Emulation of cosmological mass maps with conditional generative adversarial networks

Nathanaël Perraudin
Swiss Data Science Center
ETH Zürich
nathanael.perraudin@sdsc.ethz.ch

Sandro Marcon
ETH Zürich
8006 Zürich, Switzerland

Aurelien Lucchi
Department of Computer Science
ETH Zürich
aurelien.lucchi@inf.ethz.ch

Tomasz Kacprzak
Institute for Particle Physics and Astrophysics
ETH Zürich
tomaszk@phys.ethz.ch

Abstract

Mass maps created using weak gravitational lensing techniques play a crucial role in understanding the evolution of structures in the universe and our ability to constrain cosmological models. The mass maps are based on computationally expensive N-body simulations, which can create a computational bottleneck for data analysis. Simulation-based emulators of observables are starting to play an increasingly important role in cosmology, as the analytical predictions are expected to reach their precision limits for upcoming experiments [12, 10]. Modern deep generative models, such as Generative Adversarial Networks (GANs), have demonstrated their potential to significantly reduce the computational cost of generating such simulations and generate the observable mass maps directly [15, 22, 16]. Until now, most GAN approaches produce simulations for a fixed value of the cosmological parameters, which limits their practical applicability. We instead propose a new conditional model that is able to generate simulations for arbitrary cosmological parameters spanned by the space of simulations. Our results show that unseen cosmologies can be generated with high statistical accuracy and visual quality. This contribution is a step towards emulating weak lensing observables at the map level, as opposed to the summary statistic level.

1 Introduction

Recent advances in the field of deep learning have triggered a lot of interest for their applications in cosmology. In particular, some recent works [15, 22, 16] have demonstrated the potential of Generative Adversarial Networks (GAN) [7] to efficiently produce N-body simulations, which are at the core of many experimental studies in cosmology.

N-body simulations are typically produced by simulating the evolution of the universe from soon after the big bang, where the mass distribution was approximately a Gaussian random field, to today, where, under the action of gravity, it has become highly non-Gaussian. The result of an N-body simulation consists of a 3D volume where the positions of particles represent the density of matter in specific regions. This 3-dimensional representation can then be projected in 2 dimensions by integrating the mass along the line of sight with a lensing kernel. The resulting images are called *sky convergence maps*. These maps can be compared with real observations with the purpose of estimating the cosmological parameters and testing alternatives to the standard cosmological model. Indeed, a single large N-body simulation can take from a few hours to several weeks on a supercomputer

Second Workshop on Machine Learning and the Physical Sciences (NeurIPS 2019), Vancouver, Canada.

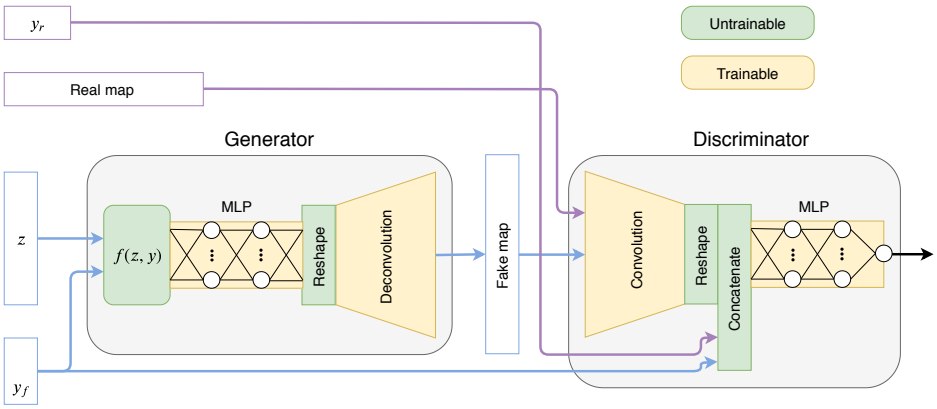

Figure 1: Sketch of the proposed model. z is the latent variable and y the parameter vector.

[2, 20, 27, 26]. Simulation-based emulators of cosmological observables, such as the power spectrum, are starting to play an increasingly important role in cosmology [12, 10].

The aforementioned contributions [15, 22, 16] attempt to address one of the main issues of N-body simulations: their computational cost. A major practical drawback of the solution proposed in these contributions is that they can only produce samples for a single set of cosmological parameters. We address this problem by building a conditional model that generates the convergence map dependent on the values of two cosmological parameters: Ω_m, which controls the matter density as a fraction of total density, and σ_8, which normalises the matter power spectrum. After training, the conditional model can then interpolate to unseen values of σ_8 and Ω_m by varying the distribution of the input latent variable.

In this work we use the data generated by [5]. We build a sky convergence map dataset made of 57 different cosmologies divided into a training set and a test set. We evaluate our GAN using both cosmological metrics, namely the power spectral density (PSD), mass histogram, and peak histogram, and generative modeling metrics (Frechet Inception Distance (FID) [11] and Multi-Scale Structural Similarity (MS-SSIM) [29]). Our experiments show that our model is able to produce samples with very high statistical accuracy on the test set.

