Emulating Cosmological Simulations with GANs

CAPT, Nottingham University

Andrius Tamošiūnas

andrius.tamosiunas@nottingham.ac.uk

In collaboration with:

Hans Winther, Kazuya Koyama, David Bacon, Bob Nichol, Ben Mawdsley

Contents

Introduction: Generative Adversarial Networks

Generative Models in Physics

- One of the key issues in observational sciences is producing statistically realistic **mock data**;
- This is of great value in astronomy, particle physics, high energy physics and many other fields;
- A typical astronomy use case: generating mock data for future surveys;
- A wide array of techniques exist:

  1. Classic techniques: Latent Dirichlet Allocation, Gaussian Mixture Models, various Bayesian techniques;
  2. GANs and VAEs;
  3. Other types of emulators.

Generative Adversarial Networks

- GANs refer to a system of neural networks that are trained adversarially;
- The **generator** produces novel data;
- The **discriminator** classifies the data into *real* or *fake*;

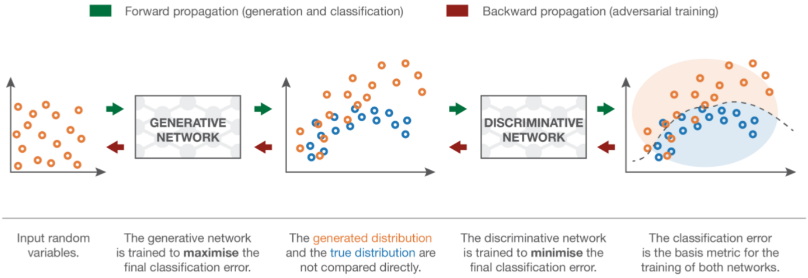


Fig.: The GAN training procedure (Joseph Rocca, 2019)

What Can GANs Do?

Oh! horrid Night! Melody release thee from my aching Heart, And Fate copied on my Mind the wandring Misfortunes?

Home, home!
Home, sweet home!
Home, sweet!
Home! home! home! home!

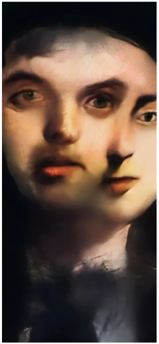
Frail, frail, fiery phantom,
Death's wan, ghostly form soared
Into the ether; threatening
To burn her forever, and die.

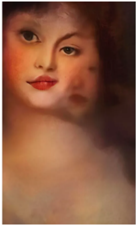
Ours learned the wisdom of
ancient groups
Of ancient kings.
I remember the wild useless war

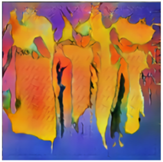
My Imagination is a
Storehouse of all that is within

Saeed et al. (2019)
ai-fragments.com

What Can GANs Do?



Ellie Packard, 2019
Dmitry Soshnikov, 2020
AIArtists.org
obvious-art.com

GANs in Cosmology

- GANs have also been extensively used in cosmology for a variety of applications:
  1. Weak lensing convergence map generation;
  2. Generating mock galaxies;
  3. Generating deep field images;
  4. Emulating the cosmic web in 2D and 3D (Rodriguez et al., 2018 + Perraudin et al., 2019).

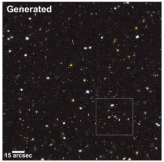


Fig.: A GAN-generated deep field image (Smith & Geach, 2019).

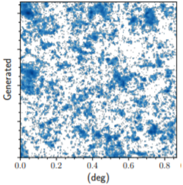


Fig.: A GAN-generated convergence map (Mustafa et al., 2019).

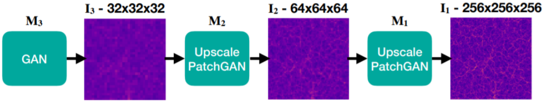


Fig.: A GAN-generated cosmic web distribution (Perraudin et al., 2019).

