

Creating Virtual Universes Using Generative Adversarial Networks

Mustafa Mustafa^{*1}, Deborah Bard¹, Wahid Bhimji¹, Zarija Lukić¹, Rami Al-Rfou², and Jan Kratochvil³

¹Lawrence Berkeley National Laboratory, Berkeley, CA 94720

²Google Research, Mountain View, CA 94043

³University of KwaZulu-Natal, Westville, Durban 4000, South Africa

Abstract

Inferring model parameters from experimental data is a grand challenge in many sciences, including cosmology. This often relies critically on high fidelity numerical simulations, which are prohibitively computationally expensive. The application of deep learning techniques to generative modeling is renewing interest in using high dimensional density estimators as computationally inexpensive emulators of fully-fledged simulations. These generative models have the potential to make a dramatic shift in the field of scientific simulations, but for that shift to happen we need to study the performance of such generators in the precision regime needed for science applications. To this end, in this letter we apply Generative Adversarial Networks to the problem of generating cosmological weak lensing convergence maps. We show that our generator network produces maps that are described by, with high statistical confidence, the same summary statistics as the fully simulated maps.

The scientific success of the next generation of sky surveys (e.g. [1–5]) to test the current “standard model” of cosmology (Λ CDM), hinges critically on the success of underlying simulations. Answering questions in cosmology about the nature of cold dark matter, dark energy and the inflation of the early universe, requires relating observations of a large number of astrophysical objects which trace the underlying matter density field, to simulations of “virtual universes” with different cosmological parameters. Currently the creation of each virtual universe requires an extremely computationally expensive simulation on High Performance Computing resources. In order to make this inverse problem practically solvable, constructing a computationally cheap *surrogate model* or an *emulator* [6, 7] is imperative.

However, traditional approaches to emulators require the use of a summary-statistic which is to be emulated. An approach that does not require such mathematical templates of the simulation outcome would be of considerable value in the field. The ability to emulate these simulations with high fidelity, in a fraction of the computational time, would boost our ability to understand the fundamental nature of the universe. While in this letter we focus our attention on cosmology, and in particular weak lensing convergence maps, we believe that this approach is relevant to many areas of science and engineering.

Recent developments in deep generative modeling techniques open the potential to meet this need. The density estimators in these models are built out of neural networks which can serve as universal approxima-

tors [8], thus having the ability to learn the underlying distributions of data and emulate the observable without being biased by the choice of summary-statistics, as in the traditional approach to emulators.

In this letter, we study the ability of a recent variant of generative models - Generative Adversarial Networks (GANs) [9] to generate weak lensing convergence maps. The training and validation maps are produced using N-body simulations of Λ CDM cosmology. We show that maps generated by the neural network exhibit, with high statistical confidence, the same power (Fourier) spectrum of the fully-fledged simulator maps, as well as higher order non-Gaussian features, thus demonstrating that such scientific data is amenable to a GAN treatment for generation. The very high level of agreement we achieve offers promise for building emulators out of deep neural networks. We first present our results and analysis then outline the future investigations which we think are critical to build such emulators in the Discussion section.

Results

Gravitational lensing has potential to be one of the most sensitive probes of the nature of dark energy [10], and affects the shape and apparent brightness of every galaxy we observe. Convergence $\kappa(\boldsymbol{\nu})$ is the quantity that defines the brightness of an observed object as it is affected by the matter along the line of sight between that galaxy and the observer. It can be interpreted as a measure of the density of the universe observed from a particular direction. A full N-body simulation cre-

^{*}Corresponding author: mmustafa@lbl.gov

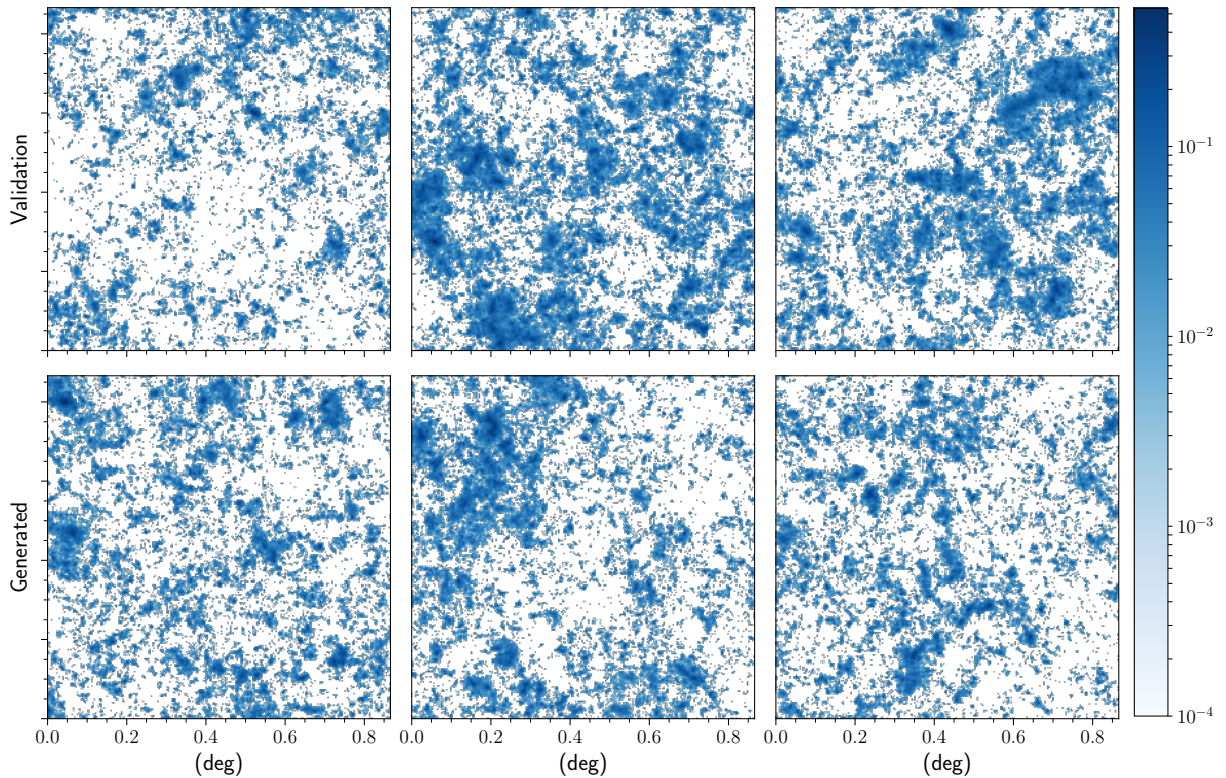


Figure 1: Weak lensing convergence maps for a Λ CDM cosmological model with $\sigma_8 = 0.798$, $w = -1.0$, $\Omega_m = 0.26$ and $\Omega_\Lambda = 0.74$. Randomly selected maps from validation dataset (top) and GAN generated examples (bottom).

