Cosmological matter density field emulator with CVAE

Marta Corioni, Luigi Ernesto Ghezzer

June 2024

Abstract

The analysis and research of structures in the universe often require the quick generation of mock data. Traditional emulation methods, which approximate the behavior of complex simulations using simplified models, are being integrated with machine learning techniques in cosmology. In this study, we investigate the use of Variational Autoencoders for generating cosmological matter density fields. The code for this research is available at:
`https://github.com/luigiEG/FYS5429_Project2/Slices_Main`

Contents

1 Introduction

N-body simulations are powerful tools used in precision cosmology to study the evolution of large-scale structures in the universe. These simulations involve tracking the motion of a large number of particles over time under the influence of gravity (and possibly other forces). Through these simulations, it is possible to investigate the complex process of structure formation in detail.
However, running these simulations can be computationally demanding and time-consuming. Large-scale simulations may take several days or even weeks to complete, requiring significant computational resources. To address these challenges, several emulation techniques have been developed. These methods aim to approximate the behavior of complex simulations using simplified models.

Machine learning techniques have gained attention in cosmology for their ability to replace traditional emulation methods. Deep learning algorithms can accurately predict the formation of non-linear cosmic structures, providing results that are comparable or even superior to traditional methods. One notable example is the use of generative models.
Generative models are machine learning models that are designed to learn and replicate the underlying data distribution of a given dataset. In other words, they learn the patterns, structures, and correlations present in the data and then generate new samples that closely resemble the original data.
Generative models are primarily used for tasks such as data synthesis, data augmentation, and sample generation. They enable the creation of new data samples that exhibit the same statistical properties as the original data.
These techniques offer a simpler approach to approximating the behavior of N-body simulations, reducing the computational burden while still producing accurate results. This is particularly crucial in upcoming surveys like Euclid, where the quick generation of mock data is essential for analysis and understanding of large-scale structures in the universe.
This kind of approach has already been explored in [7] where a GAN was implemented to emulate simulation in the production of density map.
In this work, we will focus instead on Variational Autoencoders (VAEs), which are another specific type of generative model. In particular, we will employ a VAE to generate density maps of the large-scale structures in the universe. To generate the density maps, we will train the VAE on a dataset of existing density maps derived from N-body simulations.

To evaluate the accuracy of the generated density maps from the VAE, we will employ some qualitative and quantitative methods: visual inspection, comparison of the power spectrum, and the independence of generated images.
Firstly, visual inspection involves visually examining the generated density maps and comparing them with the original density maps from the N-body simulations. We can assess whether the VAE has captured the main structural features and patterns observed in the real data. This visual comparison can provide an initial indication of the quality and fidelity of the generated density maps.

Secondly, we will compare the power spectra of the generated density maps with the power spectra of the original density maps. It provides valuable information about the underlying structures and their spatial distribution. The power spectrum in astrophysics describes the density contrast of the universe as a function of scale. It is important for astrophysics because it helps in understanding the structures of the universe over a broad range of scales, probing cosmological parameters, testing cosmological models, and informing observational strategies for cosmological surveys and experiments.
By comparing the power spectra, we can quantitatively assess how well the VAE has captured the physical and statistical properties of the density maps. Specifically, we can examine whether the power spectra of the generated maps match the power spectra of the original maps, indicating the preservation of structures and relevant characteristics.
Lastly, we propose a method to measure the independence of the generated images, which is a fundamental factor in the data augmentation/sample generation task.

This combination of visual inspection and power spectrum and independence comparison allows us to evaluate both the visual fidelity and statistical accuracy of the generated density maps from the VAE. These evaluations help validate the effectiveness of the VAE and provide confidence in its ability to generate realistic density maps that accurately represent the large-scale structures in the universe.

2 Data

To obtain the density maps, we conducted N-body simulations using the COLA algorithm, presented in [3]. The COLA algorithm is a computational technique that allows us to model the evolution of structures in the universe.
In a cosmological simulation, we treat the universe as a collisionless fluid. This means that we ignore any particle interactions and only take into account the geodesic equation for motion and the gravitational field's influence on the momentum of all physical particles in the fluid. This is done by solving the Poisson equation.