Part 1: Emulating the Cosmic Web and WL Convergence Maps

Based on arXiv:2004.10223

Key Goals

- Modify the **cosmoGAN** algorithm to allow emulating datasets with different parameters (cosmic web slices and weak lensing convergence maps);
- Simultaneously emulate **dark matter** and **hydrodynamic** simulation data (Illustris);
- Investigate the **latent space** of the algorithm (LS interpolation, LS arithmetic, Riemannian geometry).

The Algorithm

- A modification of the **cosmoGAN** algorithm (Mustafa et al., 2019). Also inspired by Rodriguez et al., 2018;
- The summary of the used architecture:

	Activ.	Output shape	Params.
Latent	–	64	–
Dense	–	$512 \times 16 \times 16$	8.5M
BatchNorm	ReLU	$512 \times 16 \times 16$	1024
TConv 5×5	–	$256 \times 32 \times 32$	3.3M
BatchNorm	ReLU	$256 \times 32 \times 32$	512
TConv 5×5	–	$128 \times 64 \times 64$	819K
BatchNorm	ReLU	$128 \times 64 \times 64$	256
TConv 5×5	–	$64 \times 128 \times 128$	205K
BatchNorm	ReLU	$64 \times 128 \times 128$	128
TConv 5×5	Tanh	$1 \times 256 \times 256$	1601
Total trainable parameters			**12.3M**

	Activ.	Output shape	Params.
Input map	–	$1 \times 256 \times 256$	–
Conv 5×5	LReLU	$64 \times 128 \times 128$	1664
Conv 5×5	–	$128 \times 64 \times 64$	205K
BatchNorm	LReLU	$128 \times 64 \times 64$	256
Conv 5×5	–	$256 \times 32 \times 32$	819K
BatchNorm	LReLU	$256 \times 32 \times 32$	512
Conv 5×5	–	$512 \times 16 \times 16$	3.3M
BatchNorm	LReLU	$512 \times 16 \times 16$	1024
Linear	Sigmoid	1	131K
Total trainable parameters			**4.4M**

Fig.: The architecture of the generator (**left**) and the discriminator (**right**) neural networks.

The Pipeline

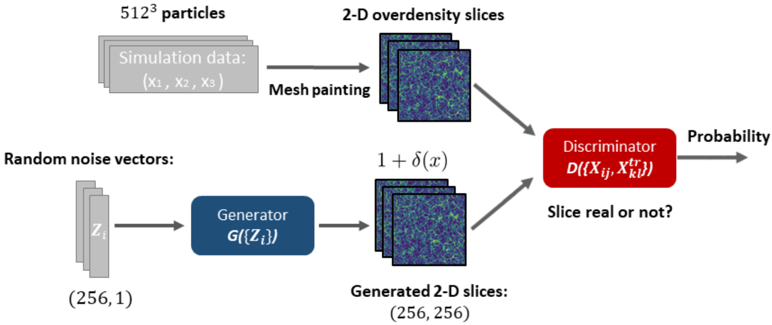
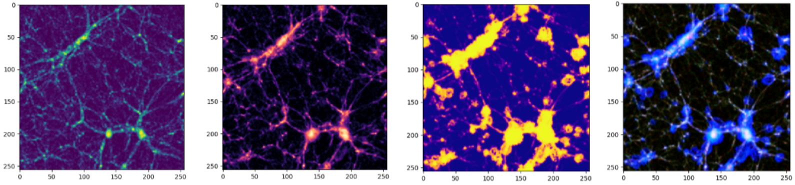


Fig.: Training the GAN pipeline.

The Dataset

1. WL convergence maps: 256×256 pixel, $\Omega_m = 0.233$, $\sigma_8 = \{0.436, 0.814\}$, 8000 maps (Columbia Lensing (2020));
2. CW slices (2D): 256×256 pixel, $h = 0.7, \Omega_\Lambda = 0.72$ and $\Omega_m = 0.28$ for $z = \{0, 1\}$, $\sigma_8 = \{0.7, 0.9\}$ and $f_{\mathrm{R0}} = \{10^{-7}, 10^{-1}\}$, 5000 slices (L-PICOLA/MG-PICOLA);
3. Illustris 2D slices (DM + gas overdensity, internal energy field): $\Omega_m = 0.2726, \Omega_\Lambda = 0.7274, h = 0.704$, 5000 slices (Illustris-3)



(a) DM overdensity field (b) Gas overdensity field (c) Internal energy field (d) All components combined

Fig.: The Illustris-3 data.

