of image properties as input features. We started with a histogram of pixel intensities which we consider our baseline model. In 2007, Bosch et al. [6] used random forests for image classification. They captured information on shape by computing histograms of oriented gradients (HOG) of edges inside image regions. We computed a similar histogram for the entire image and combined it with the histogram of pixel intensities, which improved the accuracy of the regression.

What worked best was a histogram of the power spectrum of the images. To capture the power spectrum in a histogram, we used the fast Fourier transform (FFT) and applied a log transformation. Lastly, we searched for regions of interest (ROIs) by filtering out high frequencies, low frequencies or certain orientations.

2) CNN baseline: We used a convolutional neural network (CNN) with a deep architecture (see for instance LeCun et al. [7]) as our second model for approximating the similarity function. To speed up computations, we performed most experiments on images with dimensions 125×125 and 250×250. We experimented with convolution depth, normalization (see Ioffe and Szegedy [8] for BatchNorm), number of features, residual connections (He et al. [9]), preprocessing and augmentation. Our final architecture is shown in Figure 1. The network contains a total amount of $228\,680$ trainable parameters. The images are all preprocessed with the log-transformed power spectrum. For training, we augmented the data using horizontal and vertical flips with probability 0.5 as well as random spatial shifts by up to $20\,\%$. The model was trained for 140 epochs with a mean absolute error (MAE) loss function.

B. Cosmology Image Generation

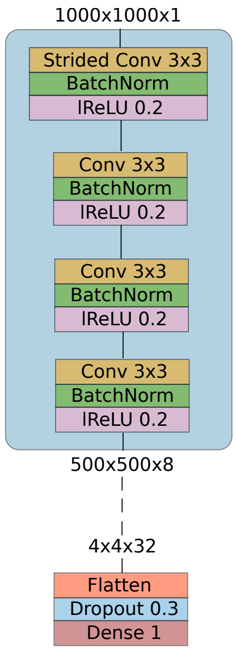

Figure 1. Illustration of the CNN architecture. "(Strided) Conv 3x3" denotes a convolutional layer with a 3×3 kernel; the stride is 2 for "Strided Conv" and 1 for the rest. The input is an image of dimension $1000 \times 1000 \times 1$. The spatial resolution is reduced by a factor of 2 in height and width per stacked convolutoinal block (colored in blue), of which there are 8 in total. The padding layer used for mapping the spatial resolution from 125×125 to 128×128 is omitted. The number of feature maps increases by a factor of 2 with each block until 32 is reached. "Dense 1" outputs the final 1-dimensional prediction score.

1) Adhoc Generator (AG) baseline: Cosmology images in the given data sets are, in essence, white stars on a black background. As such we were able to use a simple tiling method for our easy baseline, where we copied stars from the available data and placed them onto a black background. Using the available labeled data, we detected stars in all the cosmology images by finding contours in the image. Doing this also allowed us to find the minimum s_{min} and maximum s_{max} amount of stars in a cosmology image.

For each image to generate, we started with a black base image. Then, we selected a random number s_{rand} between s_{min} and s_{max}. Finally, we extracted s_{rand} stars from random source cosmology images and placed them on random spots into the destination image.

2) Large DCGAN: The deep convolutional generative adversarial network (DCGAN) was introduced in 2015 by Radford et al. [10]. It is a generative adversarial network (GAN) where both generator and discriminator are convolutional neural networks (CNNs) with architectural constraints which stabilize training. We used a large DCGAN to generate entire cosmology images.

Our implementation is based on a reference implementation on the TensorFlow website [11] as well as on the original paper and implementation. As proposed by Odena et al. [12] we used nearest neighbor interpolation and convolution instead of transposed convolution.

For training, all labeled images and all scored images in the data set with similarity score greater or equal to 2.61 were used. The pixel intensities were normalized to the range $[-1\ ,\ 1]$ and the images were padded to dimensions 1024×1024 to simplify upsampling and strided convolutions. Generator and discriminator contain a total amount of $6\,491\,125$ trainable parameters and were trained with the cross entropy loss.

3) VAE on stars baseline: In this approach, we focused on the generation of the most important features only. We used a variational autoencoder (VAE) to generate images of stars and placed them within a black background.

The VAE was introduced in 2014 by Kingma and Welling [13]. It allows the encoding of data points to low-dimensional latent representations $\sim \mathcal{N}(0, I)$ and to generate new data points by decoding arbitrary latent vectors $\sim \mathcal{N}(0, I)$. This is achieved by learning to map each data point x to a Gaussian distribution $q_\phi(z|x)$, determined by the mean μ and standard deviation σ, and each latent vector z to a Bernoulli distribution $p_\theta(x|z)$ determined by parameter p.

