Emulating deviations from Einstein's General Relativity using conditional GANs

Yash Gondhalekar
Department of CSIS
BITS Pilani, K.K. Birla Goa Campus
yashgondhalekar567@gmail.com

Sownak Bose
Institute for Computational Cosmology
Department of Physics, Durham University, Durham, DH13LE, UK
sownak.bose@durham.ac.uk

Abstract

Computationally expensive simulations pose a severe bottleneck, especially in astronomy, where several realizations of the same physical processes are required to facilitate scientific studies, such as exploring new physics or constraining the underlying physics by comparing it with observations. Simulations that modify Einstein's gravity require solving highly non-linear equations and take $\sim$10-20 times more time than the normal ones. In order to mitigate this bottleneck, we use a conditional generative adversarial network (cGAN) to map output fields from normal simulations to output fields of time-consuming simulations. Our model uses a frequency-based loss during training and uses indirect emulation wherein the mapping is achieved using ratio fields instead of the traditional input $\rightarrow$ output domain translation. Our cGAN agrees well with the ground-truth images despite the visually minor differences between fields from the input and output domains.

1 Introduction

A cosmological model describes the Universe by providing information about space-time geometry, matter, its physical behavior, and how matter and geometry are related. The Λ Cold Dark Matter (ΛCDM) model is the currently most widely accepted model (also called the standard model of cosmology) and is based on Einstein's general theory of relativity (GR). It agrees well with several observed properties of the Universe, such as the accelerated expansion of the Universe and the observed distribution of clusters of galaxies in the Universe [e.g., 21, 17, 24]. However, it has also been found to be in tension with some observational datasets and suffers from theoretical issues such as coincidence and fine-tuning problems [see, e.g., 6, 18, for reviews]. These prevailing issues have motivated the development of alternative theories to ΛCDM that can resolve these issues while explaining observations well.

Testing deviations from GR on cosmological scales is a key science target of several ground- and space-based surveys on which the community has spent billions of dollars. One promising alternative to ΛCDM is the class of modified gravity ([5]; MG) models, which modify GR equations from ΛCDM. As a result, the behavior and evolution of matter in MG differs from ΛCDM. Since GR is experimentally confirmed to be accurate in our local Universe but not yet on the (large) cosmological scales, MG models devise a new force law that converges to GR in the local Universe. Apart from the four fundamental forces of nature, these modifications to gravity can be considered a fifth force. The properties of this new fifth force are well constrained so that it is compatible with experiments

Machine Learning and the Physical Sciences Workshop, NeurIPS 2023.

deeming GR to be valid in the local Universe. The constraint is obtained by a 'screening' mechanism that suppresses the fifth force or, equivalently, hides modifications to Einstein's gravity in local regions, such as our solar system (such regions are also of high density compared to the average density of the entire Universe).

A particularly widely studied MG theory is $f(R)$ gravity [1], parametrized by two free parameters, n and $|f_{R0}|$. However, due to the modified nature of gravity, the resulting equations in $f(R)$ (and MG, in general) become highly non-linear, due to which N-body simulations are necessary for studying matter evolution in $f(R)$ [see e.g., 13]. In our internal experiments, a typical $f(R)$ simulation run requires at least ten times more time than the corresponding ΛCDM simulations, which is a considerable bottleneck.

Deep learning methods have been regarded as promising emulators in cosmology applications in recent years [e.g., 16, 4, 19]. In this study, we aim to leverage a cGAN [15] to learn a mapping from the ΛCDM density fields (denoting matter distribution) to its $f(R)$ counterpart by casting the mapping as an image-to-image translation problem. We show that meaningful mapping can be learned despite the intricate differences between the matter distribution of the two gravity models.

2 Methods

2.1 Data

Simulations for the ΛCDM and $f(R)$ models are run using the MG-GLAM [8] code, an efficient extension of the GLAM [12] parallel particle mesh (PPM) N-body code. The implementation of the $f(R)$ gravity model in MG-GLAM follows [9]. The modified gravity parameters are turned off for the corresponding ΛCDM simulations. We use $n = 1$ and a variant of $f(R)$ gravity having $|f_{R0}| = 10^{-4}$. The length of our simulation box is $L = 128\ h^{-1}$ Mpc, and the number of particles is $N_p = 256^3$. We focus on simulation results from redshift zero, which represents the current epoch of the Universe. Five different realizations for ΛCDM and $f(R)$ simulations are run to generate sufficient training data for training the neural network.