WL Results

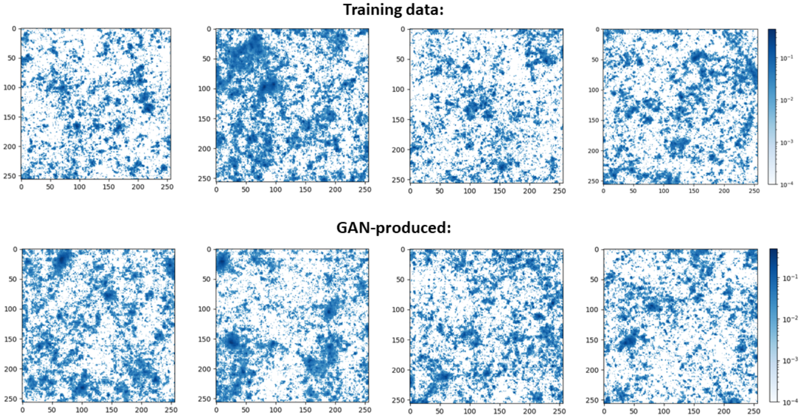


Fig.: GAN-produced WL samples (log-normalized color map).

WL Results

- The training takes **24-72 hours** on Google Cloud;
- The GAN produces **novel** realistic WL convergence maps;
- The results are **log-normalized** between [0,1] to emphasize key features.

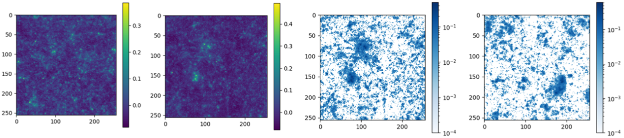
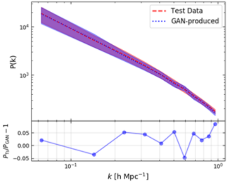


Fig.: GAN-produced WL samples.



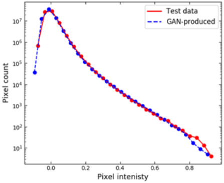


Fig.: Powerspectrum and the convergence histogram.

WL Results: Minkowski Functionals

- Minkowski functionals correspond to the integrated area, boundary length and the geodesic curvature at different value thresholds:

$$V_0(\nu) = \int_{Q_\nu} d\Omega, \quad V_1(\nu) = \int_{\partial Q_\nu} \frac{1}{4} dl, \quad V_2(\nu) = \int_{\partial Q_\nu} \frac{\kappa_c dl}{2\pi}. \qquad (1)$$

- MFs allow to capture the non-Gaussian information in some field.

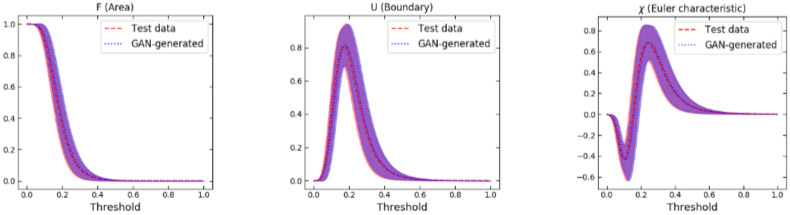


Fig.: MFs for the test and the GAN-generated datasets.

WL Results: Different σ_8 Values

- The GAN can also be trained on WL maps with $\sigma_8 = \{0.436, 0.814\}$;
- We find that the generator produces realistic WL maps and does not get *confused* between the different σ_8 values:

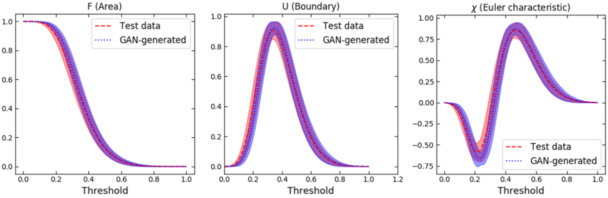


Fig.: MFs for the test and the GAN-generated datasets.