N-body simulations simplify this process by using a relatively small number of "tracer" particles. These particles do not represent individual physical particles, but instead, they represent a population of particles that follow the same phase-space trajectory as determined by the distribution function.
In the early universe, where perturbations are small, the evolution of perturbations can be computed analytically using linear approximation. However, as we move towards studying the late universe, simulations become necessary as perturbations are no longer in the linear regime and require more complex nu-

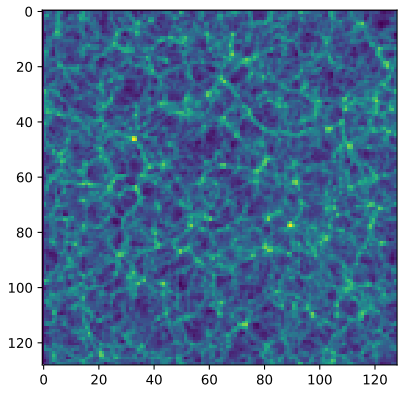

Figure 1: Example of a density map from the test database (colors in logarithmic scale)

merical methods to accurately model their evolution.

In this scenario, we are utilizing a gravity model for the simulation that incorporates the time-dependent parameterization of standard gravity. In Fourier space, the Poisson equation can be adapted as follows:

$$-k^2\Psi \equiv 4\pi G a^2 \mu(a)\rho\Delta \tag{1}$$

The primary motivation behind constructing modified gravity models is to address phenomena such as dark energy. Therefore, it seems natural to model the time evolution to scale with the dark energy density.

$$\mu(a) = 1 + \mu_0 \frac{\Omega_\Lambda(a)}{\Omega_\Lambda} \tag{2}$$

μ_0 is then the parameter describing the strength of the modification to standard gravity. In our case, we chose to use $\mu_0 = 0.5$ and fiducial values for all other cosmological parameters.

The density field maps are created using a modification of COLA simulations with a specific μ_0 value, evaluated at redshift 0.
The CIC (Cloud-In-Cell) interpolation method is employed to calculate the 2D density distribution from the particle positions over a slice of the simulation box. This method, presented in [1], involves spreading each particle's density across nearby grid cells based on its position and using linear interpolation to distribute density contributions to the surrounding cells.
We generate 128 slices across the simulation box. Each slice represents a two-dimensional plane within the three-dimensional volume. For each slice, we evaluate the density maps with a resolution of 128x128.

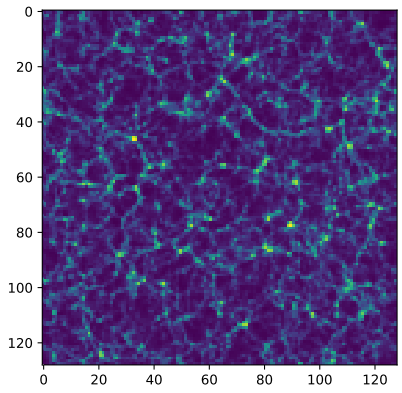

Figure 2: Same density maps as in Fig. 1
but with higher contrast using function **** REF , and normalization. Note that the colors are not plotted on a logarithmic scale, if we were to do it the contrast would be even more pronounced, but this is effectively the images that were fed to the networks.

In order to highlight the characteristics of the large-scale structure, we implemented the same nonlinear transformation as described in [4]. This transformation, denoted as $s(s)$, is expressed as

$$s(x) = \frac{2x}{(x+a)} - 1 \tag{3}$$

By applying this transformation with a value of $a = 8$, the overdensity values are rescaled to a range of $[1, 1]$, and the contrast of the resulting images is increased. This allows for a clearer visualization and analysis of the large-scale structure, as shown in fig 2. The data is then normalized before being fed to the model.

3 VAE

VAE can be seen also as a generative method in ML. This method is derived from Auto Encoders. We will briefly describe AE architecture and training method, then we will describe VAE.

AE consists of two main components: an encoder and a decoder. The encoder takes in the input and compresses it into a low-dimensional representation called the latent space. The encoder intuitively learns a map from a high-dimensional space to a lower-dimensional space where most of the features of the input should still be represented. A decoder then takes the points in latent space and

reconstructs them into full-size input. The network is trained to minimize the reconstruction loss between the reconstructed input and the real one.

The idea behind VAE is to regularize the encoder map. Without any regularization method, the encoder map will be sparse, in the sense that between points that represent real input, there will be points in the latent space that do not represent any realistic input. This is because in general, the net will try to "cheat" and place latent representation points far away from each other to make the decoder work easier.
In a VAE the encoder does not output a point in the latent space, but a distribution. A popular choice is a Gaussian distribution and so from an input, the encoder gives the parameters of a Gaussian distribution in the latent space. After this, a point will be sampled from this distribution and fed to the decoder.

$$E : \mathbf{x} \to (\boldsymbol{\mu}, \boldsymbol{\sigma}) \tag{4}$$

$$\boldsymbol{z} \sim N(\boldsymbol{\mu}, \boldsymbol{\sigma}) \tag{5}$$

$$D : \mathbf{z} \to \boldsymbol{x}^* \tag{6}$$

In addition to the reconstruction error, a regularization term is added. This term gives a penalty to distributions that are far away from a standard normal distribution as $N(0, 1)$.