2 Conditional generative adversarial networks

A GAN consists of two neural networks, D and G, competing against each other in a zero-sum game. The task of the *discriminator* D is to distinguish real (training) data from fake (generated) data. Meanwhile, the *generator* G produces samples with the goal of deceiving the discriminator into believing that the generated data is real. Both networks are trained simultaneously and if the optimization process is carried out successfully, the generator will learn to produce the data distribution [7]. Learning the optimal parameters of the discriminator and generator networks can be formulated as optimizing a min-max objective. Optimizing a GAN is a challenging due to the fact that it consists of two networks competing against each other. In practice, one often observes unstable training behaviors which can be mitigated by relying on various types of regularization methods [23, 8]. In this paper, we rely on Wasserstein GANs [1] with the regularization approach suggested in [8].

The model we use conditions both the generator and the discriminator on a given random variable y, yielding the following objective function,

$$\min_G \max_D \mathop{\mathbb{E}}_{(\boldsymbol{x},\boldsymbol{y})\sim\mathbb{P}_r} [D(\boldsymbol{x},\boldsymbol{y})] - \mathop{\mathbb{E}}_{\boldsymbol{z}\sim\mathbb{P}_z,\boldsymbol{y}\sim\mathbb{P}_y} [D(G(\boldsymbol{z},\boldsymbol{y}))] + \lambda \mathop{\mathbb{E}}_{(\boldsymbol{x},\boldsymbol{y})\sim\mathbb{P}_r\cup\mathbb{P}_g} [(\|\nabla_{\boldsymbol{x}} D(\boldsymbol{x},\boldsymbol{y})\|_2 - 1)^2, \quad (1)$$

where $\mathbb{P}_r$ and $\mathbb{P}_z$ are the data and latent variable distributions. $\lambda \geq 0$ is the penalty coefficient of the regularization term that ensures that the norm gradient of the discriminator is close to 1. This ensures that the discriminator is 1-Lipschitz, which is a requirement for optimizing the Wasserstein distance [1, 8]. The prior distribution of the latent variable, e.g., a uniform or a Gaussian distribution, defines implicitly the generator distribution $\mathbb{P}_g$ by $(\boldsymbol{x},\boldsymbol{y}) = G(\boldsymbol{z},\boldsymbol{y}), \boldsymbol{z} \sim \mathbb{P}_z, \boldsymbol{y} \sim \mathbb{P}_y$.

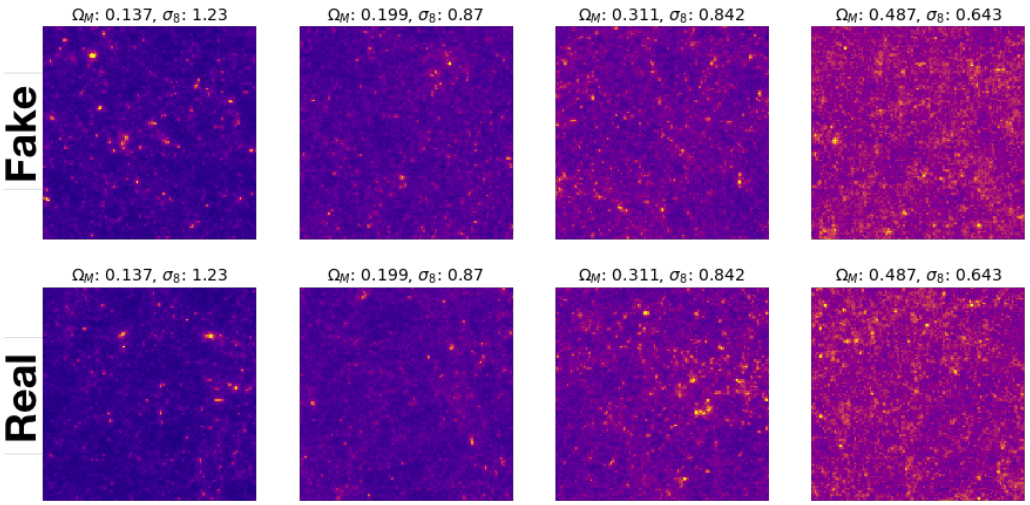

Figure 2: GAN generated images and real images for four combinations of cosmological parameters.

Practically, there exist many techniques and architectures to condition the generator and the discriminator [6, 21, 18, 17, 14]. While all the architectures in these contributions are conditioned on discrete parameters, most of them can be trivially adapted to continuous parameters. Nevertheless, here, we utilize a different design that works specifically for continuous parameters. For simplicity we describe the case of a single parameter, but our technique was implemented for the case of two parameters. Our idea is to adapt the distribution of the latent vector according to the conditioning parameters using the function $\hat{z} = f(z, y)$. Specifically, the function f simply rescales the norm of the latent vector according to the parameter y. Given the range $y \in [a, b]$, f reads:

$$\hat{z} = f(z, y) = \left(l_0 + \frac{l_1 - l_0}{b - a}(y - a) \right) \|z\|^{-1} z. \tag{2}$$

Using this function the length of z is thus mapped to the interval $[l_0, l_1]$. In our case, we used $l_0 = 0.1\sqrt{n}$ and $l_1 = \sqrt{n}$, where n is the size of the latent vector. For the discriminator, the parameters are concatenated directly after the convolutional layers as in [21]. The relation between the features extracted from the convolutional layers and the parameters might in general be non-local. We therefore increase the complexity of the mapping functions of the discriminator and generator by adding some linear layers (as in a multi-layer perceptron) at the end of each network. The proposed model is sketched in Figure 1 and the complete architecture is described in more details in Table 1.

