Prepared for submission to JCAP

The Universe is worth 64^3 pixels: Convolution Neural Network and Vision Transformers for Cosmology

Se Yeon Hwang,[a,b] **Cristiano G. Sabiu,**[1][a,b] **Inkyu Park,**[a,b]
Sungwook E. Hong[c,d]

[a]Department of Physics, University of Seoul, 163 Seoulsiripdae-ro, Dongdaemun-gu, Seoul 02504, Republic of Korea

[b]Natural Science Research Institute, University of Seoul, 163 Seoulsiripdae-ro, Dongdaemun-gu, Seoul 02504, Republic of Korea

[c]Korea Astronomy and Space Science Institute, 776 Daedeok-daero, Yuseong-gu, Daejeon 34055, Republic of Korea

[d]Astronomy Campus, University of Science and Technology, 776 Daedeok-daero, Yuseong-gu, Daejeon 34055, Republic of Korea

E-mail: worldhsy@gmail.com, csabiu@uos.ac.kr, icpark@uos.ac.kr, swhong@kasi.re.kr

Abstract. We present a novel approach for estimating cosmological parameters, Ω_m, σ_8, w_0, and one derived parameter, S_8, from 3D lightcone data of dark matter halos in redshift space covering a sky area of 40° × 40° and redshift range of $0.3 < z < 0.8$, binned to 64^3 voxels. Using two deep learning algorithms—Convolutional Neural Network (CNN) and Vision Transformer (ViT)—we compare their performance with the standard two-point correlation (2pcf) function. Our results indicate that CNN yields the best performance, while ViT also demonstrates significant potential in predicting cosmological parameters. By combining the outcomes of Vision Transformer, Convolution Neural Network, and 2pcf, we achieved a substantial reduction in error compared to the 2pcf alone. To better understand the inner workings of the machine learning algorithms, we employed the Grad-CAM method to investigate the sources of essential information in heatmaps of the CNN and ViT. Our findings suggest that the algorithms focus on different parts of the density field and redshift depending on which parameter they are predicting. This proof-of-concept work paves the way for incorporating deep learning methods to estimate cosmological parameters from large-scale structures, potentially leading to tighter constraints and improved understanding of the Universe.

[1] Corresponding Author

Contents

1 Introduction

The large-scale structure (LSS) of the universe offers invaluable insights into the underlying cosmological model. The distribution of galaxies and dark matter in the LSS is governed by the initial conditions and cosmological parameters that dictate the evolution of the universe [1, 2]. Thus, comparison between observational and theoretical models or numerical simulations provides constraints on the components and physics that govern the Universe on the largest scales. Visually, the LSS of galaxies is a complex web-like distribution exhibiting over-dense “clusters”, near-empty “void” regions and intermediate density features known as “walls” and “filaments” [3]. These structures are also not static but change and evolve over cosmic time, or redshift, due to gravity and cosmic expansion. However, to compare observations with theoretical models, we typically condense this rich 3-dimensional galaxy distribution into a 2nd-order clustering statistic such as the power spectrum (PS) or correlation function (CF), although there are some efforts to move into higher order statistical descriptions of the LSS [4–6].

Measurements of the galaxy PS or CF have provided exquisite constraints on the expansion history of the Universe via the Baryon Acoustic Oscillations (BAO) [7–9] and the Alcock-Paczynski effect [10–16], and on the the growth of structure via Redshift Space Distortions (RSD) [17–21]. In many of these works, the observed statistic are compared with analytic models, templates or more numerical perturbation theory predictions. However, in an effort to reliably predict the clustering on small scales, cosmological N-body simulations have been employed [22–25]. Also, using a suite of simulations, one can build emulators predicting clustering statistics via efficient sampling and interpolation, for a range of cosmological models and parameters [26, 27]. Although numerically expensive, this approach has many advantages, including easy applications of observational systematics.