3.1 Reparametrization trick

To sample point in the latent distribution the reparametrization trick is used. This ensures that the output is still completely differentiable along the parameters and one can use backpropagation to train the model.

From the graph, we can see that:

$$\mathbf{z} = \boldsymbol{\mu} + \boldsymbol{\sigma}\epsilon \tag{7}$$

$$\epsilon \sim N(0, 1) \tag{8}$$

Then it holds:

$$\mathbf{z} \sim N(\boldsymbol{\mu}, \boldsymbol{\sigma}) \tag{9}$$

So $\mathbf{z}$ is distributed following the encoder prescription but one can still differentiate.

3.2 Loss function

The theoretical framework to explain VAE loss function is probabilistic inference with latent variables. The hypothesis is that $\mathbf{x}$, for us a density distribution map, is a realization of a random variable X of unknown distribution. Another latent random variable is used, $\mathbf{z}$ realization of Z and we model the joint probability distribution as $p(\mathbf{x}, \mathbf{z}; \theta)$ where θ are parameters to be optimized. In a VAE $\mathbf{z}$ is a realization of the latent distribution mapped by the encoder.

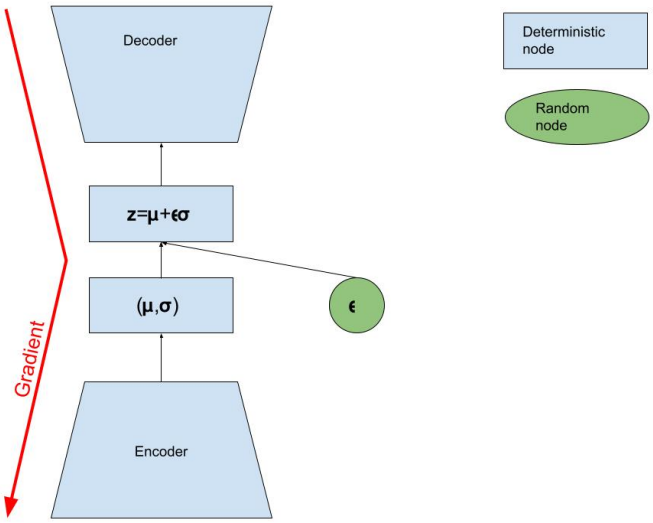

Figure 3: Illustration of the reparametrization trick, even if a random node is present, the gradient can be calculated along the model.

What one wants to maximize is the evidence. The evidence is the probability of observing $\mathbf{x}$ given our model $p(\mathbf{x}, \mathbf{z}; \theta)$. Intuitively if our parameters and model are correct, no other probability distribution can give higher evidence on a representative sample of $\boldsymbol{x}$. If we know the probability distribution $q(\boldsymbol{z})$, for us a Gaussian distribution, then one can show that marginalizing $\boldsymbol{x}$ from the joint distribution one gets:

$$\log p(x; \theta) \geq E_{z \sim q}\Big[\log \frac{p(x, Z; \theta)}{q(Z)}\Big] \tag{10}$$

The right hand of the equation is called evidence lower bound, ELBO. In our specific case, with latent variables Gaussian distributed, it is possible to demonstrate [8] that maximizing the ELBO is equivalent to minimizing:

$$L_{VAE} = L_{reco} + L_{KL} \tag{11}$$

Where L_{reco} is a loss function that measures how distant is the reconstructed input to the original one while L_{KL} is the Kullback-Leibler divergence that measures the overlap of the latent distribution with a normal distribution $N(0, 1)$. In the Gaussian case, this latter takes a simple analytical form that depends on $(\boldsymbol{\mu}, \boldsymbol{\sigma})$.

Intuitively we can say that during the training process, the VAE learns to minimize the reconstruction error, ensuring that the generated density maps closely resemble the original ones. Additionally, the VAE incorporates a regularization