3 Sky convergence maps dataset

The data used in this work is the non-tomographic training and testing set as in [5], without noise and intrinsic alignments. The simulation grid consisted of 57 different cosmologies assuming a flat ΛCDM universe. Each of these 57 configurations was run with different values of Ω_m and σ_8, resulting in the parameter grid shown in Figure 3. The projected matter distribution was pixelised into images of size 128 px $\times$ 128 px, which correspond to 5 deg $\times$ 5 deg of the sky. Eventually, the resulting dataset consists of 57 sets of $12'000$ sky convergence maps for a total of $684'000$ samples. The dataset was split into a training and test set in the following way: 11 cosmologies were selected for the test set and the remaining 46 cosmologies were assigned to the training set, as depicted in Figure 3 (left). This split was used to ensure that the model could interpolate to unseen cosmologies. At evaluation time, we use the cosmologies from the test set to validate the interpolation ability of our network. Eventually, the training set was augmented by random rotations and flips.

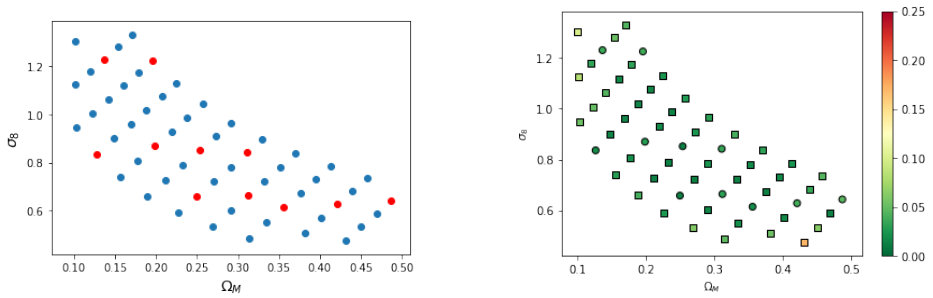

Figure 3: Left: The grid corresponding to the different parameters of the sky convergence maps. The red cosmologies compose the test set and the blue cosmologies the training set. Right: Average fractional difference between the PSD curves (averaged over 2000 samples) for different parameters.

4 Experiments

Training. We used RMSProp as an optimizer with an initial learning rate of 10^{-5} and a batch size of 64. The discriminator was updated 5 times more than the generator. The gradient penalty was set to 10 and the negative slope of the LeakyRelu $\alpha = 0.2$. It took a week to train the model for 40 epochs on a GeForce GTX 1080 GPU. Similar to [13, 21, 14, 17], we use batches composed of samples from different parameter sets. Note that the batches were composed of samples from different cosmologies from the *training* set.

Visual evaluation. Figure 2 shows images generated by the conditional GAN and as well as original ones. It is nearly impossible to distinguish them for the human eye. Furthermore, the image structure evolves similarly with respect of the cosmological parameters change. As predicted by the theory, increasing Ω_m results in convergence maps with additional mass and increasing σ_8 in images with higher variance in pixel intensities. In Figure 4 the same latent variable z is used to generate different cosmologies. The smooth transition from low to high mass density hints that the latent variable control the overall mass distribution and the conditioning parameter its two cosmological properties σ_8, Ω_m.

Cosmological statistics. As cosmologists mainly rely on statistics in their analyses, a good statistical agreement is of prior necessity. Here we focus on 3 cosmological statistics: the power spectral density (PSD), the mass histogram and the peak histogram. Details about these statistics are given in Appendix A. These statistics are computed using 2000 real and fake samples for every pair of parameters of the *test* set. As illustrated in Figure 5 of the Appendix, the PSD curves overlap almost perfectly for all the cosmologies *lying inside the parameter grid used for training*. The average absolute fractional difference (defined in Appendix A) for these cosmologies is $2.71\% \pm 0.94\%$. Regarding the peak and mass density histograms, the same considerations hold.

Variance in the produced images. To measure the variance in the produced images, we use the Multi-Scale Structural Similarity (MS-SSIM) described in Appendix A. The absolute difference between the MS-SSIM scores of real and fake samples is 0.0018 globally and 0.0012 ± 0.0009 per cosmology, meaning that generated images are as varied as the original ones. This is evidence that the model did not suffer from mode collapse.

Frechet Inception Distance. The Fréchet Inception Distance (FID) [11] has become a standard metric for GANs. As the Inception network was trained with the ImageNet dataset and not sky convergence maps, we needed to re-train a CNN (we performed regression from image to comological parameters on the training set similar to [25, 4, 9]) to build a Cosmological FID (see Appendix A). In our case the FID averaged over 2000 samples for each cosmology of the test set is 0.011 ± 0.08. As there exist no other contribution to compare with, we propose another indicative experiment. In Figure 6 of the Appendix, we show the output of 2000 real and fake samples for each parameter sets. We expect that a well working conditional GAN should generate samples with similar output distribution to the one of the real samples. We observe a good agreement between the two distributions.