We use the Delaunay Tessellation Field Estimator (DTFE) code of [3] to estimate the volume-averaged density fields and interpolate them on a grid of size 512^3. For computational reasons, we extract 2D maps across the three axes (X–Y, Y–Z, X–Z) from each three-dimensional density field to generate our data. As a common practice in cosmology applications, maps are not randomly split into training, validation, and testing sets, as it can lead to overestimating model performance [20]. Instead, maps from 3D fields of three realizations are used for training. One realization is used for validation, and the other for testing.

For ease of notation, density fields from ΛCDM and $f(R)$ will be called GR and F4, respectively.

2.2 Model architecture and training details

Our basic GAN architecture follows the `pix2pix` software [10]. The discriminator of our cGAN (called PatchGAN) uses the loss from the Least Squares Generative Adversarial Network (LSGAN), where an L2 loss is used instead of binary cross-entropy that addresses the vanishing gradient problem [14]. The objective is: $\min_D \mathcal{L}_{\text{LSGAN}}(D) = \frac{1}{2}\mathbb{E}_{x,y}[(D(x,y)-b)^2] + \frac{1}{2}\mathbb{E}_{x,z}[(D(x,G(x,z))-a)^2]$ and $\min_G \mathcal{L}_{\text{LSGAN}}(G) = \frac{1}{2}\mathbb{E}_{x,z}[(D(x,G(x,z))-c)^2]$, where x is the input GR map, y is the prediction (in our case, this is the pixel-wise ratio of $f(R)$ and GR maps; see Sect. 2.3), and z is a random noise vector, D and G are the discriminator and generator models. $a = 0$, $b = 1$, since the fake and real images are labeled 0 and 1, respectively, and $c = 1$ since the generator aims to fool the discriminator.

The additional L1 loss term used in `pix2pix` takes the form $\lambda\mathbb{E}_{x,y,z}[||y - G(x,z)||_1]$, which G tries to minimize and enforces the generated images closer to the ground truth. The L1 loss can prevent mode-collapse and takes advantage of the supervised nature of cGAN. We use $\lambda = 200$ after finding that it works best for our case.

The L1 loss primarily forces the low-frequency components to match the ground truth. While `pix2pix` accounts for high-frequency correctness using the PatchGAN discriminator [10], a study by [11] proposed that explicitly incorporating a frequency-based loss function, complementary to the

(spatial-domain) L1 loss improves image translation. The authors call this focal frequency loss (FFL), which adapts itself to pay more attention to frequency components that are difficult to produce. Since cosmological fields carry important frequency information (quantified by the power spectrum, see Sect. 3), such a loss function is a natural choice. Our application uses a combination of L1 and FFL losses with equal weightage (0.5) for both terms. During training, we use the common practice of performing one gradient descent step of D followed by one step for G [7]. While optimizing D, we divide the objective by two to slow down the rate at which D learns compared to G. Dropout layers in the model introduce stochasticity during training [10], which are turned off during evaluation.

The generator in `pix2pix` is a U-Net [23], an encoder-decoder architecture equipped with skip connections between mirrored layers from the encoder and the decoder. These connections are essential to prevent the loss of spatial details during image downsampling. Our encoder consists of nine convolutional layers, transforming the 512×512 maps into 1×1 dimensions at the bottleneck. The decoder then transforms this latent representation using a series of transposed convolutional operations. Our PatchGAN contains four convolutional layers and predicts whether a given image is real or fake. Each convolutional layer in our architecture has a kernel size of 4, ReLU activation (leaky for the discriminator and the encoder in the generator with slope 0.2 but non-leaky for the decoder in the generator), and instance normalization [25] is used. The final operation performed in the generator is the Tanh activation, which restricts the dynamic range of densities to $[-1, +1]$. We have used a scaled natural logarithm transformation ($\ln x/c$) for preprocessing the maps since we found it to perform better than the transformation of [22] commonly used in GAN applications in cosmology. We set $c = 10$ based on the densities from the training set.

Models are saved after every ten epochs during training. After training, all saved models are used on the validation set maps, and the model with the best performance (see Sect. 3 for the metrics) is selected for evaluation on the test set.

2.3 Ratio fields approach

The differences in the density fields of GR and F4 are minor, making it challenging to learn meaningful mapping. Our internal experiments suggested that a direct mapping of fields from GR to $f(R)$ could not produce plausible results based on the metrics in Sect. 3. We thus propose to map the GR field to the ratio field, i.e., $f(R)/$GR, since it better reveals the intricate differences and assists the cGAN in learning the mapping better. The final F4 prediction is obtained by multiplying the predicted ratio map with the simulated GR map.