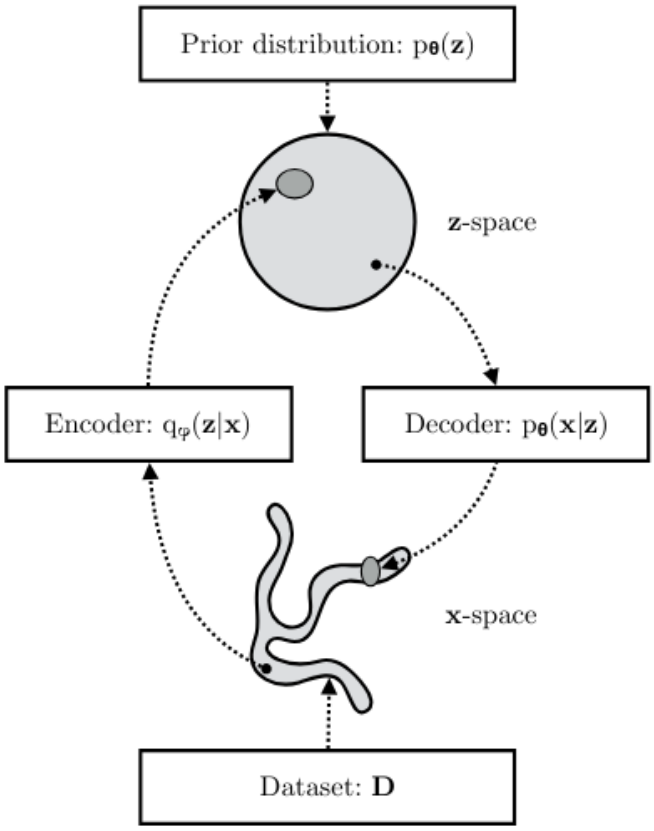

Figure 4: The map from the $\boldsymbol{x}$ and $\boldsymbol{z}$ performed by a VAE. Image from [8].

term that encourages the latent space to follow a specific distribution. This regularization term promotes a "dense" latent space where it is easier to generate realistic density maps.

3.3 CVAE

When dealing with images as input, convolutional layers boost the performance of different kinds of architecture. In other related work as [6] VAE contains convolutional layers in the encoder and transpose convolutional layers in the decoder. This kind of network is referred to as CVAE. Our architectures are inspired by this work, but we will take an auto-ML approach using a model where the architecture will be fine-tuned.

The main hyperparameters of the model are:

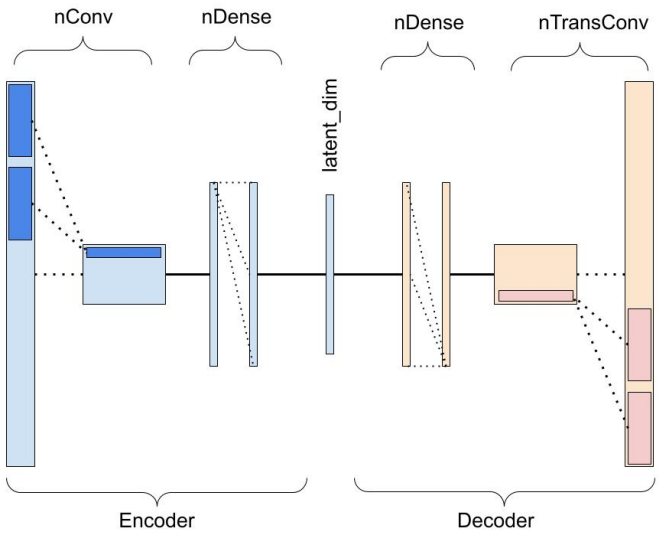

Figure 5: Representation of our CVAE implementation. The reparametrization trick is not shown.

- `nConv`: number of convolutional layers and transpose convolutional layers in the encoder and decoder. This can be 2 or 3, and the number of kernels are respectively (32,64,128). Kernel dimension, stride and padding are fixed to (4x4, 1, 2) respectively.
- `nDense`: after convolutional layers, the tensor is flattened and fed to an MLP with `nDense` layers with an equal number of `nHidden` units.
- `latent_dim`: dimension of the latent space.

The tuning process is discussed in the methods section.

4 Methods

4.1 Training

The goal of this work is to train a CVAE as an emulator of density maps. We start from a dataset of 128 density maps obtained with the simulation obtained with fixed cosmological parameters. This dataset is divided into train and test. As a base model, we will use a CVAE with 2 convolutional layers and 1 dense layer with a 2-dimensional latent space. We trained the model for 3k epochs with a batch size of 24. We will refer to this model as the vanilla one. Code can be found in `Slices_Main/Emulator/Emulator_vanilla.ipynb`

We will also explore other architectures performing tuning on hyperparameters presented in the VAE section. We take a sample of 24 possible hyperparameter configurations and we select the best using the asynchronous hyperband algorithm [9], that we already discussed in the previous project, using `ray.tune` [11].
The best configuration will be trained on a bigger dataset of 258 images with the same cosmological parameters. Lastly, the tuned model will be trained for 5k epochs. Code can be found in `Slices_Main/Emulator/Emulator_tuning.ipynb`

	`epochs`	`dataset size`	`nConv`	`nDense`	`latent_dim`	`n_hidden`
Vanilla	$3 \cdot 10^3$	128	2	1	2	2
Tuned	$5 \cdot 10^3$	256	3	2	6	40

Table 1: Hyperparameters configuration of Vanilla and Tuned model.