Error within the parameter space. To analyze the performance of the model in different regions of the parameter space in more detail, we show the average fractional difference between the PSD curves in Figure 3 (right). Our model produces more accurate results close to the center of the grid, where the average fractional difference stays below 2.5%. The performance decreases as the parameters move away from the central region but, overall, it almost always stay below 10%. We also note that there is no important difference in performance between training and validation sets. This shows the excellent interpolation capacity of our model.

5 Conclusion

We proposed a new conditional GAN model for continuous parameters where conditioning is done within the latent space. We demonstrated the ability of this model to generate sky convergence maps when conditioning on cosmological parameters Ω_m and σ_8. Our model is able to produce samples from the test set with good statistical accuracy, which demonstrates its generalization abilities. Moreover, the model is able to capture the variability in the conditioned dataset. This offers good prospects for GAN-based conditional models to be used as emulators of cosmology-dependent mass maps. One significant advantage of such model in cosmology is its ability to produce maps at a speed that is several orders of magnitude faster than a traditional, full N-body simulator. Moreover, the generation time is comparable to standard emulators of summary statistics, such as the power spectra [12, 10]. The model and the code will be released with the article version of this contribution.

Acknowledgments

This work was supported by a grant from the Swiss Data Science Center (SDCS) under project *DLOC: Deep Learning for Observational Cosmology* and grant number 200021_169130 from the Swiss National Science Foundation (SNSF). We thank Alexandre Refregier, Thomas Hofmann, and Fernando Perez-Cruz for advice and helpful discussions. We thank the Cosmology Research Group of ETHZ and particularly Janis Fluri for giving us access to the dataset. Finally, we thank the two anonymous reviewers who provided extensive feedback that greatly improved the quality of this paper.

References

[1] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *International conference on machine learning*, pages 214–223, 2017.

[2] Euclid Collaboration, Mischa Knabenhans, Joachim Stadel, Stefano Marelli, Doug Potter, Romain Teyssier, Laurent Legrand, Aurel Schneider, Bruno Sudret, Linda Blot, et al. Euclid preparation: Ii. the euclidemulator–a tool to compute the cosmology dependence of the nonlinear matter power spectrum. *Monthly Notices of the Royal Astronomical Society*, 484(4):5509–5529, 2019.

[3] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[4] Janis Fluri, Tomasz Kacprzak, Aurelien Lucchi, Alexandre Refregier, Adam Amara, and Thomas Hofmann. Cosmological constraints from noisy convergence maps through deep learning. *arXiv preprint arXiv:1807.08732*, 2018.

[5] Janis Fluri, Tomasz Kacprzak, Aurelien Lucchi, Alexandre Refregier, Adam Amara, Thomas Hofmann, and Aurel Schneider. Cosmological constraints with deep learning from KiDS-450 weak lensing maps. *arXiv e-prints*, page arXiv:1906.03156, Jun 2019.

[6] Jon Gauthier. Conditional generative adversarial nets for convolutional face generation. *Class Project for Stanford CS231N: Convolutional Neural Networks for Visual Recognition, Winter semester*, 2014(5):2, 2014.

[7] 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*, pages 2672–2680, 2014.

[8] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In *Advances in Neural Information Processing Systems*, pages 5767–5777, 2017.

[9] Arushi Gupta, José Manuel Zorrilla Matilla, Daniel Hsu, and Zoltán Haiman. Non-gaussian information from weak lensing data via deep learning. *Physical Review D*, 97(10):103515, 2018.

[10] Katrin Heitmann, Derek Bingham, Earl Lawrence, Steven Bergner, Salman Habib, David Higdon, Adrian Pope, Rahul Biswas, Hal Finkel, Nicholas Frontiere, and Suman Bhattacharya. The Mira-Titan Universe: Precision Predictions for Dark Energy Surveys. *The Astrophysical Journal*, 820(2):108, Apr 2016.

[11] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems*, pages 6626–6637, 2017.

[12] Mischa Knabenhans, Joachim Stadel, Stefano Marelli, Doug Potter, Romain Teyssier, Laurent Legrand, Aurel Schneider, Bruno Sudret, Linda Blot, Saeeda Awan, Carlo Burigana, Carla Sofia Carvalho, Hannu Kurki-Suonio, Gabriele Sirri, and Euclid Collaboration. Euclid preparation: II. The EUCLIDEMULATOR - a tool to compute the cosmology dependence of the nonlinear matter power spectrum. *Monthly Notices of the Royal Astronomical Society*, 484(4):5509–5529, Apr 2019.

[13] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

[14] Takeru Miyato and Masanori Koyama. cGANs with projection discriminator. In *International Conference on Learning Representations*, 2018.

[15] Mustafa Mustafa, Deborah Bard, Wahid Bhimji, Rami Al-Rfou, and Zarija Lukić. Creating virtual universes using generative adversarial networks. *arXiv preprint arXiv:1706.02390*, 2017.

[16] Perraudin Nathanaël, Srivastava Ankit, Tomasz Kacprzak, Aurelien Lucchi, Thomas Hofmann, and Alexandre Réfrégier. Cosmological n-body simulations: a challenge for scalable generative models. *arXiv preprint arXiv:1908.05519*, 2019.

[17] Augustus Odena, Christopher Olah, and Jonathon Shlens. Conditional image synthesis with auxiliary classifier gans. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 2642–2651. JMLR. org, 2017.