Literature	Data Type	Data Size	Prediction
[30]	Halo, Snapshot $(z = 0)$	64^3, $(128\,\mathrm{Mpc}/h)^3$	$\Omega_\mathrm{m}\,, \sigma_8$
[31]	Halo, Snapshot $(z = 0)$	128^3, $(256\,\mathrm{Mpc}/h)^3$	$\Omega_\mathrm{m}\,, \sigma_8, n_\mathrm{s}$
[32]	DM Density, Snapshot $(z = 0)$	32^3, $(64\,\mathrm{Mpc}/h)^3$	$\Omega_\mathrm{m}\,, \sigma_8$
[33]	DM Density, Snapshot $(z = 0)$	64^3, $(1\,\mathrm{Gpc}/h)^3$	$\Omega_\mathrm{m}\,, \sigma_8$
[34]	HOD, Snapshot $(z = 0.3)$	$275 \times 275 \times 55$, $550 \times 550 \times 220$ $(\mathrm{Mpc}/h)^3$	$\Omega_\mathrm{m}\,, \sigma_8$
Our Study	Halo, Lightcone $(0.3 < \mathrm{z} < 0.8)$	64^3, $(2\,\mathrm{Gpc}/h)^3$ $40^\circ \times 40^\circ$	$\Omega_\mathrm{m}\,, \sigma_8, w_0$

Table 1: Summary of the cosmological parameter predictions using 3D CNN with simulations described in the texts.

In recent years, machine learning techniques have demonstrated tremendous potential for extracting information from large and complex datasets in various scientific disciplines [28, 29]. In the context of cosmology, recently, Convolution Neural Networks (CNNs) have shown promise in predicting cosmological parameters from the LSS density field [30–34] and from 21cm temperature maps at higher redshift [35]. One advantage of this technique is that it allows us to directly use the distribution of matter without the need of somewhat processed statistics. In ref. [30], they first use a 3D CNN with N-body simulations to predict the matter density parameter Ω_m and the root-mean-square of the amplitude of matter perturbation at $8\,\mathrm{Mpc}/h$-scale σ_8. In ref. [31], they used a similar model structure as ref. [30] to predict Ω_m, σ_8 and the scalar spectral index of the primordial PS n_s. In ref. [32], they leveraged these pioneering works in a more lightweight approach with smaller input size to predict Ω_m and σ_8. In ref. [33], while they predicted Ω_m and σ_8 using a 3D CNN, they were also able to predict n_s by combining with PS. In ref. [34], by using mock galaxy simulations with various Halo Occupation Distributions (HOD), they predicted Ω_m and σ_8. Alternatively, cosmological parameters can also be predicted using the projection of 3D data, as demonstrated by ref. [36]. The examples presented here are summarized in table 1.

While CNNs have been widely used in astronomy, Vision Transformer (ViT) emerged from computer vision tasks [37], and the attention mechanism [38] underlying them was widely used in Natural Language processing. While CNN reduces the data size by half using a pooling layer after the convolution layer, ViT maintains the original data size. ViT divides the 2D or 3D image into small patches and flatten to 1D array to calculates scores indicating how much attention should be given between each patches. This approach allows us to identify the important patches and extract the desired features from the data. Furthermore, they are suitable for finding global relationships between each patch. Several studies have utilized ViT in the field of astronomy. In ref. [39], they used ViT to solve galaxy morphological classification with the Galaxy Zoo dataset. They found that, although the total accuracy means of CNN were higher than ViT, it performed better in classifying smaller and fainter galaxies than CNN. In ref. [40], galaxy classifications were also studied using Galaxy Zoo 2, Sloan Digital Sky Survey Data Release 17 (SDSS-DR17), and Galaxy Zoo Dark Energy Camera Legacy Survey (DECaLS). They found that using the combination of CNN and ViT had the best performance, followed by the CNN and the ViT model. In ref. [41], they predicted strong gravitational lensing parameters with ViT using 31,200 simulated strongly lensed quasar images, where the ViT model outperformed the CNN model except for one parameter. Lastly, ref. [42] tested the morphological classification of AI-augmented images of radio galaxies, where a Fully-Connected Neural Network (FCN) model outperforms both CNN and ViT. In some cases, ViT has competitive performances with CNN especially when they are pre-trained with large data set. On the other hand, in this paper, we apply ViT for the first time in predicting cosmological parameters using 3d simulations without pre-training and compare them with CNN results.