Our implementation is based on the original paper as well as on the tutorial on variational autoencoders by Doersch [14] and on a reference implementation on the TensorFlow website [15]. Because stars are shaped in a similar fashion, a simple multilayer perceptron (MLP) with a single hidden layer of size 500 is used for both the probabilistic encoder $q_\phi(z|x)$ and the probabilistic decoder $p_\theta(x|z)$. The parameters of the MLPs are denoted by ϕ and θ. The latent dimension is set to 16.

For training, we only used the labeled images. As with the adhoc generator in Section II-B1, we extracted the stars and centered them inside images of size 28×28. The pixel intensities were normalized to the range $[0\ ,\ 1]$. The MLPs contain a total amount of $811\,816$ trainable parameters. To create cosmology images, we distributed the generated star images randomly inside an image with a black background. We drew the number of stars from a normal distribution estimated from the labeled images and rounded to unsigned integers.

4) cDCGAN on stars: Taking the same approach as with the VAE, we also trained a smaller DCGAN on our data set of extracted stars. In order to control the star distribution in an image more precisely, we decided to cluster the star images. Considering our large data set of about $15\,000$ star images, most of them looked very similar, which is why we chose to divide them into five distinct classes. To achieve that, we trained a simple deep convolutional autoencoder (DCAE) on the star images for 400 epochs and applied k-means clustering on each image's latent code. We did not have to worry about overfitting the DCAE, because it was only used on the data it was trained on.

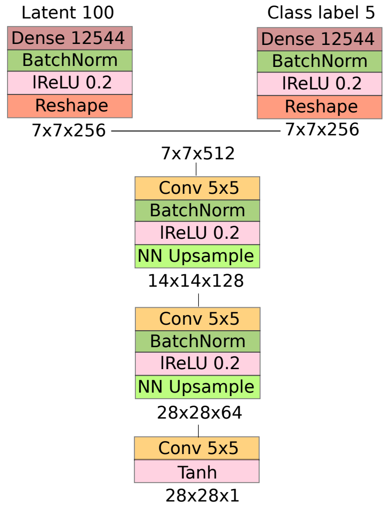

Figure 2. Illustration of the architecture of the conditional generator. A latent vector of size 100 and a class label vector of size 5 are each mapped to a vector of size $12\,544$ by a dense layer, then reshaped to 3D and combined. "Conv 5x5" denotes a 1-strided convolutional layer with a 5×5 kernel, "NN Upsample" a nearest neighbor upsampling in spatial sizes by a factor of 2.

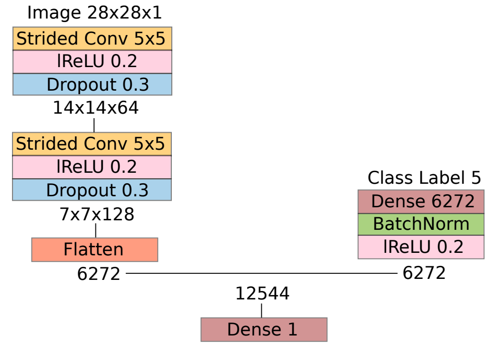

Figure 3. Illustration of the architecture of the conditional discriminator. The image of dimension $28 \times 28 \times 1$ is downsampled by two convolutional layers with stride 2 and a 5×5 kernel (denoted by "Strided Conv 5x5), flattend and then combined with the class label's latent code, which was mapped from dimension 5 to 6272 by a dense layer. "Dense 1" maps this combined latent code to a 1-dimensional prediction for the image.

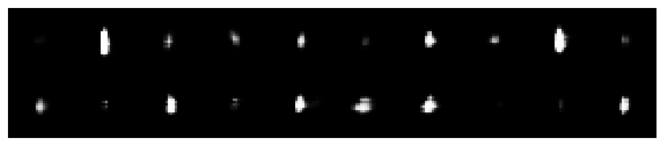

Figure 4. Random selection of stars generated by the large DCGAN.

Afterwards, we used the clustered data to train a conditional DCGAN (cDCGAN). Beside the image/latent code, the generator/discriminator was also fed the class label belonging to a star. In order to find a good distribution for the final images, we measured the number of occurrences of each star class per cosmology image and approximated it with a normal distribution bound to unsigned integers. We still assumed the positioning of a star to be distributed uniformly. We then placed the stars generated by the cDCGAN into 100 background images and repeated this process 2000 times to find those random numbers that produced the highest mean similarity scores (MSS) as estimated by our random forest (RF) and CNN models. We included this deterministic image stitching and MSS estimation as validation score into the training of our final cDCGAN.