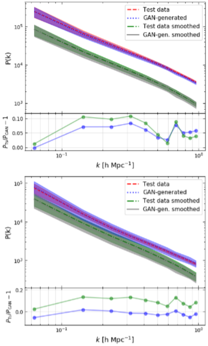


Fig.: Powerspectra comparison.

CW Results: Different z Values

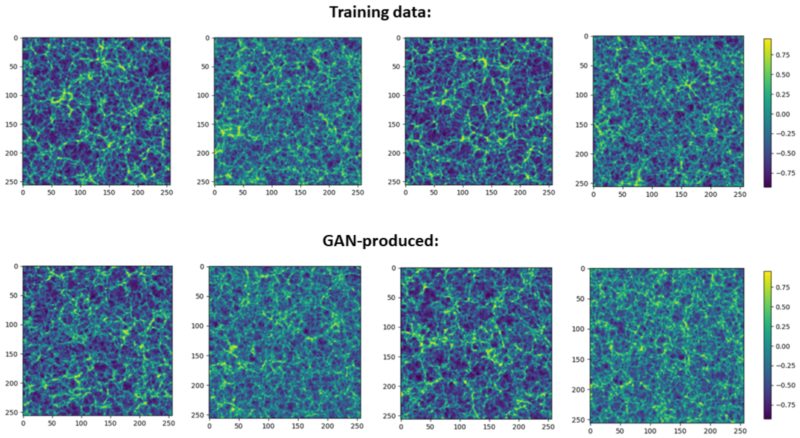


Fig.: Comparing the training data against the GAN-generated samples.

CW Results: Different z Values

- Using the same architecture we can generate 2D cosmic web slices;
- This can be done for different cosmologies, redshifts, modified gravity models.

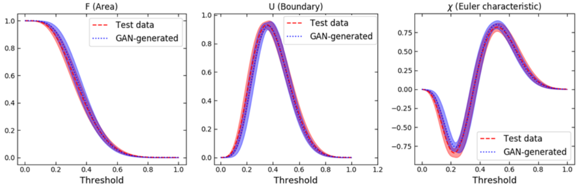


Fig.: MFs for the test and the GAN-generated datasets.

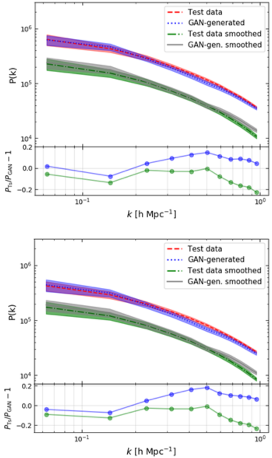


Fig.: Powerspectra comparison.

Illustris Results

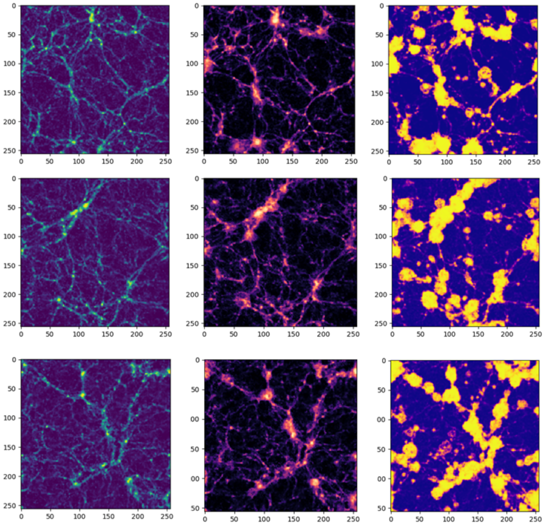


Fig.: Illustris dataset results. Training data (**top row**) vs. GAN-produced samples (**mid and bottom rows**).