The remainder of this paper is organized as follows: Section 2 describes the data which is used to train and test our model and also the data which we used for making two point correlation functions. In Section 3, we introduce the machine learning models that we test in this work, both for Convolution Neural Network and Vision Transformer. In Section 4, we analysis the result from CNN, ViT and from the two-point correlation function. We discuss how we can interpret and understand the procedure of deep learning and which information was valuable when deep learning predicts the cosmological parameters. Lastly, we make conclusions in Section 5.

2 Data

2.1 Cosmological Parameter Sets

By assuming flat wCDM cosmologies, we vary five cosmological parameters within specified ranges: $\Omega_{\rm m} \in [0.25, 0.4]$, $\sigma_8 \in [0.4, 1.1]$, the equation-of-state of dark energy $w_0 \in [-1.5, -0.5]$, $n_{\rm s} \in [0.9, 1.1]$, and $h \in [0.6, 0.8]$. Then, we calculate the derived matter perturbation amplitude $S_8 \equiv \sigma_8(\Omega_{\rm m}/0.3)^{0.5}$. We adopt Latin Hypercube Sampling to select 994 distinct parameter sets for our simulation suite.

Our datasets can be categorized into two types. First, training (80% of total datasets) and test (20% of total datasets) datasets comprise different cosmological parameters and initial random seeds. Then, a systematic test set assesses the performance of the trained model on a single, fixed cosmological parameter set, where only the initial seeds are varied. For this systematic test set, we fix the cosmological parameters as $(\Omega_{\rm m}, \sigma_8, w_0, n_{\rm s}, h) = (0.3133, 0.8079, -1, 0.9649, 0.6736)$, which is in a concordance with the flat ΛCDM cosmology from Planck 2018 [43], and generate 100 single cosmology datasets.

2.2 Mock Lightcone Halo Catalogs

In predicting cosmological parameters, it is crucial to consider the application of observational data. The ultimate goal is to predict more accurate parameters from observational data. As summarized in Table 1, our study's distinguishing feature is the use of lightcone data with a broad redshift range. This enabled us to try predicting the dark energy parameter w_0 for the first time using the 3D CNN method.

For each cosmological parameter set from the above subsection, we employ the PINpointing Orbit Crossing Collapsed HIerarchical Objects (PINOCCHIO) algorithm [44] to generate a mock lightcone dark matter (DM) halo catalogs. PINOCCHIO uses Lagrangian Perturbation Theory (LPT) to produce DM halo catalogs containing halo mass, position, and velocity information. Each simulation was performed with a comoving box size of $2\,{\rm Gpc}/h$ and 1024^3 DM particles. Then we utilize the automatic past lightcone data generation of the PINOCCHIO code, which assigns the right ascension (RA), declination (DEC), and observed redshift (z)

for each DM halo. We set the output range of RA, DEC, and redshift to have a complete past lightcone halo catalog within $-20° < \mathrm{RA}, \mathrm{DEC} < +20°$ and $0.3 < z < 0.8$.