The architecture of the cDCGAN was adopted from the reference implementation on the TensorFlow website [11]. No adjustments to the resolution had to be made, only the conditional property had to be added. The architecture of the conditional generator is shown in Figure 2. It uses a total amount of $3\,212\,480$ trainable parameters. The discriminator uses $269\,313$ trainable parameters. Its architecture is illustrated in Figure 3. The total amount of parameters is high compared to e.g. the CNN and could have been reduced. However, since the generation of such small images is very stable, no architectural experiments were conducted. The cDCGAN was trained for 185 epochs with LSGAN (Mao et al. [16]) as loss function, and about half the training time was spent on validation.

III. Results

Table I shows the results of our different random forest regression models (RF). By using histograms of different image properties, we were able to improve the accuracy of the regression as measured by the MAE by a factor of approximately 3 with respect to the baseline.

All of our models for approximating the similarity function were evaluated on Kaggle. The results are shown in Table II. For the CNN, we additionally calculated the MAE on our local $20\,\%$ validation split. Since a RF is not prone to overfitting, it was trained without a local validation split and local results are omitted. We included training and scoring time for these models in Table IV.

For our generative models, we decided to follow the evaluation process of the Kaggle competition and used our best RF model and the CNN to estimate the mean similarity score of the generated images. The results are displayed in Table III. The time needed for training and the generation of 100 images is shown in Table V.

As shown in Figure 4, the large DCGAN did not manage to capture the shape of important features in detail. Figure 5 shows linear interpolations between vectors in the latent space of the variational autoencoder (VAE) to demonstrate that the model works well. Figure 6 shows samples of the five different star classes generated by the cDCGAN showing that they are distinct.

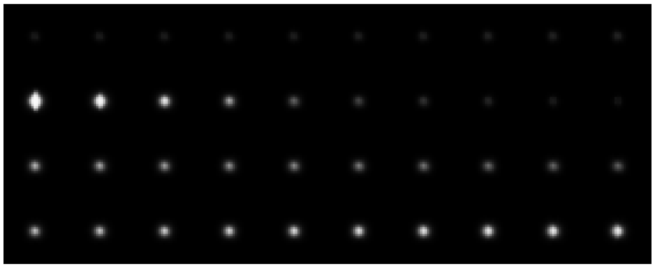

Figure 5. Linear interpolation between vectors in the latent space of the variational autoencoder on stars.

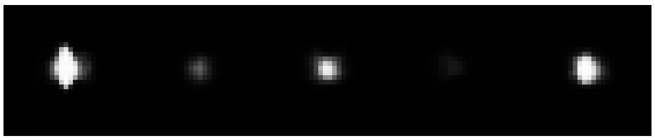

Figure 6. Samples of the five different star classes generated by the cDCGAN.

	# features	MAE (pub.)	MAE (priv.)
Pixel intensity (baseline)	32	0.258	0.287
Pixel intensity + oriented gradients	$32 + 36$	0.183	0.203
Power spectrum	48	0.107	0.124
Power spectrum + ROI	$48 + 48$	0.095	0.105

Table I
MEAN ABSOLUTE ERROR (MAE) FOR RANDOM FOREST REGRESSION USING DIFFERENT FEATURES.

Model	MAE (loc.)	STD (loc.)	MAE (pub.)	MAE (priv.)
RF (baseline)	-	-	0.269	0.288
CNN	0.174	0.279	0.168	0.188
RF (best)	-	-	0.095	0.105

Table II
MEAN ABSOLUTE ERROR (MAE) AND STANDARD DEVIATION OF THE ABSOLUTE ERROR (STD) OF OUR MODELS APPROXIMATING THE SIMILARITY FUNCTION.

Model	RF MSS	RF STD	CNN MSS	CNN STD
AG (baseline)	1.718	0.910	1.450	0.697
large DCGAN	1.069	0.717	1.154	0.773
VAE	1.690	0.555	2.031	0.662
cDCGAN	3.018	1.220	2.746	1.048

Table III
MEAN SIMILARITY SCORE (MSS) AND STANDARD DEVIATION OF THE SIMILARITY SCORE (STD) OF THE IMAGES GENERATED BY OUR MODELS AS ESTIMATED BY OUR CNN AND BEST RF MODEL.