3 Results

Fig. 1 shows an example visualization of the simulated GR and F4 maps and the cGAN prediction for a random map from the test set. The string-like structures in the maps are called filaments and are thicker in $f(R)$ due to enhanced gravitational forces compared to GR [13], as shown by the white and green ovals. The cGAN successfully reproduces the enhanced thickness. Some extremely high densities (denoted by the orange-white color) tend to not be reproduced accurately by the cGAN, as seen in the lower-right part of the green oval.

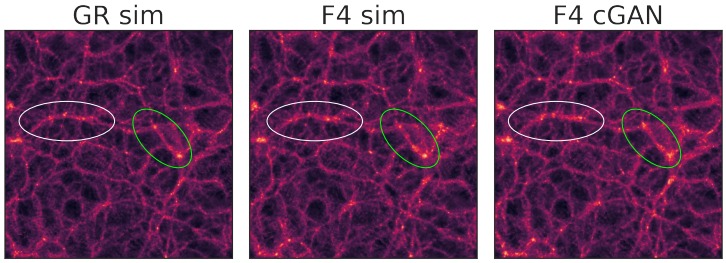

Figure 1: Sample visualization of the simulated GR and F4 and the cGAN predicted F4 map. Maps are shown in the logarithm scale. The map's dimensions are 512×512 (see Sect. 2.1 for details).

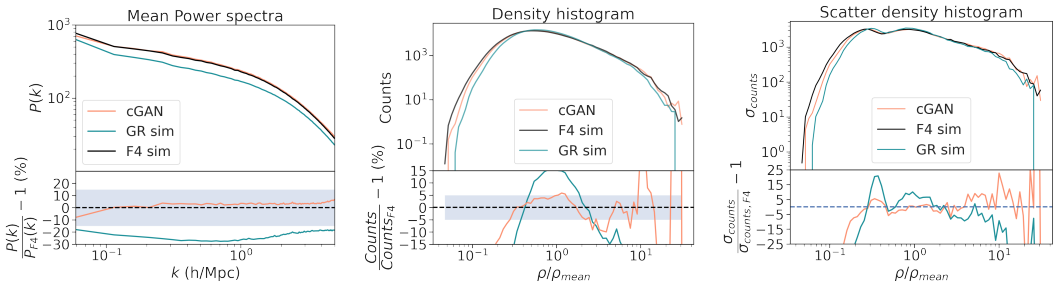

Figure 2: The mean power spectra, density histogram, and the standard deviation of counts averaged across the entire test set. For histograms, binning is done in log space.

To quantify the performance of the cGAN, we compare the power spectrum calculated by the Pylians package [26], the histogram of the densities, and the standard deviation in the histogram counts in Fig. 2. The density histograms are calculated on fields smoothed using a spherical top-hat filter of smoothing scale, $R_{th} = 10\,h^{-1}$Mpc, similar to [2]. The power spectrum of the cGAN prediction agrees exceptionally well with the ground truth (F4 sim) with, on average, $\lesssim 5\%$ error. The density histogram suggests that the agreement of the cGAN prediction is good for the densities, $0.25 \lesssim \rho/\rho_{\rm mean} \lesssim 10$, whereas worse for extremely low or high densities. While this suggests that outlier densities in the maps are not reproduced well, the bulk of the dynamic range of densities is reproduced reasonably accurately by the cGAN. The rightmost plot shows the bin-wise standard deviation of the counts from the middle histogram plot. This figure shows that cGAN can reasonably reproduce the scatter in the density distribution across the entire range of densities except for the most empties regions, $\rho/\rho_{\rm mean} \lesssim 0.2$. Thus, the cGAN can generate density maps consistent with the ground-truth F4 maps in terms of the density distribution's first and second moments (i.e., mean and standard deviation, respectively).

Table 1: Comparison of execution times of F4 simulations and our proposed approach. t_{cGAN} includes the elapsed wall-clock time on the CPU for preprocessing the input map and the forward pass of the model in inference mode and is calculated using a randomly selected map from the test dataset. t_{f4} and t_{gr} are the time required for F4 and GR simulations, and N_g is the grid size in one dimension, which is 512 here. Simulations use 32 cores (each core contains a single thread in our case), whereas cGAN inference uses a single thread, so the simulation time is the total CPU time obtained by scaling the wall-clock time by the no. of threads. I/O overhead for cGAN emulation and simulations are assumed to be similar.