ates convergence maps corresponding to many random realizations of the same cosmological model. We set out to train a GAN model on 256×256 pixels convergence maps taken from these simulations. A description of the simulations and data preparation methods is in the Methods section. Before we describe our results we first outline the objective of generative models and the GANs framework.

The central problem of generative models is the question: given a distribution of data \mathbb{P}_{data} can one devise a generator G such that the distribution of model generated data $\mathbb{P}_{model} = \mathbb{P}_{data}$? Our information about \mathbb{P}_{data} comes from the training dataset, typically an independent and identically distributed random sample x_1, x_2, \dots, x_n which is assumed to have the same distribution as \mathbb{P}_{data} . Achieving a high fidelity generation scheme amounts to the construction of a density estimator of the training data. In the GANs framework a generator function G is optimized to generate samples that are indistinguishable from training data as judged by a discriminator function D . D is optimized to discriminate between training data and generated data. In the neural network formulation of this framework the generator network G_θ parametrized by network parameters θ and discriminator network D_w parametrized by w are simultaneously optimized using gradient-descent.

Of interest to us here is the generator G_θ . Its param-

eters are optimized to map a vector \mathbf{z} sampled from a prior to the support of \mathbb{P}_{model} . The only requirement on the generator is that it is differentiable with respect to its parameters and input (except at possibly finitely many points). For the 256×256 convergence maps we study, we choose a normal prior, so:

$$\mathbf{z} \sim [\mathcal{N}_0(0, 1), \dots, \mathcal{N}_{63}(0, 1)] \quad (1)$$

$$G_\theta : \mathbf{z} \rightarrow x \in \mathbb{R}^{256 \times 256}. \quad (2)$$

The dimension of the vector \mathbf{z} needs to be commensurate with the support of the training convergence maps \mathbb{P}_{data} in $\mathbb{R}^{256 \times 256}$. Because the underlying physics of the convergence maps is translation and rotation invariant [11], we chose to construct the generator and discriminator networks mainly from convolutional layers. To allow the network to learn the proper correlations on the components of the input \mathbf{z} early on, the first layer of the generator network needs to be a fully-connected layer. A well studied architecture that meets these criteria is the Deep Convolutional Generative Adversarial Networks (DCGAN) [12]. DCGAN is a set of empirically chosen architectural guidelines and hyper-parameters which have been shown to be robust to excel at a variety of tasks. We experimented with DCGAN architectural parameters and we found that most of the hyper-parameters optimized for natu-

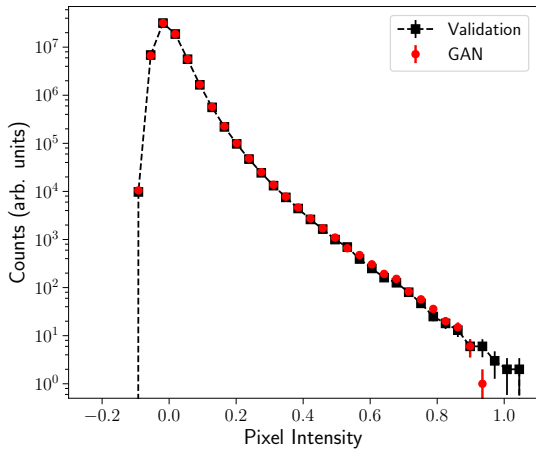


Figure 2: Pixel intensity distribution of 1000 generated maps (red circles) compared to those of 1000 validation maps (black squares). The GAN is able to emulate the distribution of intensities in the maps. The Kolmogorov-Smirnov test of similarity of these distributions gives a p-value > 0.999 .

ral images by the original authors perform well on the convergence maps, for example, changing the learning rates or the kernel sizes worsens the performance. We used DCGAN with slight modifications to meet our problem dimensions as described in the Methods section.

Figure 1 shows examples of maps from the validation and GAN generated datasets. The validation dataset has not been used in the training or tuning of the networks. By eye an expert cannot distinguish the generated maps from the full simulation ones. We now move to a quantitative comparison of these datasets.

Evaluation of generative models. Once one has a density estimator of the original data the first practical question would be to determine the goodness of the fit. Basically, *how close is \mathbb{P}_{model} to \mathbb{P}_{data} ?* This issue is critical to understanding and improving generative models formulation and training procedure and is still an active area of research [13]. Significant insights into the training dynamics of GANs from a theoretical point of view have been gained recently [14, 15]. We think that when it comes to practical applications of generative models, such as in the case of emulating scientific data, *the criterion to evaluate generative models is to study their ability to reproduce the characteristic statistics which we can measure from the original dataset.*

To this end, we calculate three statistics on the generated convergence maps: a first order statistic (pixel intensity), second order correlations and a measurement of the non-Gaussian corrections. Such measurements and several other corrections schemes are known as summary-statistics of the maps. The ability to reproduce such summary-statistics is a reliable metric to evaluate generative models from an information en-

coding point of view. To test our statistical confidence of the matching of the summary-statistics we perform bootstrapped two-tailed Kolmogorov-Smirnov (KS) test and Andersen-Darling (AD) test of the null-hypothesis that the summary-statistics in the generated maps and the validation maps have been drawn from the same continuous distributions.

Figure 2 shows a histogram of the distribution of pixel intensities of an ensemble of generated maps compared to that of a validation dataset. It is clear that the GAN generator has been able to learn the probabilities of pixel intensities in the real simulation dataset, the KS p-value is > 0.999 .