- The powerspectra shows good agreement for all the components (**1-10%**);
- **Cross-power spectrum** shows the correlation between the different data components on different scales:

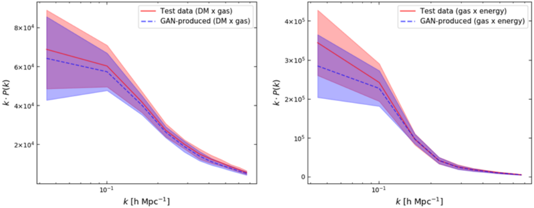


Fig.: Cross-power spectrum analysis.

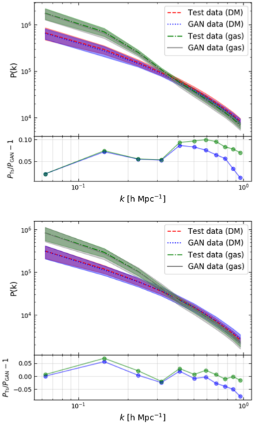


Fig.: Powerspectra comparison.

Illustris Results

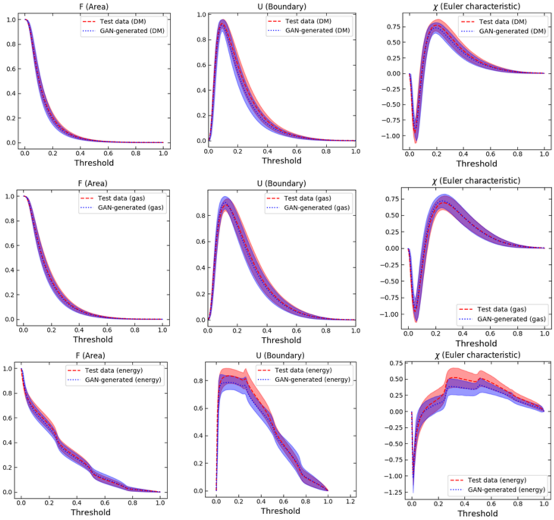


Fig.: Minkowski functional analysis.

Part 3: Latent Space Interpolation

Latent Space interpolation

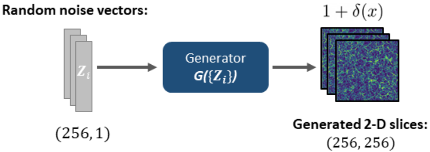


Fig.: Latent space input to the Generator.

Fig.: An intuitive example of the latent space interoplation (StyleGAN).

Latent Space Interpolation

- We can define an N-dimensional space spanned by the latent space vectors;
- During the training procedure, the structure of the latent space is produced;
- Clusters in the latent space correspond to certain features of the dataset;
- One can interpolate between different points in the latent space:

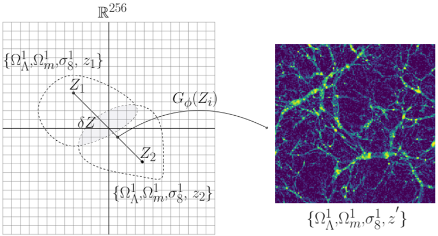


Fig.: Latent space interpolation in cosmoGAN.

- The latent space can be visualized by projecting the 256-dimensional space in 2D;
- We sample points around the latent space line and measure $\Delta P_k \equiv \sum_i \log_{10}(P_2(k_i)) - \log_{10}(P_1(k_i))$:

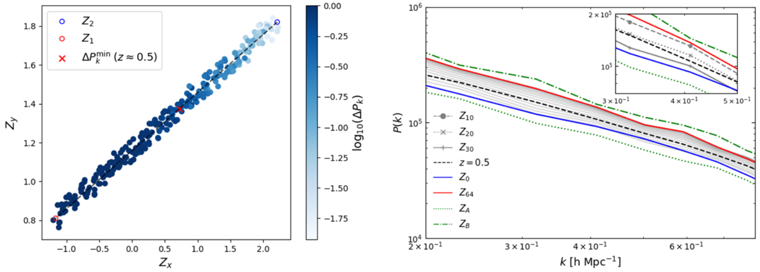
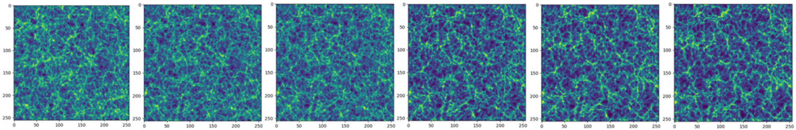


Fig.: Latent space visualization (**left**) and the corresponding powerspectra (**right**).