[18] Guim Perarnau, Joost Van De Weijer, Bogdan Raducanu, and Jose M Álvarez. Invertible conditional gans for image editing. *arXiv preprint arXiv:1611.06355*, 2016.

[19] A. Petri. Mocking the weak lensing universe: The LensTools Python computing package. *Astronomy and Computing*, 17:73–79, October 2016.

[20] Douglas Potter, Joachim Stadel, and Romain Teyssier. PKDGRAV3: beyond trillion particle cosmological simulations for the next era of galaxy surveys. *Computational Astrophysics and Cosmology*, 4(1):2, May 2017.

[21] Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. Generative adversarial text to image synthesis. *arXiv preprint arXiv:1605.05396*, 2016.

[22] Andres C Rodriguez, Tomasz Kacprzak, Aurelien Lucchi, Adam Amara, Raphael Sgier, Janis Fluri, Thomas Hofmann, and Alexandre Réfrégier. Fast cosmic web simulations with generative adversarial networks. *arXiv preprint arXiv:1801.09070*, 2018.

[23] Kevin Roth, Aurelien Lucchi, Sebastian Nowozin, and Thomas Hofmann. Stabilizing training of generative adversarial networks through regularization. In *Advances in neural information processing systems*, pages 2018–2028, 2017.

[24] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *Advances in Neural Information Processing Systems*, pages 2234–2242, 2016.

[25] Jorit Schmelzle, Aurelien Lucchi, Tomasz Kacprzak, Adam Amara, Raphael Sgier, Alexandre Réfrégier, and Thomas Hofmann. Cosmological model discrimination with deep learning. *arXiv preprint arXiv:1707.05167*, 2017.

[26] RJ Sgier, Alexandre Réfrégier, Adam Amara, and Andrina Nicola. Fast generation of covariance matrices for weak lensing. *Journal of Cosmology and Astroparticle Physics*, 2019(01):044, 2019.

[27] V. Springel, S. D. M. White, A. Jenkins, C. S. Frenk, N. Yoshida, L. Gao, J. Navarro, R. Thacker, D. Croton, J. Helly, J. A. Peacock, S. Cole, P. Thomas, H. Couchman, A. Evrard, J. Colberg, and F. Pearce. Simulations of the formation, evolution and clustering of galaxies and quasars. *nature*, 435:629–636, June 2005.

[28] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

[29] Zhou Wang, Eero P Simoncelli, and Alan C Bovik. Multiscale structural similarity for image quality assessment. In *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*, volume 2, pages 1398–1402. Ieee, 2003.

Appendix

A Evaluation metrics

One important challenge of GANs and generative models in general is the lack of standard metrics to evaluate performances of a given model. For natural images visual quality has often been quantified using human inspection. Problematically, this technique is very subjective and impractical for our data, as samples may look as the human eye is not trained to perceive small statistical changes in a semi-random pattern. Fortunately, the field of cosmology relies on specific tractable statistics that we can borrow to evaluate the accuracy of the generated samples.

Power spectral density In the case of convergence maps, the most important diagnostic measure is the power spectral density (PSD). It corresponds to the measure of a signal energy with respect to frequency. In this work, we rely on LensTools [19] to compute this statistic.

Mass density histogram Another measure used to evaluate the goodness of the generated images is the mass density histogram. This is simply the normalized histogram of the pixel intensities of the images. We refer to it as the mass histogram, since for the sky convergence maps pixel intensities correspond to the mass integrated in one line of sight.

Peak histogram Similarly, we use the peak histogram, i.e. the histogram of the intensities of the maxima (peaks). The computation of this measure is performed in two steps. First the peaks are extracted by searching for all pixels greater than their 5×5 patch neighborhood, i.e. their 24 neighbours. In the second step, the histogram of the extracted peaks is computed.

Relative and fractional difference As a single cosmological statistic such as the PSD or the mass histogram is composed of value with very different scale, computing a summary number requires to relatively evaluate the difference between the real and the fake statistic. Given n real and n fake samples x_i^r, x_i^f for $i = 1 \dots n$ and a statistic s, the absolute fractional difference is then defined using the means $m_{sr} = \frac{1}{n}\sum_i s(x_i^r)$ and $m_{sf} = \frac{1}{n}\sum_i s(x_i^f)$ as:

$$d_{\text{f-abs}} = \left| \frac{m_{sr} - m_{sf}}{m_{sr}} \right| \tag{3}$$

Furthermore, using the standard deviation $\text{std}_{sr} = \sqrt{\frac{1}{n}\sum_i (s(x_i^r) - m_{sr})^2}$ we also define the absolute relative difference as:

$$d_{\text{f-rel}} = \left| \frac{m_{sr} - m_{sf}}{\text{std}_{sr}} \right| \tag{4}$$

The absolute relative difference is particularly useful when the standard deviation of a variable is greater than its mean. In this contribution we use the absolute fractional difference to analyze the agreement between curves for the PSD and the relative difference for the peak and mass density histograms.