The second-order statistical moment in gravitational lensing is the shear power spectrum. This is a measure of the correlation in gravitational lensing at different distances, and characterizes the matter density fluctuations at different length scales. Assuming we have only Gaussian fluctuations $\delta(x)$ at comoving distance x , the matter density of the universe can be defined by its power spectrum P_κ :

$$\langle \tilde{\delta}(l)\tilde{\delta}^*(l') \rangle = (2\pi)^2 \delta_D(l - l') P_\kappa(l) \quad (3)$$

where δ_D is the Dirac delta function, and l is the Fourier mode [11]. The power spectrum (and its corresponding real-space measurement, the two-point correlation function) of convergence maps has been long used to constrain models of cosmology by comparing simulated maps to real data from sky surveys [16–18]. Numerically, the power spectrum is the amplitudes of the Fourier components of the map. We calculate the power spectrum at 248 Fourier modes of an ensemble of generated maps using LensTools [19], and compare them to the validation dataset. Since each map is a different random realization, comparison has to be done at the ensemble level. Figure 3(a) shows two bands representing the mean $\mu(l) \pm$ standard deviation $\sigma(l)$ of the ensemble of power spectra at each Fourier mode of the validation and a generated dataset. As is clear in the figure, the bands completely overlap with each other. To confirm that the range of the power spectrum at a given l is completely reproduced in the generated maps we look differentially at the underlying 1-D distributions. Samples of such distributions at equally spaced Fourier modes are shown in Figure 3(b). The bootstrapped KS (AD) p-values for 236 (205) Fourier modes are > 0.99 , of the remaining modes, 10 (26) are > 0.95 , all the remaining are larger than 0.8. The power spectrum is the figure of merit for evaluating the success of an emulator for reproducing the structures of the convergence map, and we have shown with statistical confidence that GANs are able to discover and reproduce such structures.

In Fig. 4 we show the correlation matrices of the power spectra shown in Fig. 3, and the difference between the correlation matrices from the validation maps and the ones produced by our GAN. We also show a comparison of the correlation matrices from

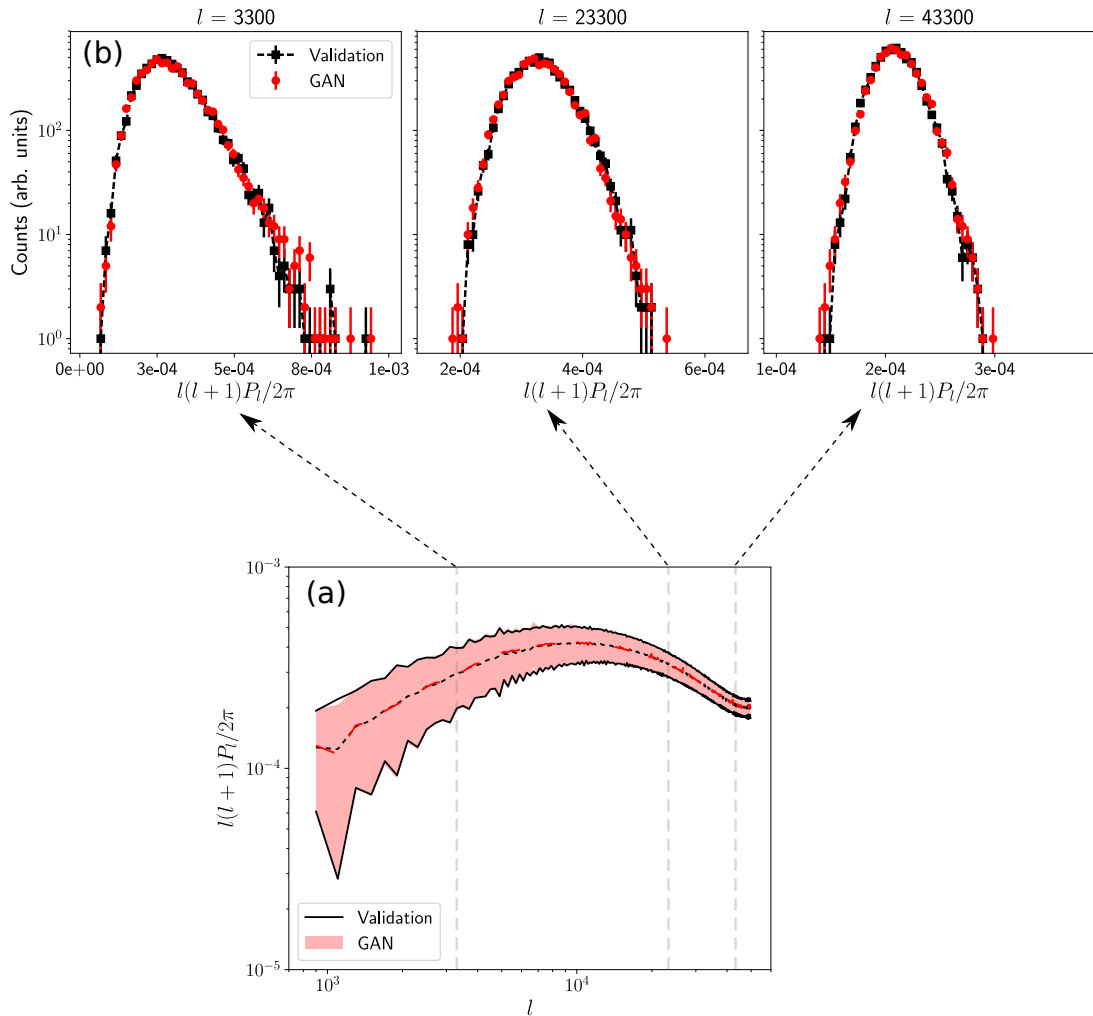


Figure 3: The power spectrum of the convergence maps, evaluated at 248 Fourier modes. We use 100 batches of generated maps (6400 in total) for this comparison. Shown in (a) are bands of the $\mu(l) \pm \sigma(l)$, the dashed lines represent the means $\mu(l)$, (b) shows the underlying distributions at 3 equidistant modes for illustration.

the validation dataset and an auxiliary dataset (where the auxiliary dataset consists of 1000 maps produced using the same simulation code and cosmological parameters, but with a different random seed, resulting in a set of convergence maps that are statistically independent from those used in our training and validation). There are differences between the validation and auxiliary correlations matrices of up to 7%. The differences between the validation and generated correlation matrices are of a similar level (up to 10%) but show a pattern of slightly higher correlation between high- and low- l modes. We find it interesting that the GAN generator assumes a slightly stronger correlation between small and large modes, thus a smoother power spectra, than in the simulations. This is due to the large uncertainty of the power spectra at small modes in the original simulations as seen in Fig. 3 (a).