	F4 simulation	Our approach
Time required (mins)	$t_{f4} = 1610.28$	$t_{gr} = 88.74$ $t_{cGAN} = 0.0268$
Total time (mins)	$t_{f4} = \mathbf{1610.28}$	$t_{gr} + N_g \times t_{cGAN} = \mathbf{102.46}$

Table 1 compares the execution times of our proposed approach using the cGAN emulator and the N-body simulations using the MG-GLAM code. We recall that the output of N-body simulations is particle positions in a 3D volume, which is then interpolated onto the 512^3 grid. However, our cGAN is designed to emulate 2D maps, which are slices of the 3D grid. Hence, the time required by the cGAN is scaled by 512 to find the time required to emulate the entire 3D density field. We assume that the time DTFE requires to interpolate the particle positions output from the GR and F4 simulations is the same and is not included in the comparison. We average each of these times across a few independent runs. The table shows that our approach requires $\sim$16 times less time than the F4 simulations, significantly alleviating the computational burden. We further stress that the MG-GLAM simulation code was designed with a focus on efficiency and optimization and was found to be $\sim$100-300 times faster than other full N-body simulation codes such as ECOSMOG or MG-AREPO in [8]. Thus, our cGAN approach would reduce the computational costs by an order of

thousands compared to full N-body simulations of F4. Time for cGAN emulation was measured on the CPU; access to GPUs would make the benefits more pronounced.

The generator of our cGAN contains 267.93M parameters, and its computational complexity is 284.69 GMac[1] (Mac: Multiply-Accumulate Operations) for a single 512×512 resolution map. These facts are noted to inform the computational costs of our method and to decide whether our cGAN approach is feasible based on the computing resources available to the user.

Our preliminary results hint that the generator's test-time performance shows a non-trivial variance across different training runs, which could arise due to the large size of the generator or due to the prevailing issue of unstable GAN training. More investigation is needed to find explanations and quantify the performance variance.

4 Visualization of discriminator outputs

The LSGAN variant of our PatchGAN aims to output scores close to 1 for real data (F4 sim) and 0 for fake data (cGAN's F4 prediction). The PatchGAN outputs a $n \times n$ matrix, where each entry is the discriminator score for a specific $N \times N$ patch of the original image. Here, we visualize the $n \times n$ discriminator output; here, $n = 30$. We select the PatchGAN corresponding to the optimal generator model selected from the description in Sect. 2.2. It is then applied to the F4 sim, and cGAN prediction maps from Fig. 1, and the output matrices are compared in Fig. 3. As seen, output scores for F4 sim are closer to one, whereas output scores for F4 cGAN are closer to zero, which indicates that the PatchGAN outputs are non-random and a strong indication that it has forced the generator to emulate plausible results. Additionally, each entry in the matrix corresponds to a patch in the image, and the figure shows that the output scores have some spatial correlation, which can be explored further to gain more insights.

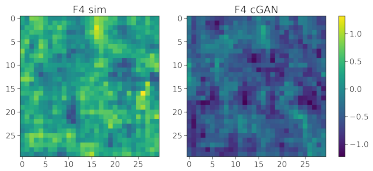

Figure 3: PatchGAN outputs for an example real and fake image.

5 Conclusion and future work

We modified the traditional `pix2pix` cGAN by (a) incorporating a frequency-based loss function with the traditional L1 loss used with the cGAN loss and (b) using a new mapping where the image from the input domain is mapped to the pixel-wise ratio of the images from the target and the input domain, instead of the target domain directly. Due to our modifications, the cGAN predictions successfully reproduced key expected features in the images from the target domain (in our case, modifications to Einstein's gravity) despite the minor differences between the input and target domains. A difference of $\sim 20 - 25\%$ in the power spectrum of the density fields between GR and $f(R)$ was reduced to $\sim 2\%$ using our cGAN. The density histograms revealed that the cGAN well reproduces the densities in $f(R)$ for all ranges of densities except for a few regions with extremely low or high densities. Since L1 loss weights all pixels equally, future work could involve using a weighted L1 loss measure or its equivalent in the frequency domain for improved synthesizing capabilities, applying the model on different $f(R)$ gravity models, and testing model generalization to different resolutions and redshifts. We also aim to trace back discriminator outputs to patches in the density maps to interpret the learning process of the discriminator.

This work presents a step towards replacing computationally expensive simulations (such as MG simulations) with deep-learning-based emulators to eliminate bottlenecks.