MS-SSIM score One common problem when training GANs is that the generator produces only a small subset of the training data distribution. This is commonly referred as mode collapse. Detecting this undesirable behavior is not trivial as statistics can agree during mode collapse. Taking inspiration from [17], one solution is to leverage the Multi-Scale Structural Similarity (MS-SSIM) score from [29] to quantify this effect. This metric was first proposed by aiming to predict human perceptual similarity. Taking two images as input it returns a value between 0 and 1, where 1 means identical and 0 completely different. In our case we are not interested in the similarity between two images, but in the similarity of a set of images. To keep our computation tractable, we randomly select 100 pairs of images and average the obtained MS-SSIM score. Note that this metric does not only identify mode collapses but also indicates how different the generated images are.

Fréchet Inception Distance Recently Inception Score (IS) [24] and Fréchet Inception Distance (FID) [11] have become standard measures for GANs. The idea consists to compare statistics of the output of the inception network [28] for the ImageNet dataset [3]. This has proven to be well correlated with human score. In this work, we cannot directly use the inception network generally used to compute IS and FID as our data is very different from the ImageNet dataset. Hence, similar to [25, 4, 9], we first train a Convolution Neural Network (CNN) on a regression task using real data from the training set (see Table 2). Then, we feed it with both generated and real samples. The FID is then computed using the following formula:

$$\text{FID} = \|\mu_r - \mu_g\|^2 + \text{Tr}\left(\Sigma_r + \Sigma_g - 2(\Sigma_r\Sigma_g)^{1/2}\right), \tag{5}$$

where μ_r, μ_g, Σ_r, Σ_g are the means and variances of the activations of the second to last fully connected layer of the CNN for real and generated samples respectively. Tr is the trace operator.

B Parameters

Layer	Operation	Activation	Dimension
Generator			
$\mathbf{z}$			$b \times 128$
h_0	linear	Relu	$b \times 256$
h_1	linear	Relu	$b \times 512$
h_2	linear	Relu	$b \times 8 \times 8 \times 512$
h_3	deconv	Relu	$b \times 16 \times 16 \times 256$
h_4	deconv	Relu	$b \times 32 \times 32 \times 128$
h_5	deconv	Relu	$b \times 64 \times 64 \times 64$
h_6	deconv	Relu	$b \times 128 \times 128 \times 32$
h_7	deconv	Relu	$b \times 128 \times 128 \times 1$
Discriminator			
$\mathbf{X}$			$b \times 128 \times 128$
h_0	conv	LeakyRelu	$b \times 128 \times 128 \times 32$
h_1	conv	LeakyRelu	$b \times 64 \times 64 \times 64$
h_2	conv	LeakyRelu	$b \times 32 \times 32 \times 128$
h_3	conv	LeakyRelu	$b \times 16 \times 16 \times 256$
h_4	conv	LeakyRelu	$b \times 8 \times 8 \times 512$
h_5	reshape	concatenate	$b \times 32770$
h_6	linear	LeakyRelu	$b \times 512$
h_7	linear	LeakyRelu	$b \times 256$
h_8	linear	LeakyRelu	$b \times 128$
h_9	linear	LeakyRelu	$b \times 1$

Table 1: Conditional GAN architecture. Here b is the batch size and h_5 in the discriminator is a layer that reshapes the tensor to a vector and then concatenates the conditioning parameters to it.

Layer	Operation	Activation	Dimension
$\mathbf{X}$			$b \times 128 \times 128$
h_0	conv	LeakyRelu	$b \times 128 \times 128 \times 32$
h_1	conv	LeakyRelu	$b \times 64 \times 64 \times 64$
h_2	conv	LeakyRelu	$b \times 32 \times 32 \times 128$
h_3	conv	LeakyRelu	$b \times 16 \times 16 \times 256$
h_4	conv	LeakyRelu	$b \times 8 \times 8 \times 512$
h_5	linear	LeakyRelu	$b \times 512$
h_5	linear	LeakyRelu	$b \times 256$
h_5	linear	LeakyRelu	$b \times 128$
h_5	linear	linear	$b \times 2$

Table 2: Architecture of the regressor. Here b is the batch size.