The power spectrum only captures the Gaussian components of matter structures in the universe. However, gravity produces non-Gaussian structures at small scales which are not described by the two-point

correlation function [11]. There are many ways to access this non-Gaussian statistical information, including higher-order correlation functions, and topological features such as Minkowski functionals V_0 , V_1 , and V_2 [20, 21] which characterize the area, perimeter and genus of the convergence maps. We investigate the ability of the networks to discover and reproduce these non-Gaussian structures in the maps by evaluating the three Minkowski functionals. Figure 5 compares bands of $\mu \pm \sigma$ of the three Minkowski Functionals (calculated using [19]) to those in the real dataset. Each functional is evaluated at 19 thresholds. As we did with the power spectrum, we show the Minkowski functionals at different thresholds in Figure 6. The bootstrapped KS (AD) for 40 (32) threshold histograms are > 0.95 p-values 7 (6) are > 0.9 all the remaining are > 0.6 . Again, it is clear that the GANs have successfully reproduced those non-Gaussian structures in the maps.

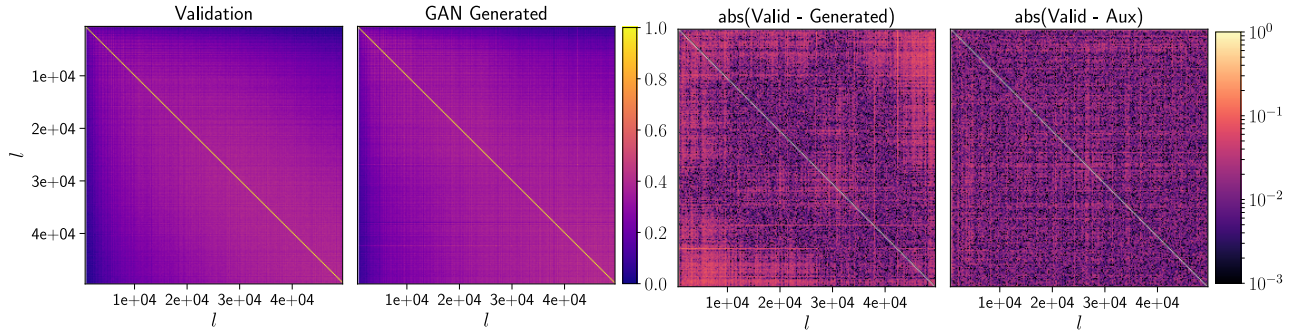


Figure 4: The correlation matrices of the modes of the power spectra shown in Fig. 3. The first two panels show the correlation matrices of the validation and the GAN generated maps, the third panel shows the difference between these correlation matrices. To provide a scale for comparison, in the fourth panel we also show the difference between the validation dataset correlation matrix and the correlations in an auxiliary dataset, see text for details.

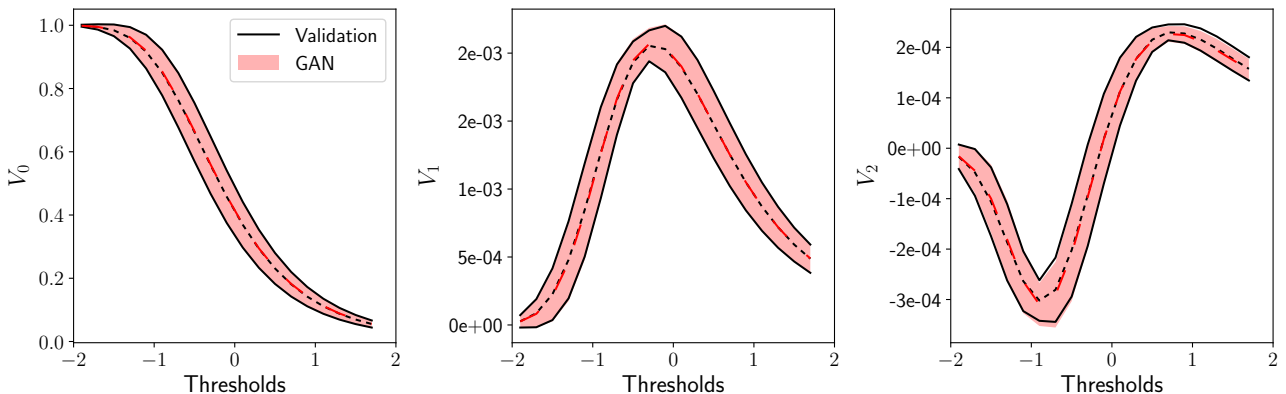


Figure 5: The Minkowski functionals of the convergence maps, which are sensitive to the non-Gaussian structures. We carried out the measurements on 100 batches of generated maps (6400 in total) and compare them to those of the validation maps. The functionals are evaluated at 19 thresholds and shown here are the bands of $\mu \pm \sigma$ at each threshold. The dashed lines represent the mean μ .

Discussion

Generative models for emulating scientific data and parameter inference. Without loss of generality, the generation of one random realization of a science dataset (simulation or otherwise) can be posited as a black-box model $S(\boldsymbol{\sigma}, r)$ where $\boldsymbol{\sigma}$ is a vector of the physical model parameters and r is a set of random numbers. The physical model S can be based on first principles or effective theories. Different random seeds generate statistically independent mock data realizations of the model parameters $\boldsymbol{\sigma}$. Such models are typically computationally expensive to evaluate at many different points in the space of parameters $\boldsymbol{\sigma}$.