Acknowledgments and Disclosure of Funding

The authors acknowledge insightful discussions with Cheng-Zong Ruan, Baojiu Li, and Carolina Cuesta-Lazaro. SB is supported by the UK Research and Innovation (UKRI) Future Leaders Fellowship [grant number MR/V023381/1].

[1]We have used the `ptflops` package, available at `https://github.com/sovrasov/flops-counter.pytorch`

References

[1] Sean M. Carroll, Antonio de Felice, Vikram Duvvuri, Damien A. Easson, Mark Trodden, and Michael S. Turner. Cosmology of generalized modified gravity models. *Physical Review D*, 71 (6):063513, March 2005. doi: 10.1103/PhysRevD.71.063513.

[2] Matteo Cataneo, Cora Uhlemann, Christian Arnold, Alex Gough, Baojiu Li, and Catherine Heymans. The matter density PDF for modified gravity and dark energy with Large Deviations Theory. *Monthly Notices of the RAS*, 513(2):1623–1641, June 2022. doi: 10.1093/mnras/stac904.

[3] Marius C. Cautun and Rien van de Weygaert. The DTFE public software: The Delaunay Tessellation Field Estimator code. *arXiv e-prints*, art. arXiv:1105.0370, May 2011.

[4] Jonathan Chardin, Grégoire Uhlrich, Dominique Aubert, Nicolas Deparis, Nicolas Gillet, Pierre Ocvirk, and Joseph Lewis. A deep learning model to emulate simulations of cosmic reionization. *Monthly Notices of the RAS*, 490(1):1055–1065, November 2019. doi: 10.1093/mnras/stz2605.

[5] Timothy Clifton, Pedro G. Ferreira, Antonio Padilla, and Constantinos Skordis. Modified gravity and cosmology. *Physics Reports*, 513(1):1–189, March 2012. doi: 10.1016/j.physrep.2012.01.001.

[6] Antonino Del Popolo and Morgan Le Delliou. Small Scale Problems of the ΛCDM Model: A Short Review. *Galaxies*, 5(1):17, February 2017. doi: 10.3390/galaxies5010017.

[7] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014. URL `https://proceedings.neurips.cc/paper/2014/file/5ca3e9b122f61f8f06494c97b1afccf3-Paper.pdf`.

[8] César Hernández-Aguayo, Cheng-Zong Ruan, Baojiu Li, Christian Arnold, Carlton M. Baugh, Anatoly Klypin, and Francisco Prada. Fast full n-body simulations of generic modified gravity: derivative coupling models. *Journal of Cosmology and Astroparticle Physics*, 2022(01):048, jan 2022. doi: 10.1088/1475-7516/2022/01/048. URL `https://dx.doi.org/10.1088/1475-7516/2022/01/048`.

[9] Wayne Hu and Ignacy Sawicki. Models of $f(r)$ cosmic acceleration that evade solar system tests. *Phys. Rev. D*, 76:064004, Sep 2007. doi: 10.1103/PhysRevD.76.064004. URL `https://link.aps.org/doi/10.1103/PhysRevD.76.064004`.

[10] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, 2017.

[11] Liming Jiang, Bo Dai, Wayne Wu, and Chen Change Loy. Focal frequency loss for image reconstruction and synthesis. In *ICCV*, 2021.

[12] Anatoly Klypin and Francisco Prada. Dark matter statistics for large galaxy catalogues: power spectra and covariance matrices. *Monthly Notices of the Royal Astronomical Society*, 478 (4):4602–4621, 06 2018. ISSN 0035-8711. doi: 10.1093/mnras/sty1340. URL `https://doi.org/10.1093/mnras/sty1340`.

[13] Baojiu Li, Wojciech A. Hellwing, Kazuya Koyama, Gong-Bo Zhao, Elise Jennings, and Carlton M. Baugh. The non-linear matter and velocity power spectra in f(R) gravity. *Monthly Notices of the Royal Astronomical Society*, 428(1):743–755, 10 2012. ISSN 0035-8711. doi: 10.1093/mnras/sts072. URL `https://doi.org/10.1093/mnras/sts072`.

[14] Xudong Mao, Qing Li, Haoran Xie, Raymond Y. K. Lau, Zhen Wang, and Stephen Paul Smolley. Least Squares Generative Adversarial Networks. *arXiv e-prints*, art. arXiv:1611.04076, November 2016. doi: 10.48550/arXiv.1611.04076.