The halo mass function strongly depends on cosmological parameters [45–48]. Consequently, the number of halos in each simulation inherently provides some amount of cosmological information, which we aim to utilize. However, the minimum halo mass for each simulation, related to the mass of individual DM particles, depends crucially on Ω_m. Therefore, simulations with different values of Ω_m probe different regimes of the halo mass function, resulting in significantly different values for the mean halo number density. We want to retain the former effect while eliminating the latter. To achieve this, we use the largest minimum halo mass across our simulation suite $M_\mathrm{min} = 8.27 \times 10^{12}\,\mathrm{M}_\odot/h$ to impose a mass cut on all simulations. This would allow us to compare more directly to observations where a faint magnitude or luminosity cut is naturally imposed, although we leave such comparisons for future work. Our range of mean halo number density for different cosmology sets vary from $3.29 \times 10^{-5}(h/\mathrm{Mpc})^3$ to $6.88 \times 10^{-4}(h/\mathrm{Mpc})^3$. On the other hand, the mean halo number density from 100 single cosmology datasets is $(3.9 \pm 0.017) \times 10^{-4}(h/\mathrm{Mpc})^3$, which is close with galaxy number density from SDSS-III BOSS CMASS ($\sim 4 \times 10^{-4}(h/\mathrm{Mpc})^3$). This CMASS sample targets Luminous Red Galaxies (LRGs) at $0.4 < \mathrm{z} < 0.7$, which is similar to our redshift range. Additionally most LRGs are expected to be around the center of relatively massive clusters. Since our DM halos have a similar number density to CMASS galaxies, we consider that this data would be appropriate to approximately mimic CMASS-like galaxies, with the caveat that we are imposing a fixed halo occupation scheme.

The mock simulations also do not consider observational effects such as fiber collision or complicated angular masking. However, this simplification affects both our 2pcf and machine learning analyses. So while these mock simulations cannot immediately be compared to observational samples, we expect that they are complex enough to compare the cosmological information content between ML and traditional statistics, which is the primary goal of our study.

2.3 Input Data for Machine Learning & Two-point Correlation Function

For making the input data for the 3D CNN and ViT deep learning methods, we divide the RA and DEC into two segments each: $(-20°, 0°)$ and $(0°, +20°)$, which leads to a fourfold increase in the number of samples. Then we bin the DM halo positions of each lightcone to 64^3 voxels, each axis consists of 64 bins, with one bin representing 0.3125 degrees in RA and DEC, while containing different comoving distances. Before feeding the data into deep learning algorithms, we normalize the values by dividing the total maximum number of DM halos in a single voxel from all 994 datasets.

As a comparison to the deep learning methods, we compute the isotropic two-point correlation function $\xi(r)$ using the public code `KSTAT` [49] with the Landy & Szalay estimator [50]. To properly include the expansion history, we divide each lightcone halo catalog into three redshift slices: $0.3 < z < 0.5$, $0.5 < z < 0.65$, and $0.65 < z < 0.8$, which roughly contain a similar number of DM halos. We found that three is the optimal number of redshift bins to allow constraints on w_0 without degrading the constraining power for other parameters. Then, we calculate the 2pcf from each redshift shell and concatenate them. To avoid the Finger-of-God effect, we calculate the 2pcf from $5 - 150\,\mathrm{Mpc}/h$ with 29 bins, with each bin step size $5\,\mathrm{Mpc}/h$. We further remove the first two radial bins because they are smaller than the cell size of our density field in the deep learning approach and they may otherwise give an unfair advantage to the 2pcf results.

3 Machine Learning Models

Convolutional Neural Networks (CNN) are a prevalent methodology for machine learning tasks in astronomy, especially for computer vision. However, after the introduction of the Vision Transformer (ViT) mechanism, we have found potential in using ViT for predicting cosmological parameters. The main difference between CNN and ViT is related to their inductive bias, which is the ability to predict unseen images during training. ViT has a lower inductive bias than CNN, and that is why training ViT with large datasets is crucial for achieving good performance. In [37], ViT outperformed CNN when trained with a large dataset containing 300 million images. However, producing large datasets in astronomy through simulations has limitations in generating simulations, saving it and also running deep learning algorithms. If one have not enough astronomical data, they can use pre-trained ViT models with other datasets [41, 42]. However, we do not use a pre-trained model and use the same number of data for both CNN and ViT. During training we apply flipping augmentation in the angular plane, while preserving the redshift axis to retain the correct large scale structure evolution. The code is implemented using `Tensorflow` [51] and one NVIDIA RTX 3090 graphic card unit (GPU) with 24 GB VRAM. For the loss function, we use the Mean Squared Error (MSE),