A common problem for doing statistical inference from observations is making accurate predictions for a given summary statistic, for example the weak lensing power spectrum, or Minkowski functionals. Even in the simplest case, when the likelihood of the observed data \mathbf{S}_o is a multivariate Gaussian, it is characterized by both the mean data vector $\boldsymbol{\mu}$ and a covariance matrix \mathbf{C} :

$$\mathcal{L}(\mathbf{S}_o | \boldsymbol{\mu}, \mathbf{C}) = \frac{1}{\sqrt{|2\pi\mathbf{C}|}} \exp \left[-\frac{1}{2} (\mathbf{S}_o - \boldsymbol{\mu})^T \mathbf{C}^{-1} (\mathbf{S}_o - \boldsymbol{\mu}) \right] \quad (4)$$

Dealing with the non-linear problem of structure formation in the universe, we usually do not have analytic expressions for the mean and covariance in terms of the model parameters, and instead they have to be estimated from simulations mimicking the experiment. Modeling the mean of a target summary statistic can efficiently be done via existing emulator techniques [22–24]. That however is not true for the covariance matrix, where importantly any statistical noise in \mathbf{C}^{-1} propagates to an error in cosmological parameter constraints. In fact, future surveys will require of order 10^4 mock realizations per cosmological model just to prevent parameter degradation due to covariance uncertainty [25]. Alternative approaches to expensive simulations are therefore mandated for generating such large numbers of mock data sets.

The inference problem becomes even harder for

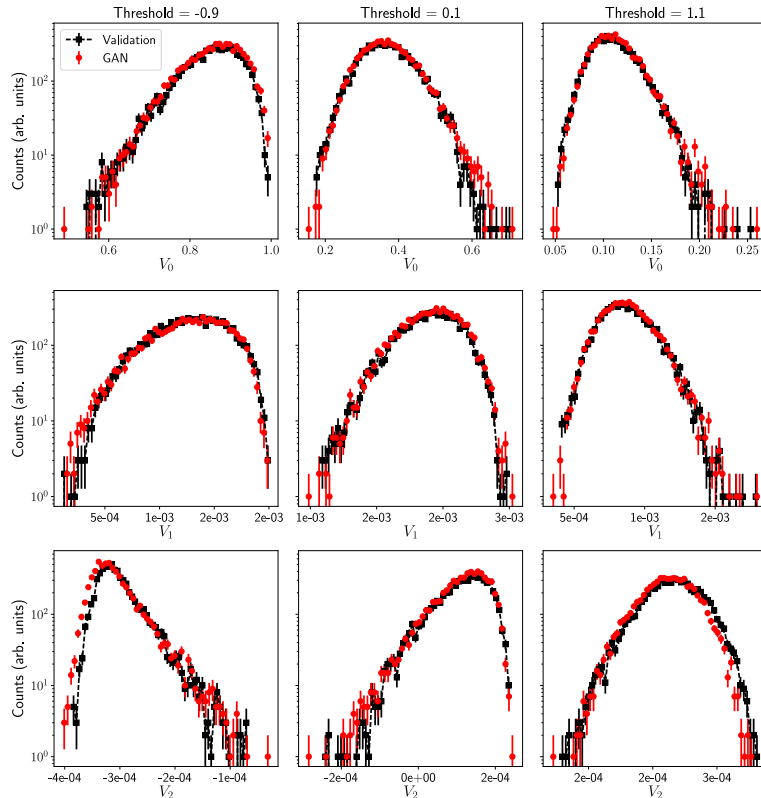


Figure 6: The distributions of the Minkowski functionals at 3 equidistant thresholds shown for illustration.

cases where a multivariate Gaussian likelihood is not a good approximation (for an example, see [26]). In those cases, it is often the best to adopt a likelihood-free method such as Approximate Bayesian Computation [27], which approximates the likelihood function through a massive Monte Carlo simulation of mock data realizations, and a summary statistic is used to quantify distance between real data and any mock realization. Needless to say, a “generator” which cheaply produces those mock realizations is an essential tool for those methods.

The idea of applying deep generative models techniques to emulating scientific data and simulations has been gaining attention recently [28–31]. In this letter we have shown that, with high statistical confidence, GANs are able to generate convergence maps of a particular Λ CDM model. Not only those maps share the same mean summary statistics as simulation data, but even more importantly, they reproduce the *statistical variety* present in simulations.

In our minds, the most important next step to build on this foundation and achieve an emulator of model S is the ability of generative models to generalize in the space of model parameters σ from datasets at a finite number of points in the parameter space. More specifically, *can we use GANs to build parametric density estimators $G_\theta(\sigma, z)$ of the physical model $S(\sigma, r)$?* Such generalization rests on smoothness and continuity of the response function of the physical model S

in the parameter space, but such an assumption is the foundation of any surrogate modeling. Future extensions of this work will seek to add this parameterization in order to enable the construction of robust and computationally inexpensive emulators of cosmological models.

Finally, the remarkable success of GANs in this work demonstrates that, unlike natural images which are the focus of the deep learning community, scientific data come with a suite of tools (statistical tests) which enable verifying and improving the fidelity of generative networks. The lack of such tools for natural images is one of the most serious challenges to understanding the relative performance of the different generative models architectures, loss functions and training procedures [13, 32]. As well as the impact within these fields, scientific data with its suite of tools have the potential to provide the grounds for benchmarking and improving on the state-of-the-art generative models.

Methods

Dataset. We use the cosmological simulations described in [21, 33], produced using the Gadget2 [34] N-Body simulation code and ray-traced with the Inspector Gadget weak lensing simulation pipeline [21, 33, 35] to produce weak lensing shear and convergence maps. A total of 50 runs were produced, each consisting of 512^3 particles in a box of size $240h^{-1}\text{Mpc}$. The cosmological parameters used in these simulations were $\sigma_8 = 0.798$, $w = -1.0$, $\Omega_m = 0.26$, $\Omega_\Lambda = 0.74$, $n_s = 0.96$, $H_0 = 0.72$. These simulation boxes were rotated and translated to produce

1000 ray-traced lensing maps covering 12 square degrees, with 2048×2048 pixels, which we downsampled to 1024×1024 pixels. The gravitational lensing illustrated by these maps can be described by the Jacobian matrix

$$A(\theta) = \begin{bmatrix} 1 - \kappa - \gamma_1 & -\gamma_2 \\ -\gamma_2 & 1 - \kappa + \gamma_1 \end{bmatrix} \quad (5)$$

where κ is the convergence, and γ is the shear.