[15] Mehdi Mirza and Simon Osindero. Conditional Generative Adversarial Nets. *arXiv e-prints*, art. arXiv:1411.1784, November 2014.

[16] Mustafa Mustafa, Deborah Bard, Wahid Bhimji, Zarija Lukić, Rami Al-Rfou, and Jan M. Kratochvil. CosmoGAN: creating high-fidelity weak lensing convergence maps using Generative Adversarial Networks. *Computational Astrophysics and Cosmology*, 6(1):1, May 2019. doi: 10.1186/s40668-019-0029-9.

[17] Will J. Percival, Will Sutherland, John A. Peacock, Carlton M. Baugh, Joss Bland-Hawthorn, Terry Bridges, Russell Cannon, Shaun Cole, Matthew Colless, Chris Collins, Warrick Couch, Gavin Dalton, Roberto De Propris, Simon P. Driver, George Efstathiou, Richard S. Ellis, Carlos S. Frenk, Karl Glazebrook, Carole Jackson, Ofer Lahav, Ian Lewis, Stuart Lumsden, Steve Maddox, Stephen Moody, Peder Norberg, Bruce A. Peterson, and Keith Taylor. Parameter constraints for flat cosmologies from cosmic microwave background and 2dFGRS power spectra. *Monthly Notices of the RAS*, 337(3):1068–1080, December 2002. doi: 10.1046/j.1365-8711.2002.06001.x.

[18] L. Perivolaropoulos and F. Skara. Challenges for ΛCDM: An update. *New Astronomy Review*, 95:101659, December 2022. doi: 10.1016/j.newar.2022.101659.

[19] Nathanaël Perraudin, Ankit Srivastava, Aurelien Lucchi, Tomasz Kacprzak, Thomas Hofmann, and Alexandre Réfrégier. Cosmological N-body simulations: a challenge for scalable generative models. *Computational Astrophysics and Cosmology*, 6(1):5, December 2019. doi: 10.1186/s40668-019-0032-1.

[20] Dezső Ribli, Bálint Ármin Pataki, José Manuel Zorrilla Matilla, Daniel Hsu, Zoltán Haiman, and István Csabai. Weak lensing cosmology with convolutional neural networks on noisy data. *Monthly Notices of the Royal Astronomical Society*, 490(2):1843–1860, 09 2019. ISSN 0035-8711. doi: 10.1093/mnras/stz2610. URL `https://doi.org/10.1093/mnras/stz2610`.

[21] Adam G. Riess, Alexei V. Filippenko, Peter Challis, Alejandro Clocchiatti, Alan Diercks, Peter M. Garnavich, Ron L. Gilliland, Craig J. Hogan, Saurabh Jha, Robert P. Kirshner, B. Leibundgut, M. M. Phillips, David Reiss, Brian P. Schmidt, Robert A. Schommer, R. Chris Smith, J. Spyromilio, Christopher Stubbs, Nicholas B. Suntzeff, and John Tonry. Observational evidence from supernovae for an accelerating universe and a cosmological constant. *The Astronomical Journal*, 116(3):1009, sep 1998. doi: 10.1086/300499. URL `https://dx.doi.org/10.1086/300499`.

[22] Andrés Rodríguez, Tomasz Kacprzak, Aurelien Lucchi, Adam Amara, Raphael Sgier, Janis Fluri, Thomas Hofmann, and Alexandre Réfrégier. Fast cosmic web simulations with generative adversarial networks. *Computational Astrophysics and Cosmology*, 5, 11 2018. doi: 10.1186/s40668-018-0026-4.

[23] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation, 2015. URL `https://arxiv.org/abs/1505.04597`.

[24] D. N. Spergel, L. Verde, H. V. Peiris, E. Komatsu, M. R. Nolta, C. L. Bennett, M. Halpern, G. Hinshaw, N. Jarosik, A. Kogut, M. Limon, S. S. Meyer, L. Page, G. S. Tucker, J. L. Weiland, E. Wollack, and E. L. Wright. First-year wilkinson microwave anisotropy probe (wmap)* observations: Determination of cosmological parameters. *The Astrophysical Journal Supplement Series*, 148(1):175, sep 2003. doi: 10.1086/377226. URL `https://dx.doi.org/10.1086/377226`.

[25] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance normalization: The missing ingredient for fast stylization, 2016. URL `https://arxiv.org/abs/1607.08022`.

[26] Francisco Villaescusa-Navarro. Pylians: Python libraries for the analysis of numerical simulations. Astrophysics Source Code Library, record ascl:1811.008, November 2018.