Approximator	Training time	Scoring time
RF (baseline)	40 min	2 min
CNN	24 h	1 min
RF (best)	1.5 h	2 min

Table IV
TRAINING AND SCORING TIME FOR OUR VARIOUS SIMILARITY FUNCTION APPROXIMATORS (INCLUDING PREPROCESSING).

Generator	Training time	Generation time
AG	-	$<$1 min
large DCGAN	2.5 h	$<$1 min
VAE	2 min	$<$1 min
cDCGAN	30 min	$<$1 min

Table V
TRAINING AND GENERATION TIME FOR OUR VARIOUS GENERATORS.

IV. DISCUSSION

From Table II we can observe that all of our models for approximating the similarity function outperform the baseline. From our experiments and also from Table II we find that using FFT on the sparse cosmology images allowed our models to perform better. Presently, classification is mostly dominated by CNNs. Still, our RF model performed significantly better than our CNN, which we have taken to be our hard baseline. We thus conclude that for sparse data of limited range, there are still methods beside neural networks that can be useful.

Table III shows that the cDCGAN achieves the highest mean similarity score (MSS) of all models and also significantly outperforms the baseline in terms of MSS. Besides the highest MSS, the cDCGAN also reaches the highest standard deviation (STD). This is because we try to approximate the star distribution with “good” random numbers by choosing those giving the highest MSS. An additional neural network could have learnt to produce the “perfect” set of cosmology images by learning the star distribution on its own.

The large DCGAN does not exceed the performance of the baseline and is outperformed by both cDCGAN and the baseline set by the VAE. The poor performance of the large DCGAN is caused by the shape of the stars it produces and might also be affected by the placement of stars in an image. This shows that we were indeed able to improve our results by focusing on the generation of key features only.

V. SUMMARY

By extracting the most important features from our cosmology images and by designing generative models for only these, we were able to obtain highly realistic results.

For the approximation of the similarity function, we found that our cosmology images are best characterized by their power spectrum. Lastly, despite being more complex, a CNN does not always necessarily outperform a simpler model such as random forest.

REFERENCES

[1] (2019) cil-cosmology-2019. [Online]. Available: https://inclass.kaggle.com/c/cil-cosmology-2019/overview

[2] F. Chollet *et al.*, "Keras," https://keras.io, 2015.

[3] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, software available from tensorflow.org. [Online]. Available: http://tensorflow.org/

[4] T. K. Ho, "Random decision forests," in *Proceedings of the Third International Conference on Document Analysis and Recognition (Volume 1)*, 1995, pp. 278–. [Online]. Available: http://dl.acm.org/citation.cfm?id=844379.844681

[5] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, Oct. 2001. [Online]. Available: https://doi.org/10.1023/A:1010933404324

[6] A. Bosch, A. Zisserman, and X. Munoz, "Image classification using random forests and ferns," in *2007 IEEE 11th International Conference on Computer Vision*, Oct 2007, pp. 1–8.

[7] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436 EP –, 05 2015. [Online]. Available: https://doi.org/10.1038/nature14539

[8] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *CoRR*, vol. abs/1502.03167, 2015. [Online]. Available: http://arxiv.org/abs/1502.03167

[9] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *CoRR*, vol. abs/1512.03385, 2015. [Online]. Available: http://arxiv.org/abs/1512.03385

[10] A. Radford, L. Metz, and S. Chintala, "Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks," *arXiv e-prints*, 2015. [Online]. Available: https://arxiv.org/abs/1511.06434v1

[11] TensorFlow, "Deep Convolutional Generative Adversarial Network." [Online]. Available: https://www.tensorflow.org/beta/tutorials/generative/dcgan

[12] A. Odena, V. Dumoulin, and C. Olah, "Deconvolution and checkerboard artifacts," *Distill*, 2016. [Online]. Available: http://distill.pub/2016/deconv-checkerboard

[13] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," in *2nd International Conference on Learning Representations*, 2014. [Online]. Available: http://arxiv.org/abs/1312.6114

[14] C. Doersch, "Tutorial on variational autoencoders," *arXiv preprint*, 2016. [Online]. Available: https://arxiv.org/abs/1606.05908

[15] TensorFlow, "Convolutional Variational Autoencoder." [Online]. Available: https://www.tensorflow.org/beta/tutorials/generative/cvae

[16] X. Mao, Q. Li, H. Xie, R. Y. K. Lau, and Z. Wang, "Multi-class generative adversarial networks with the L2 loss function," *CoRR*, vol. abs/1611.04076, 2016. [Online]. Available: http://arxiv.org/abs/1611.04076

ETH

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Declaration of originality

The signed declaration of originality is a component of every semester paper, Bachelor's thesis, Master's thesis and any other degree paper undertaken during the course of studies, including the respective electronic versions.