$$\mathcal{L} = \frac{1}{N_{\rm batch}} \sum_{i=1}^{N_{\rm batch}} \sum_{j=1}^{4} \left(x_{j,i}^{\rm pred} - x_{j,i}^{\rm truth} \right)^2 , \tag{3.1}$$

where $(x_1, x_2, x_3, x_4) = (\Omega_{\rm m}, \sigma_8, w_0, S_8)$, and $N_{\rm batch}$ is the number of data in each minibatch. We tried different weights which multiplied each parameters errors, but we couldn't find any advantage of this and decided not to use weights in our loss function. Also, to check the convergence of each machine learning model, we perform 10 independent runs to get the results.

3.1 Convolutional Neural Networks

Main part of CNN consists of the convolution layer and the pooling layer. During the convolution layer, a small kernal moves around the entire image and performs convolution operations with each value. After the convolution layer, we apply the pooling layer, which reduces the data size by half. There are two types of pooling layer: average pooling and maximum pooling. The average pooling calculates the average value of pixels, while the maximum pooling selects the maximum value of pixels. We use average pooling in this paper, while maximum pooling also produced similar results in our tests.

Figure 1 shows our CNN model, which consists of four convolution layers with 32, 64, 256, and 512 filters. For kernel size, we used 3^3 with strides for (1,1,1) and no padding. We put a batch normalization [52] layer before the convolution layer to prevent the vanishing gradient problem. After the convolution layer, we sequentially put an activation layer and an average pooling layer. After finishing the convolution layer, we flatten the output data into 1D data and apply 10% dropout layer [53] to prevent the overfitting problem. Then, we put seven FCN layers with 2,048, 1,024, 512, 256, 128, 64, and 4 neurons, each of which followed by batch normalization and activation. Every activation function used in the convolution layer and dense layer is a Rectified Linear Unit (ReLU, [54]) $\mathrm{ReLU}(x) = \max(x, 0)$, except for the last two activation functions, which are linearly activated. This model has a total of 15,244,200 trainable parameters.

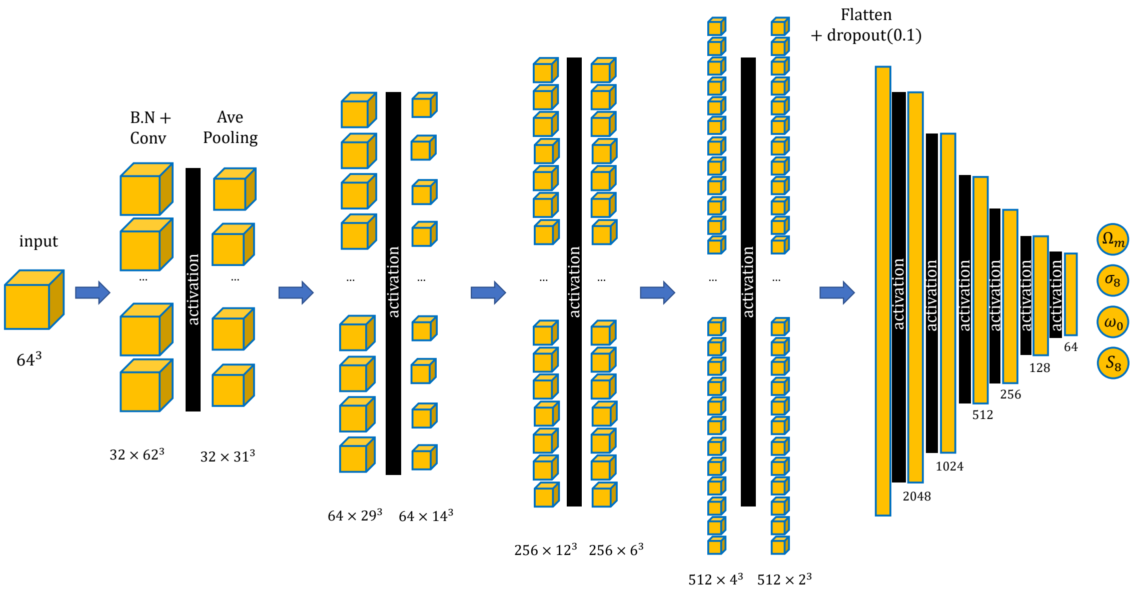

Figure 1: CNN structure of this work. 4 Convolution layers with filters 32, 64, 256 and 512 have applied for feature extraction, followed by 7 fully connected layers with 2,048, 1,024, 512, 256, 128, 64 and 4 neurons for parameter estimation.