4.2 Generative method

Once the VAE is trained, we can sample points from the latent space and pass them through the decoder to generate new density maps. The promise is that those generated density maps provide a quick and efficient way to produce mock data for further cosmological analyses.

Different methods have been explored for sampling in a relevant way the latent space to use VAE as generative models [10]. We will briefly recall some of them:

- **parametric**: modeling the latent distribution with a known distribution as a Gaussian. This method can be seen as exploiting the fact that during optimization distributions far from Gaussian ones are discouraged.
- **non-parametric** where the distribution is modelized using Kernel-Density-Esitmation. This approach can be more flexible since it is able to model more complex distribution.
- **perturbing data** sampling new latent points starting from the latent representation of train images and perturbing their position in the latent space.

All these methods have their advantages and disadvantages.
In our case, we will use another method, that is not meant to be a conclusive one but is fast and does not need any new code. The idea is to use part of the dataset, the test part, to produce points in the latent space for generating images. This is not cheating because new data that the model didn't see during the training has

to come from the same distribution as the train data. We simply circumnavigate the problem of sampling a possible complex distribution scarifying some data during the train to be used to sample the latent distribution.

4.3 Observables

To check the accuracy of the images generated by the VAE, it is essential to extract a physical quantity from these images that can be directly compared with the corresponding quantity in the original images.

4.3.1 Power spectrum

One of the most important statistics in the study of the structure of the universe is the two-point correlation function. For large-scale structure studies, the inhomogeneities in the distribution of galaxies can be characterized by the function $\delta(\boldsymbol{x}) = (n(\boldsymbol{x}) - \bar{n})/\bar{n}$, where $n(\boldsymbol{x})$ represents the local number density of galaxies and $\bar{n}$ is the mean density. This function quantifies the deviation of the local density from the average density.

Since practicality often calls for a translation of the function into Fourier space, we can obtain $\tilde{\delta}(\boldsymbol{k})$ by taking the Fourier transform of $\delta(\boldsymbol{x})$, with $\boldsymbol{k}$ denoting the wave vector in Fourier space. This Fourier transform provides a representation of the density perturbations across various scales.

Lastly, the power spectrum $P(k)$ is defined through the equation from [2],

$$\langle \tilde{\delta}(\boldsymbol{k})\tilde{\delta}(\boldsymbol{k'})\rangle = (2\pi)^3 \delta_D^3(\boldsymbol{k} - \boldsymbol{k'})P(k), \tag{12}$$

where $\langle\rangle$ denotes the ensemble average and δ_D^3 is the 3-dimensional Dirac delta function. This equation relates the statistical properties of the density perturbations in Fourier space to the power spectrum $P(k)$, which provides information about the amplitude and distribution of the perturbations across different spatial scales.

As part of our analysis, we will extract the power spectrum from both the original and generated images. Ideally, we expect to find a similar shape in the power spectra of the original and generated images. A similar shape would indicate that the generated images accurately capture the statistical properties and spatial correlations present in the original images. This consistency would validate the realism and accuracy of the VAE-generated images in representing the underlying physical structure.

By comparing the power spectra, we can quantitatively assess the degree of similarity between the original and generated images, giving us valuable insights into the performance and quality of the VAE model.

We should specify that the power spectra may vary slightly for the different slices, and it is important to account for this variability in our analysis. We should not expect the power spectra of the generated images to precisely match those of the original data. The purpose of the analysis is to compare and check that the overall shape of the power spectra remains consistent. This can help in

assessing the structural features of the large-scale structure and confirming the presence of similar patterns, despite the slight variability in the power spectra.

4.3.2 Independence

If we want to use our emulator for data augmentation we need new independent images. To have an emulator that generates images that are identical to the training dataset is useless. If the generator instead generates all identical images again it is useless.

Assessing the level of independence between two images is not a straightforward task. One approach to evaluate the dissimilarity of new images is by studying the Mean Squared Error (MSE) between pairs of images. Intuitively, a high mean MSE between pairs in a dataset indicates that the images in the dataset are distant in the image space and thus more independent of each other. This means that there is a greater variation or dissimilarity between the images, suggesting that each image contains unique information and does not exhibit strong correlations or similarities with other images in the dataset. We will compare the MSE distribution between pairs of images in the original dataset with the distribution of pair MSE of generated images. We will not directly compare the generated images with the original ones because we observe that the power spectrum is not the same. In this case, comparing the MSE between the original and generated images may not be meaningful. The emulator is producing images that may not be perfectly physical, so a high MSE between the original and generated images does not necessarily indicate that the images are independent. It could simply mean that the overall appearance or global characteristics of the generated images are different from the original ones. Therefore, the power spectrum alone may not be sufficient to evaluate the independence or similarity between the original and generated images.
Comparing the pair-wise MSE of generated images with the pair-wise MSE of original images can provide insights into the independence of the generated images from each other and their comparison with the original images.