Results

z **interpolation:**



σ_8 **interpolation:**

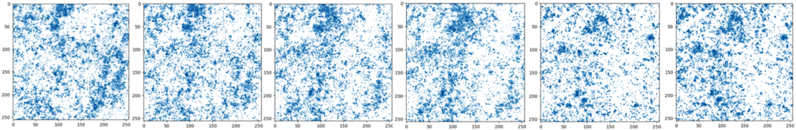


Fig.: Latent space interpolation results.

Geometry of Generative Models

- An active research direction: the properties of the GAN/VAE latent space;
- A novel approach: study the GAN properties using **Riemannian geometry** (see Shao et al., 2017 and Wang and Ponce, 2021);
- A GAN generator in this framework can be described as a mapping from latent space Z to a higher dimensional data manifold X: $G_\phi : Z \to X$;
- It is then natural to define an **induced metric** $g = J(Z)^T J(Z)$, where J is the Jacobian:

$$J = \begin{bmatrix} \frac{\partial X^1}{\partial Z^1} & \frac{\partial X^1}{\partial Z^2} & \cdots & \frac{\partial X^1}{\partial Z^n} \\ \vdots & \vdots & \ddots & \\ \frac{\partial X^m}{\partial Z^1} & \frac{\partial X^m}{\partial Z^2} & \cdots & \frac{\partial X^m}{\partial Z^n} \end{bmatrix}. \tag{2}$$

Geometry of Generative Models:

- A metric allows finding geodesics κ^α on the **data manifold**:

$$\frac{d^2\kappa^\alpha}{dt^2} = -\Gamma^\alpha_{\beta\gamma}\frac{d\kappa^\beta}{dt}\frac{d\kappa^\gamma}{dt}, \qquad (3)$$

with the Christoffel symbols:

$$\Gamma^\alpha_{\beta\gamma} = \frac{1}{2}g^{\alpha\delta}\left(\frac{\partial g_{\delta\beta}}{\partial X^\gamma} + \frac{\partial g_{\delta\gamma}}{\partial X^\alpha} - \frac{\partial g_{\alpha\beta}}{\partial X^\delta}\right). \qquad (4)$$

- The **curvature** of the data manifold:

$$R^\rho_{\sigma\mu\nu} = \partial_\mu\Gamma^\rho_{\nu\sigma} - \partial_\nu\Gamma^\rho_{\mu\sigma} + \Gamma^\rho_{\mu\lambda}\Gamma^\lambda_{\nu\sigma} - \Gamma^\rho_{\nu\lambda}\Gamma^\lambda_{\mu\sigma}. \qquad (5)$$

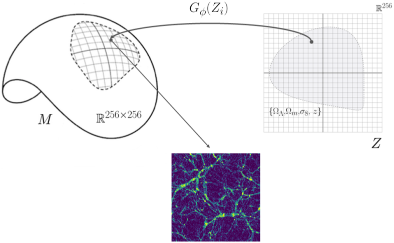


Fig.: Geometry of GANs.

Conclusions and Future Work

- A lot of avenues remain to be explored:
  1. Generating data in **3D**;
  2. **High resolution** datasets;
  3. Directly controlling the **cosmological parameters**.

- On the theoretical side:
  1. How does the GAN *build* the structure of the latent space?
  2. What are some other features of the Riemannian manifolds produced by GANs?
  3. How does this relate to the field of **information geometry**?

- For further questions or feedback contact me at: **andrius.tamosiunas@nottingham.ac.uk**