The data was augmented by randomly cropping 200 256×256 maps from each of the original 1000 maps. The probability for a map to have a single pixel value outside $[-1.0, 1.0]$ range is less than 0.9% so it was safe to use the data without any normalization.

Data and code availability.

Code and example datasets are available at github.com/MustafaMustafa/cosmoGAN. The full dataset used in this work is available upon request from the authors. We also provide the fully trained model which was used for all results in this paper, how-to at link above.

Network architecture and training. The architectures of the generator and discriminator networks follow the Deep Convolutional Generative Adversarial Networks (DCGAN) guidelines and architecture [12]. The generator takes a 64-dimensional vector sampled from a normal prior $z \sim \mathcal{N}(0, 1)$. The first layer is a fully connected layer whose output is reshaped into a stack of feature maps. The rest of the network is four layers of transposed convolutions (a convolutional layer with fractional strides where zeroes are inserted between each column and row of pixels before convolving the image with the filter in order to effectively up-sample the image) that lead to a single channel 256×256 image. The outputs of all layers, except the output one, are batch-normalized [36] (normalizing each batch by subtracting by the mean activation of the previous batch, dividing by its standard deviation and learning a linear scaling) which was found to stabilize the training. A rectified linear unit (ReLU) activation [37] (output zero when the input less than zero and output equal to the input otherwise) is used for all except the output layer where a hyperbolic-tangent (tanh) is used. We list our choice for the number of feature maps in Table 1.

Operation	Output tensor size
Linear	$16 \times 16 \times 512$
Transpose Conv	$32 \times 32 \times 256$
Transpose Conv	$64 \times 64 \times 128$
Transpose Conv	$128 \times 128 \times 64$
Transpose Conv	$256 \times 256 \times 1$

Table 1: Feature tensors sizes of the DCGAN generator architecture we use in this work. All convolutions use symmetric kernels with size 5 and strides 2.

The discriminator utilizes four convolutional layers. The number of feature maps, stride and kernel sizes are the same as in the generator (Table 1). We reduced the number of filters from the DCGAN guidelines by a factor of 2, this effectively reduce the capacity of the generator/discriminator which worked well for our problem and trains faster. Output of all convolutional layers are activated with LeakyReLU [38] with parameter $\alpha = 0.2$. The output of the last convolutional layer is flattened and fed into a fully connected layer with a 1-dimensional output that is fed into a sigmoid. Batch Normalization is used for all except the first layer. Finally, we minimize the heuristic loss function [39] with batch-size of 64 maps and using an Adam optimizer [40] with the parameters suggested in the DCGAN paper, learning rate 0.0002 and $\beta_1 = 0.5$. We flip the real and fake labels with 1% probability to avoid the discriminator overpowering the generator too early into the training. We implement the networks in TensorFlow [41].

As is well experienced in practice, GANs with the heuristic loss function suffer from unstable updates towards the end of their training. This has been recently analyzed theoretically

in [14] and shown to happen when the discriminator is close to optimality but has not yet converged. For the results shown in this letter we trained the network until the generated maps pass the visual and pixel intensity Kolmogorov-Smirnov tests. This took 45 passes (or epochs) over all of the training data. Given the un-stability of the updates at this point, the performance of the generator on the summary-statistics tests starts varying uncontrollably. Therefore, to choose a generator, we train the networks for two extra epochs after epoch 45, and randomly generate 100 batches (6400 maps) of samples at every single training step. We evaluate the power spectrum on the generated samples and calculate Chi-squared distance measurement to the power spectrum histograms evaluated on a development subset of the validation dataset. We use the generator with the best Chi-squared distance. We verify that the generator is producing new maps by firstly finding training maps that are 'near' to generated maps and verifying that they are not identical. Near is defined in terms of L^2 distance in both the map space and in the power spectrum space, examples of the latter are shown in figure 7. Furthermore we verify that the generator has a smooth behavior in both map and power spectrum when interpolating between two randomly chosen values of the vector z as shown in figure 8.

Summary statistics analysis. To calculate the two-point correlation functions and Minkowski Functionals we use LensTools [19], a suite of analysis tools for Weak Gravitational Lensing. We evaluated the power spectrum spanning the range of $l = 200$ to $l = 50,000$ in 200 step sizes. The Minkowski Functionals thresholds ranged from -2.0 to 2.0 in 0.2 step sizes. LensTools was also used for bootstrapping and covariance calculations. We used a ROOT [42] implementation of the Andersen-Darling test and Scipy [43] for the Kolmogorov-Smirnov test.

References

- [1] DESI Collaboration, et al. The DESI Experiment Part I: Science, Targeting, and Survey Design. *ArXiv e-prints* (2016).
- [2] DESI Collaboration, et al. The DESI Experiment Part II: Instrument Design. *ArXiv e-prints* (2016).
- [3] Laureijs, R., et al. Euclid Definition Study Report. *ArXiv e-prints* (2011).
- [4] LSST Dark Energy Science Collaboration. Large Synoptic Survey Telescope: Dark Energy Science Collaboration. *ArXiv e-prints* (2012).
- [5] Spergel, D., et al. Wide-Field Infrared Survey Telescope-Astrophysics Focused Telescope Assets WFIRST-AFTA 2015 Report. *ArXiv e-prints* (2015).
- [6] Heitmann, K., et al. The Coyote Universe. II. Cosmological Models and Precision Emulation of the Nonlinear Matter Power Spectrum. *The Astrophysical Journal*, 705, 156–174 (2009). doi:10.1088/0004-637X/705/1/156.
- [7] Lawrence, E., et al. The Coyote Universe. III. Simulation Suite and Precision Emulator for the Nonlinear Matter Power Spectrum. *The Astrophysical Journal*, 713, 1322–1331 (2010). doi:10.1088/0004-637X/713/2/1322.
- [8] Csáji, B.C. Approximation with artificial neural networks. *MSc Thesis, Eötvös Loránd University (ELTE), Budapest, Hungary* (2001).
- [9] Goodfellow, I., et al. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N.D. Lawrence, and K.Q. Weinberger (editors), *Advances in Neural Information Processing Systems 27*, pages 2672–2680. Curran Associates, Inc. (2014).