5 Results and analysis

As previously mentioned, we examine the outcomes generated by the vanilla model and the tuned model. Our analysis focuses on comparing these outputs with the corresponding input samples. We will start with some qualitative comments on the reconstructed image by visual inspection.

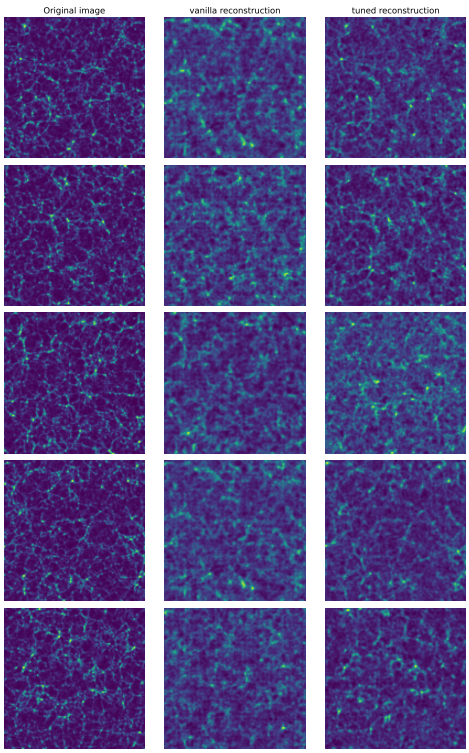

Figure 6: Comparison between original and reconstructed density fields, each row is a different test image, first column the original one, second the vanilla and third tuned reconstructed.

From fig 6 it is clear that the images generated by the tuned emulator more closely resemble the original test images, on average. In comparison, the images produced by the vanilla emulator appear blurred and do not accurately reproduce the structures of the original image, despite displaying similar patterns. The improvement in image resemblance of the tuned emulator may be attributed to the inclusion of additional convolutional layers.
Convolutional layers are responsible for detecting and capturing spatial patterns in the input images. By adding more of these layers, the tuned emulator is able to extract more intricate features and details from the original test images. However, it is important to note that adding more convolutional layers comes with the cost of increased computational resources. The additional layers require more parameters to be trained, resulting in a higher computational burden during the training and inference phases. This can lead to increased training time and potentially require more powerful hardware to handle the increased workload.

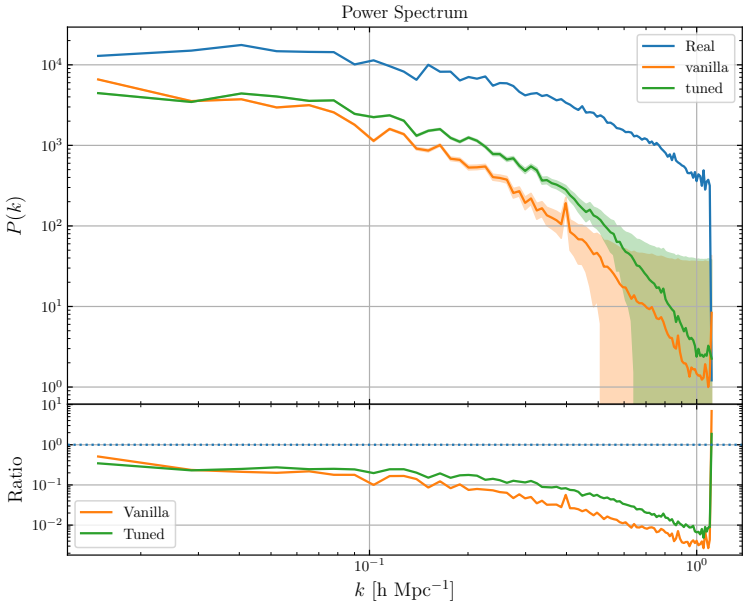

Figure 7: Above: Average power spectra of the density distributions generated by the vanilla and tuned models compared to the original test dataset. The error bars represent the mean standard deviations. Below: Ratio of the power spectra from the models to the real one.

To conduct a more comprehensive evaluation, we quantified the average Power Spectrum and relative error by analyzing the test dataset using both the vanilla structure model and the tuned Variational Autoencoder (VAE), and comparing them with the original data. The results of this analysis are presented in Figure 7.