C Extra figures

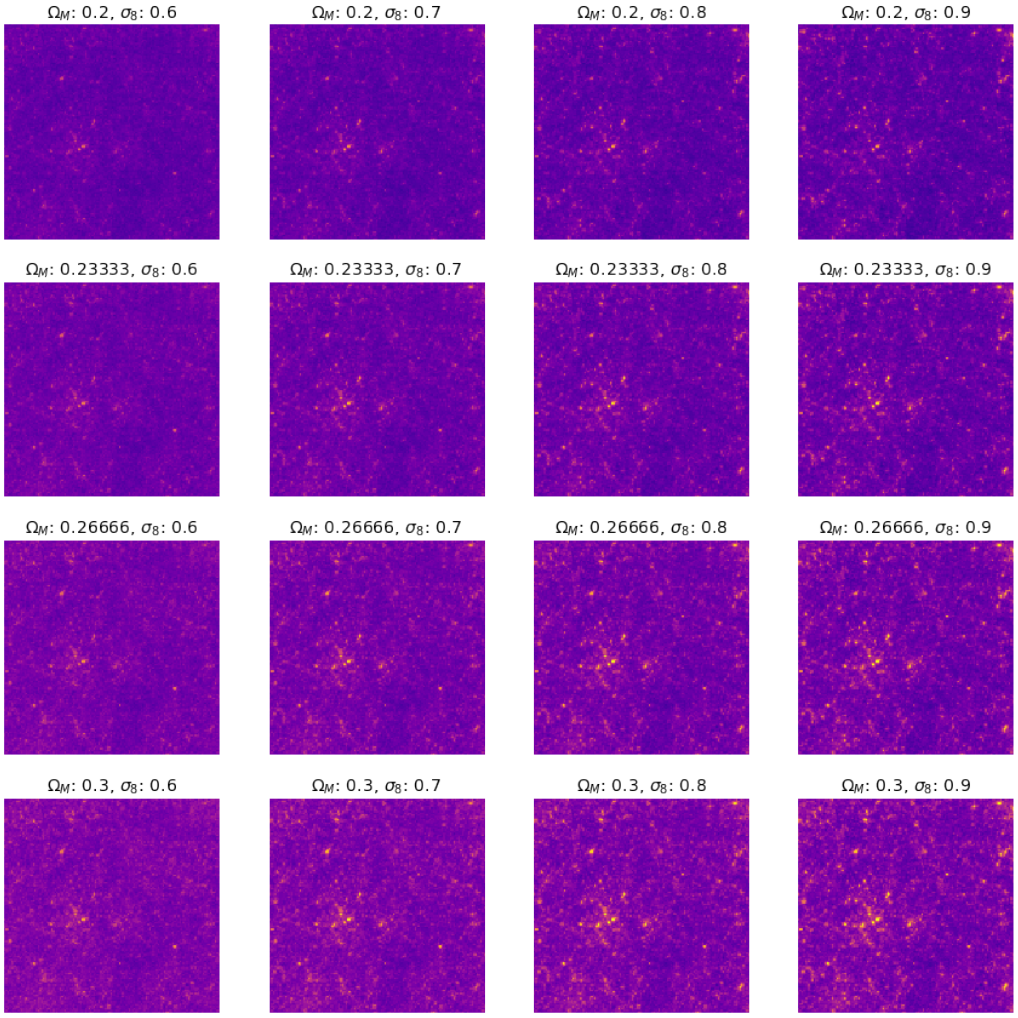

Figure 4: Images generated with the same random seed but with different input parameters.