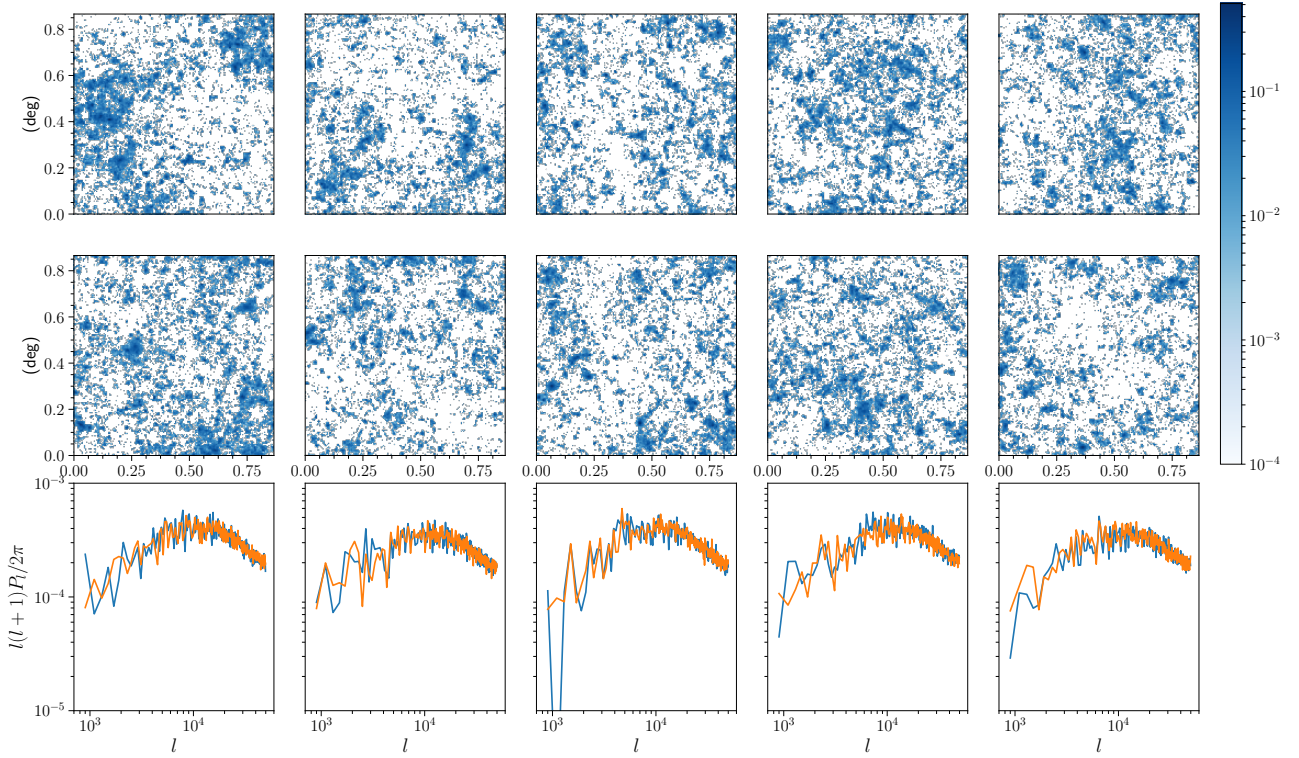


Figure 7: Comparison of randomly generated maps (top) with training maps (middle) that are their nearest-neighbor in terms of L^2 distance in power spectrum. While the power spectra (bottom) match very well, the maps are clearly different.

- [10] Albrecht, A., et al. Report of the Dark Energy Task Force. *ArXiv Astrophysics e-prints* (2006).
- [11] Kilbinger, M. Cosmology with cosmic shear observations: a review. *Reports on Progress in Physics*, 78(8), 086901 (2015).
- [12] Radford, A., Metz, L., and Chintala, S. Unsupervised representation learning with deep convolutional generative adversarial networks. *CoRR*, abs/1511.06434 (2015).
- [13] Theis, L., van den Oord, A., and Bethge, M. A note on the evaluation of generative models. *ArXiv e-prints* (2015).
- [14] Arjovsky, M. and Bottou, L. Towards Principled Methods for Training Generative Adversarial Networks. *ArXiv e-prints* (2017).
- [15] Arjovsky, M., Chintala, S., and Bottou, L. Wasserstein GAN. *ArXiv e-prints* (2017).
- [16] Liu, J., et al. Cosmology constraints from the weak lensing peak counts and the power spectrum in CFHTLenS data. *Physics Review D*, 91(6), 063507 (2015). doi:10.1103/PhysRevD.91.063507.
- [17] Abbott, T., et al. Cosmology from cosmic shear with Dark Energy Survey Science Verification data. *Physics Review D*, 94(2), 022001 (2016). doi:10.1103/PhysRevD.94.022001.
- [18] Jee, M.J., et al. Cosmic Shear Results from the Deep Lens Survey. I. Joint Constraints on Ω_M and σ_8 with a Two-dimensional Analysis. *Astrophysics Journal*, 765, 74 (2013). doi:10.1088/0004-637X/765/1/74.
- [19] Petri, A. Mocking the weak lensing universe: The lenstools python computing package. *Astronomy and Computing*, 17, 73 – 79 (2016). doi:https://doi.org/10.1016/j.ascom.2016.06.001.
- [20] Petri, A., et al. Emulating the CFHTLenS weak lensing data: Cosmological constraints from moments and Minkowski functionals. *Physics Review D*, 91(10), 103511 (2015). doi:10.1103/PhysRevD.91.103511.
- [21] Kratochvil, J.M., et al. Probing cosmology with weak lensing Minkowski functionals. *Physics Review D*, 85(10), 103513 (2012). doi:10.1103/PhysRevD.85.103513.
- [22] Kwan, J., et al. Cosmic Emulation: The Concentration-Mass Relation for Λ CDM Universes. *The Astrophysical Journal*, 768, 123 (2013). doi:10.1088/0004-637X/768/2/123.
- [23] Heitmann, K., et al. The Coyote Universe Extended: Precision Emulation of the Matter Power Spectrum. *The Astrophysical Journal*, 780, 111 (2014). doi:10.1088/0004-637X/780/1/111.
- [24] Kwan, J., et al. Cosmic Emulation: Fast Predictions for the Galaxy Power Spectrum. *The Astrophysical Journal*, 810, 35 (2015). doi:10.1088/0004-637X/810/1/35.
- [25] Dodelson, S. and Schneider, M.D. The effect of covariance estimator error on cosmological parameter constraints. *Physical Review D*, 88(6), 063537 (2013). doi:10.1103/PhysRevD.88.063537.
- [26] Davies, F.B., et al. A New Method to Measure the Post-Reionization Ionizing Background from the Joint