Upon observing the results, several interesting patterns emerged. The spatial frequencies in this context refer to the different scales of the perturbations within the density field. Low wavenumber k represents large-scale variations, while high wave number k corresponds to smaller-scale details and variations. When the power spectrum is reduced for all wave numbers, as in this case, it indicates that the amplitude of perturbations at all spatial frequencies is diminished. This can imply a loss of detail and structure in the density field. A decreased power spectrum across all wave numbers indicates that the overall intensity of the perturbations, regardless of their scale, is lower in the generated images. The perturbations within the density field are not fully captured or adequately represented, leading to a loss of fidelity and detail in the result-

ing images. This could indicate a lack of ability of the emulator to faithfully represent the complex structures and variations within the density field, potentially resulting in blurriness and a lack of fine-grained detail in the generated images. Specifically, for large scales ($k < 0.1$), the ratio between the predicted and actual values remained within 0.1, indicating that the models were able to provide relatively accurate analyses. However, when we examined smaller scales ($k > 0.1$), the accuracy of the models' predictions decreased even further.

In recent studies, such as [5], a consistent limitation has been observed in CNN-generated images in accurately capturing the decay attributes of high-frequency Fourier spectra. The power spectrum, which involves the Fourier transform, is susceptible to the limitations associated with replicating the decay attributes of the high-frequency Fourier spectrum.
The exact reason for the discrepancy is still a topic of debate. Various hypotheses have been proposed to explain this issue. One possible reason could be the use of transposed convolutions in the generator architectures. Transposed convolutions are commonly used in the generation process, and completely replacing them can be challenging. Another explanation could be attributed to the linear dependencies present in the spectrum of convolutional filters. These linear dependencies can impede the ability of CNNs to effectively learn and reproduce high-frequency information.
Furthermore, in addition to observing a decrease in precision at smaller scales, we also noticed that the relative error calculated over the dataset increased as the scale decreased. It is important to note that evaluating the power spectrum of a 2D density field becomes more challenging when the resolution is low. The models' ability to capture and analyze smaller-scale structures is inherently limited by the resolution of the data. Therefore, while the tuned VAE exhibited improvements over the vanilla structure in capturing these smaller-scale structures, there is room for further enhancement.
Lastly, we would like to include the mean square distance of the power spectrum for the two models compared to the test dataset. The MSE for the power spectrum of the tuned model is 2.17×10^7, and the MSE for the power spectrum of the Vanilla model is 2.27×10^7. We excluded the last values of k from the analysis due to their high fluctuations.

To assess the independence of the generated images, we analyze the pair-wise MSE distribution for the test outputs of the vanilla model, the tuned model, and the original test dataset. This will provide insights into the degree of independence between the original test dataset and the generated one.
Figure 8 displays the pair-wise MSE distribution, showcasing the range of MSE values obtained for each model. Based on the analysis of the mean pair-wise MSE values, it appears that the generated outputs have a smaller mean MSE compared to the original dataset. This suggests that the generated images are more correlated or similar to each other than the original dataset.

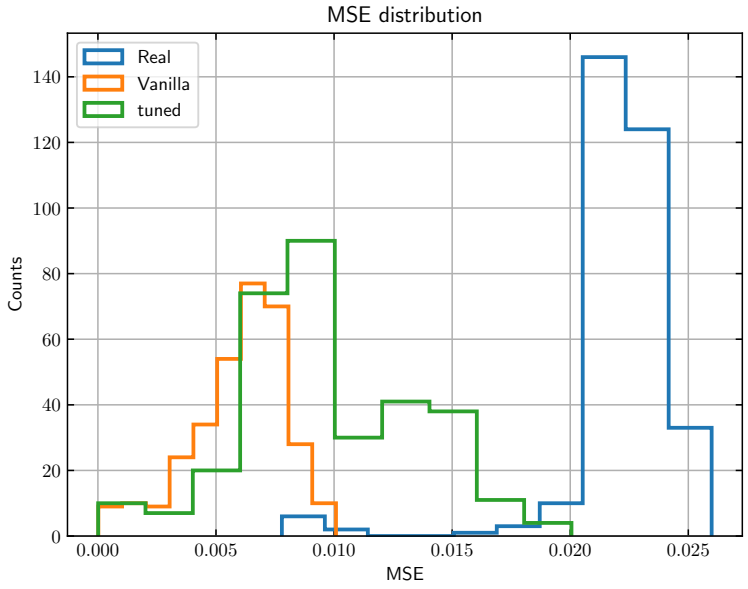

Figure 8: Distributions of pair-wise MSE across the generated images from the models and the original test dataset.