Lecturers may also require a declaration of originality for other written papers compiled for their courses.

I hereby confirm that I am the sole author of the written work here enclosed and that I have compiled it in my own words. Parts excepted are corrections of form and content by the supervisor.

Title of work (in block letters):

Deep Generative Models for High-Resolution Cosmology Images

Authored by (in block letters):
For papers written by groups the names of all authors are required.

Name(s):	**First name(s):**
Schnabel	Till Nikolaus

With my signature I confirm that

- I have committed none of the forms of plagiarism described in the 'Citation etiquette' information sheet.
- I have documented all methods, data and processes truthfully.
- I have not manipulated any data.
- I have mentioned all persons who were significant facilitators of the work.

I am aware that the work may be screened electronically for plagiarism.

Place, date	**Signature(s)**
July 4th, 2019, Bern	Till Schnabel

For papers written by groups the names of all authors are required. Their signatures collectively guarantee the entire content of the written paper.

ETH

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Declaration of originality

The signed declaration of originality is a component of every semester paper, Bachelor's thesis, Master's thesis and any other degree paper undertaken during the course of studies, including the respective electronic versions.

Lecturers may also require a declaration of originality for other written papers compiled for their courses.

I hereby confirm that I am the sole author of the written work here enclosed and that I have compiled it in my own words. Parts excepted are corrections of form and content by the supervisor.

Title of work (in block letters):

Deep Generative Models for High-Resolution Cosmology Images

Authored by (in block letters):
For papers written by groups the names of all authors are required.

Name(s):	**First name(s):**
Kellenberger	Sven

With my signature I confirm that
- I have committed none of the forms of plagiarism described in the 'Citation etiquette' information sheet.
- I have documented all methods, data and processes truthfully.
- I have not manipulated any data.
- I have mentioned all persons who were significant facilitators of the work.

I am aware that the work may be screened electronically for plagiarism.

Place, date
Zurich, July 4th, 2019

Signature(s)
Kellenberger

For papers written by groups the names of all authors are required. Their signatures collectively guarantee the entire content of the written paper.

ETH

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Declaration of originality

The signed declaration of originality is a component of every semester paper, Bachelor's thesis, Master's thesis and any other degree paper undertaken during the course of studies, including the respective electronic versions.

Lecturers may also require a declaration of originality for other written papers compiled for their courses.

I hereby confirm that I am the sole author of the written work here enclosed and that I have compiled it in my own words. Parts excepted are corrections of form and content by the supervisor.

Title of work (in block letters):

Deep Generative Models for High-Resolution Cosmology Images

Authored by (in block letters):
For papers written by groups the names of all authors are required.

Name(s):	**First name(s):**
Woon	Michelle

With my signature I confirm that
- I have committed none of the forms of plagiarism described in the 'Citation etiquette' information sheet.
- I have documented all methods, data and processes truthfully.
- I have not manipulated any data.
- I have mentioned all persons who were significant facilitators of the work.

I am aware that the work may be screened electronically for plagiarism.

Place, date	**Signature(s)**
Heerbrugg, 04.07.2019	

For papers written by groups the names of all authors are required. Their signatures collectively guarantee the entire content of the written paper.

ETH

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Declaration of originality

The signed declaration of originality is a component of every semester paper, Bachelor's thesis, Master's thesis and any other degree paper undertaken during the course of studies, including the respective electronic versions.

Lecturers may also require a declaration of originality for other written papers compiled for their courses.

I hereby confirm that I am the sole author of the written work here enclosed and that I have compiled it in my own words. Parts excepted are corrections of form and content by the supervisor.

Title of work (in block letters):

DEEP GENERATIVE MODELS FOR HIGH-RESOLUTION COSMOLOGY IMAGES

Authored by (in block letters):

For papers written by groups the names of all authors are required.

Name(s):	**First name(s):**
Pfammatter	Hannes

With my signature I confirm that

- I have committed none of the forms of plagiarism described in the 'Citation etiquette' information sheet.
- I have documented all methods, data and processes truthfully.
- I have not manipulated any data.
- I have mentioned all persons who were significant facilitators of the work.

I am aware that the work may be screened electronically for plagiarism.

Place, date	**Signature(s)**
Naters, 4.7.2019	Hannes Pfammatter

For papers written by groups the names of all authors are required. Their signatures collectively guarantee the entire content of the written paper.