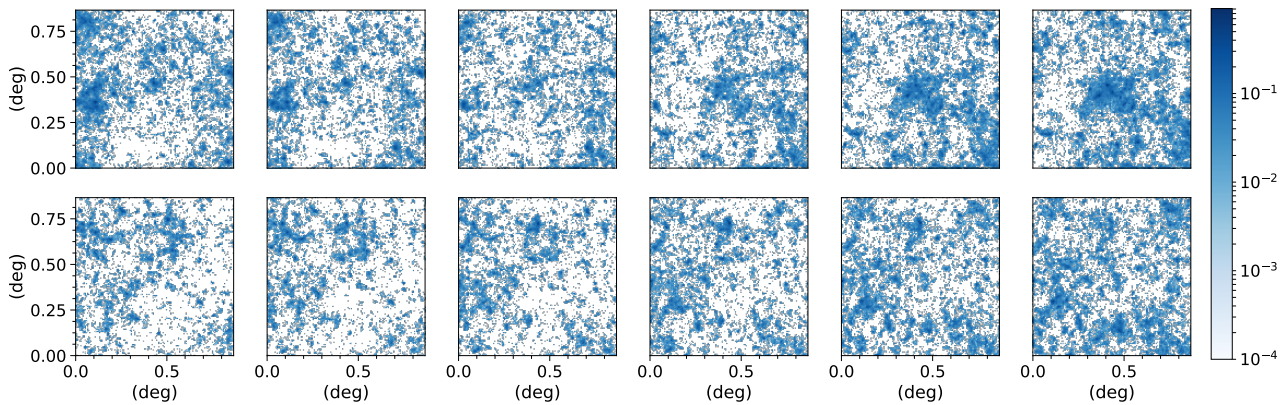


Figure 8: Each row is a random example of maps produced when interpolating between two randomly chosen vectors z (left to right). The generator varies smoothly between points in the prior space.

- Distribution of Lyman- α and Lyman- β Forest Transmission. *ArXiv e-prints* (2017).
- [27] Pritchard, J.K., et al. Population growth of human Y chromosomes: a study of Y chromosome microsatellites. *Molecular Biology and Evolution* (1999).
- [28] Ravanbakhsh, S., et al. Enabling Dark Energy Science with Deep Generative Models of Galaxy Images. *arXiv:1609.05796* (2016).
- [29] de Oliveira, L., Paganini, M., and Nachman, B. Learning Particle Physics by Example: Location-Aware Generative Adversarial Networks for Physics Synthesis. *arXiv:1701.05927* (2017).
- [30] Paganini, M., de Oliveira, L., and Nachman, B. CaloGAN: Simulating 3D High Energy Particle Showers in Multi-Layer Electromagnetic Calorimeters with Generative Adversarial Networks. *arXiv:1705.02355* (2017).
- [31] Mosser, L., Dubrulle, O., and Blunt, M.J. Reconstruction of three-dimensional porous media using generative adversarial neural networks. *arXiv:1704.03225* (2017).
- [32] Salimans, T., et al. Improved techniques for training gans. *CoRR*, abs/1606.03498 (2016).
- [33] Yang, X., et al. Cosmological information in weak lensing peaks. *Phys. Rev. D*, 84, 043529 (2011). doi:10.1103/PhysRevD.84.043529.
- [34] Springel, V. The cosmological simulation code GADGET-2. *Monthly Notices of the Royal Astronomical Society*, 364, 1105–1134 (2005). doi:10.1111/j.1365-2966.2005.09655.x.
- [35] Kratochvil, J.M., Haiman, Z., and May, M. Probing cosmology with weak lensing peak counts. *Phys. Rev. D*, 84, 043519 (2010).
- [36] Ioffe, S. and Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In F. Bach and D. Blei (editors), *Proceedings of the 32nd International Conference on Machine Learning*, volume 37, pages 448–456. PMLR, Lille, France (2015).
- [37] Nair, V. and Hinton, G.E. Rectified linear units improve restricted Boltzmann machines. In J. Fürnkranz and T. Joachims (editors), *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 807–814. Omnipress, Haifa, Israel (2010).
- [38] Maas, A.L., Hannun, A.Y., and Ng, A.Y. Rectifier nonlinearities improve neural network acoustic models. In S. Dasgupta and D. McAllester (editors), *Proceedings of the 30th International Conference on Machine Learning*, volume 28. PMLR, Atlanta, Georgia, USA (2013).
- [39] Goodfellow, I.J. NIPS 2016 tutorial: Generative adversarial networks. *CoRR*, abs/1701.00160 (2017).
- [40] Kingma, D.P. and Ba, J. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980 (2014).
- [41] Abadi, M., et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems (2015).
- [42] Brun, R. and Rademakers, F. Root — an object oriented data analysis framework. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 389(1), 81 – 86 (1997). doi:https://doi.org/10.1016/S0168-9002(97)00048-X. New Computing Techniques in Physics Research V.
- [43] Jones, E., et al. SciPy: Open source scientific tools for Python (2001–).

Acknowledgements

M.M. would like to thank Evan Racah for his invaluable deep learning related discussions throughout this project. M.M. would also like to thank Taehoon Kim for providing an open-sourced TensorFlow implementation of DCGAN which was used in early evaluations of this project. We would also like to thank Ross Girshick for helpful suggestions. No network was hurt during the adversarial trainings performed in this work. M.M., D.B. and W.B. are supported by the U.S. Department of Energy, Office of Science, Office of High Energy Physics, under contract No. DE-AC02-05CH11231 through the National Energy Research Scientific Computing Center and Computational Center for Excellence programs.