However, when considering the tuned model the mean value is closer to the original dataset and the pair-wise MSE distribution overlaps with that of the original dataset. This indicates that the tuned model is more capable of producing images that exhibits some level of diversity or variation.

Overall, these findings suggest that the generated outputs, including those

	Original	**Vanilla**	**Tuned**
$\langle \text{MSE} \rangle_{pair}$	$2 \cdot 10^{-3}$	$6 \cdot 10^{-4}$	$9 \cdot 10^{-4}$

from the tuned model, are not entirely independent from each other. There is a level of correlation or similarity present, indicating that the generated images retain certain common characteristics or patterns.

It is important to note that MSE alone may not capture all aspects of image similarity or independence, and other factors such as visual quality and perceptual differences should also be taken into consideration when evaluating the generated outputs.

6 Conclusions

Achieving a higher level of accuracy in a model often involves enhancing its capability to understand and capture the nuances of the data. A possibility is to integrate additional convolutional layers into the network. These layers are adept at extracting more complex patterns and features within images or spatial data, which can contribute significantly to the sophistication and accuracy of the model.

In addition to adding convolutional layers, another strategy for improvement is to expand the dataset and increase the number of training epochs. An enlarged dataset provides more examples for the model to learn from, enhancing its ability to generalize and perform well on unseen data. Meanwhile, a higher number of training epochs allows the model more opportunities to iterate over the data and refine its weights and biases to minimize error.

However, practical constraints such as computational resources can limit the extent to which it is possible to scale up the model training. For example, training larger models for extended periods, as seen in works such as [7], requires multiple GPUs and can span several days or weeks, which may not be feasible for everyone due to hardware limitations.

Moreover, increasing the resolution of the input data can significantly improve model performance by giving more detailed information for the model to learn from. High-resolution data can help the model in capturing finer details and subtle variations, which might be vital for tasks such as density field reconstruction.

To summarize, the refined VAE with its additional layers and expanded training regimen showed promise in identifying smaller-scale structures more effectively than the initial vanilla architecture. Nevertheless, there remains room for further refinement. Enhancing the model's architecture, training duration, and data quality—balanced with the available computational resources—will be key factors in driving improvements in the model's performance.

References

[1] S.E. Laux. "On particle-mesh coupling in Monte Carlo semiconductor device simulation". In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 15.10 (1996), pp. 1266–1277. DOI: 10.1109/43.541446.

[2] Scott Dodelson. *Modern Cosmology*. Academic Press, Elsevier Science, 2003.

[3] Svetlin Tassev, Matias Zaldarriaga, and Daniel J. Eisenstein. "Solving large scale structure in ten easy steps with COLA". In: *Journal of Cosmology and Astroparticle Physics* 2013.06 (June 2013), p. 036. DOI: 10.1088/1475-7516/2013/06/036. URL: https://dx.doi.org/10.1088/1475-7516/2013/06/036.

[4] Andres C. Rodríguez et al. "Fast cosmic web simulations with generative adversarial networks". In: *Computational Astrophysics and Cosmology* 5.1, 4 (Nov. 2018), p. 4. DOI: 10.1186/s40668-018-0026-4. arXiv: 1801.09070 [astro-ph.CO].

[5] Keshigeyan Chandrasegaran, Ngoc-Trung Tran, and Ngai-Man Cheung. *A Closer Look at Fourier Spectrum Discrepancies for CNN-generated Images Detection*. 2021. arXiv: 2103.17195 [cs.CV].

[6] "A Better Autoencoder for Image: Convolutional Autoencoder Yifei Zhang1[u6001933] Australian National University ACT 2601, AU". In: ().

[7] "CosmoGAN: creating high-fidelity weak lensing convergence maps using Generative Adversarial Networks Mustafa Mustafa, Deborah Bard, Wahid Bhimji, Zarija Lukić, Rami Al-Rfou, Jan M. Kratochvil". In: ().

[8] Max Welling Diederik P. Kingma. "An Introduction to Variational Autoencoders". In: ().

[9] "Hyperband: A Novel Bandit-Based Approach to Hyperparameter Optimization Lisha Li, Kevin Jamieson, Giulia DeSalvo, Afshin Rostamizadeh, Ameet Talwalkar". In: ().

[10] "Sampling from the latent space in Autoencoders: A simple way towards generative models? TMLR Paper960 Authors". In: ().

[11] "Tune: A Research Platform for Distributed Model Selection and Training Richard Liaw, Eric Liang, Robert Nishihara, Philipp Moritz, Joseph E. Gonzalez, Ion Stoica". In: ().