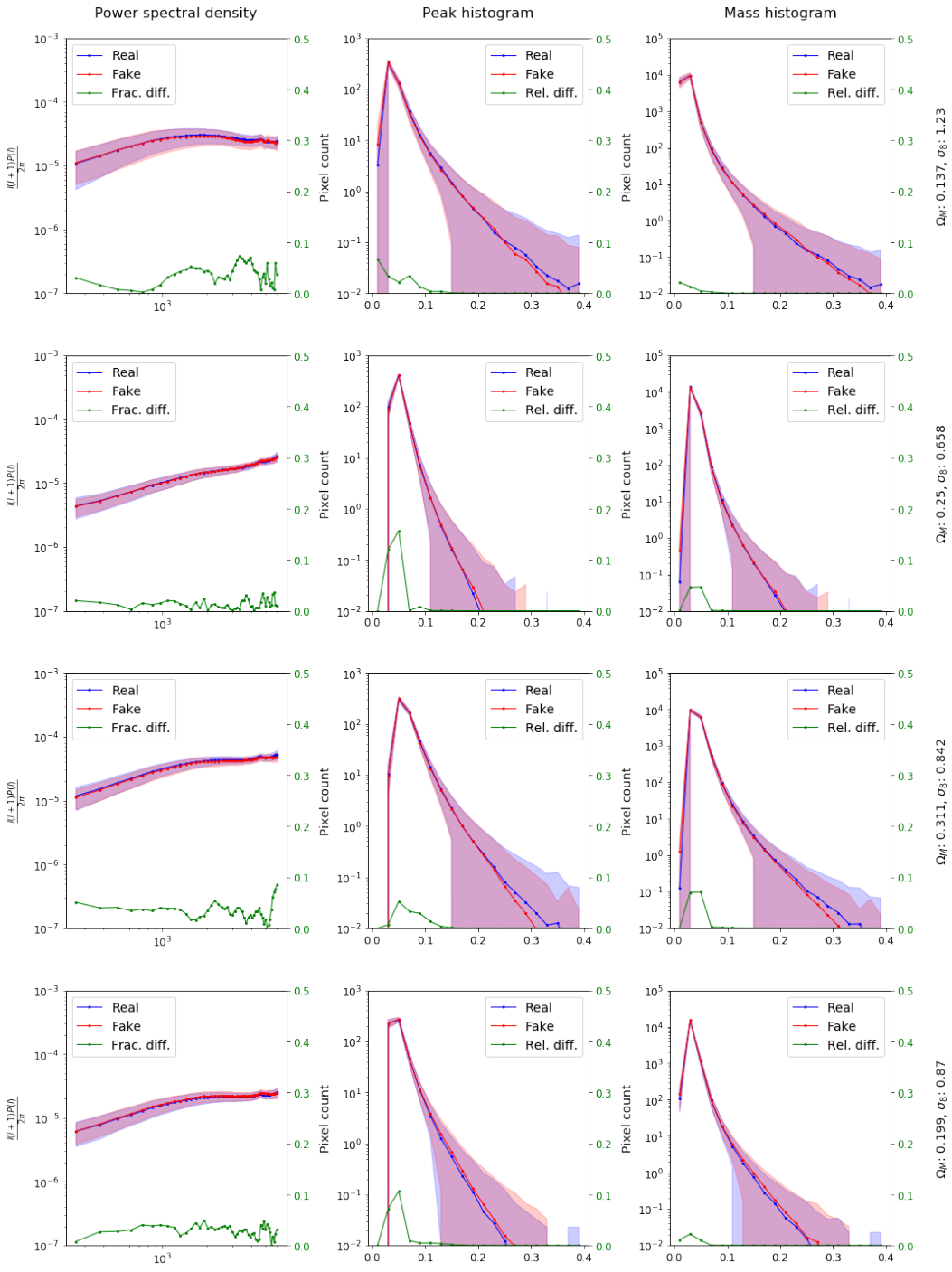

Figure 5: Comparison of PSD, peak histogram and mass density histogram between generated (red) and real samples (blue). Rows represent different cosmologies. The shading corresponds to the standard deviation and the green curve to the absolute fractional or relative difference. Note that all the parameter sets lie inside the training grid.

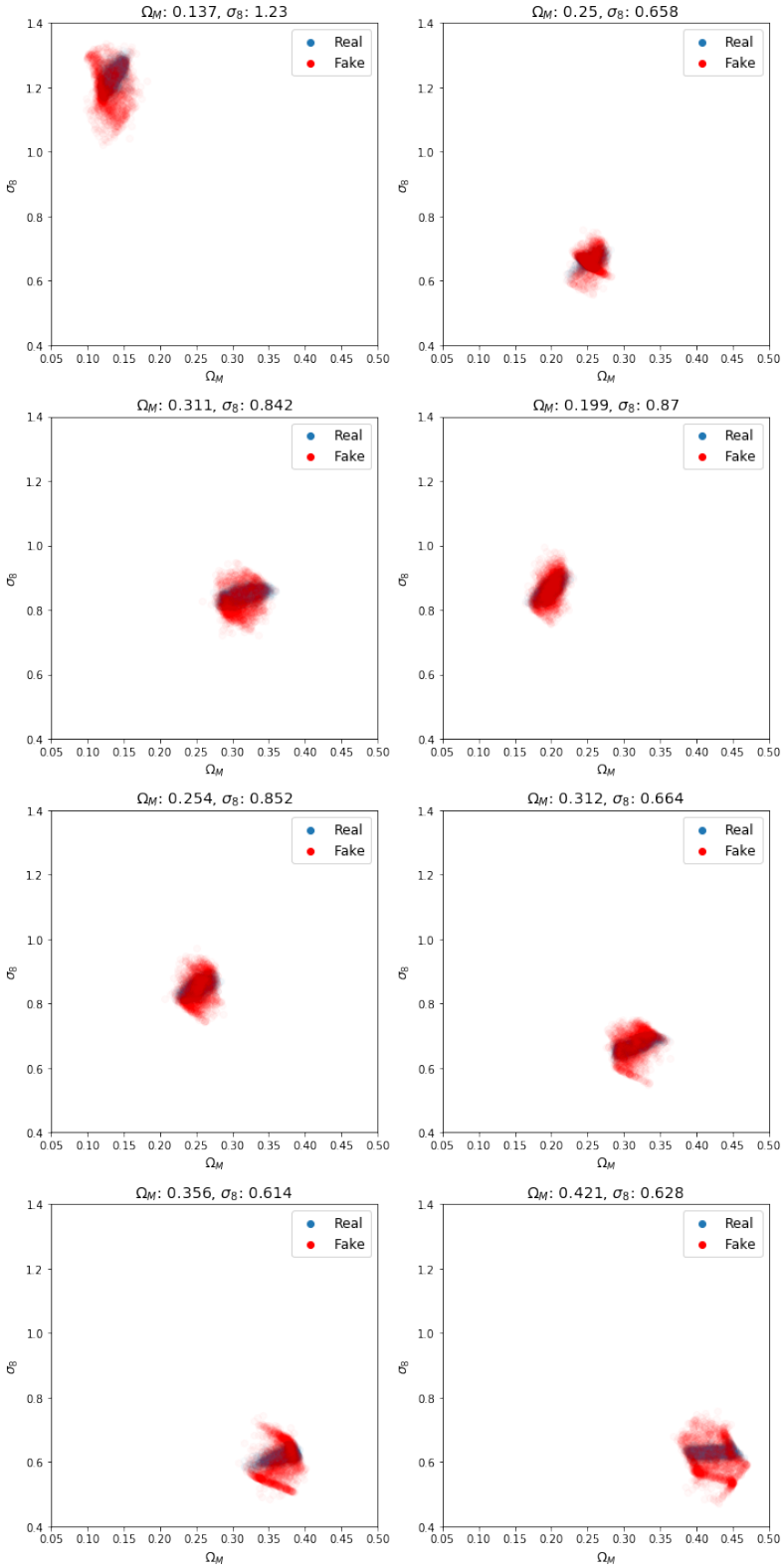

Figure 6: Predicted parameters from the CNN described in Table 2. Blue and red points represent real and generated samples respectively.