We use the Adam optimizer [55], a batch size of 64, and a learning rate of 1e-05. We train our CNN model until epoch to 700, which takes about 106 minutes with a single NVIDIA RTX 3090 GPU. We use 10% of the training data for validation and adopt the model at the epoch where the validation loss is minimum.

3.2 Vision Transformers

Vision Transformers (ViT) are a novel approach to computer vision tasks, which adapt the Transformer architecture, originally developed for Natural Language Processing (NLP), to handle image-like data [37]. In the context of this work, we can employ ViT to analyze the spatial relationships between different parts of the input data, which consist of flattened 3D patches extracted from the lightcone simulation with an input size of 64^3.

Our specific implementation of ViT, as illustrated in Figure 2, divides the input data into 8^3 non-overlapping patches, resulting in a total of 512 patches. These patches are then passed through an embedding layer with a projection dimension of 512. Note that we additionally incorporate positional encoding to convey the relative positions of the tokens in the input sequence. This is crucial for ViT because the spatial relationships between the patches are essential for understanding the structure of the input data.

The main building block of the Transformer architecture is the self-attention mechanism. This mechanism allows the model to learn the relationships between different tokens in the input sequence. In our case, the tokens are the flattened 3D patches. The self-attention mechanism operates on three components: Queries (Q), Keys (K), and Values (V). To create the Q, K, and V, the flattened patches are first projected through learnable linear transformations. These transformations produce three weight matrices, W_K, W_Q, and W_V. Then by multiplying the projected patches with weight matrices, we can finally obtain Q, K

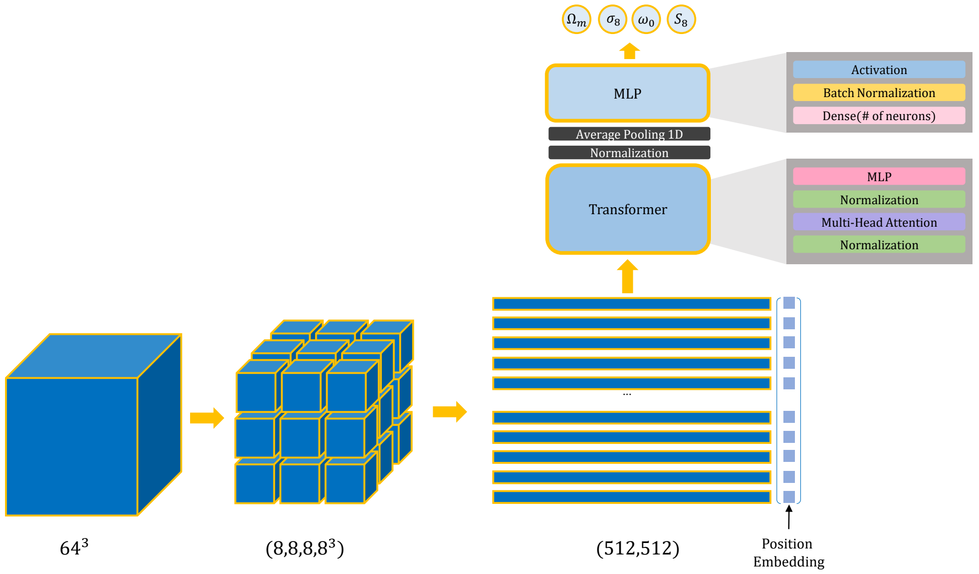

Figure 2: ViT structure of this work. We divide a single 64^3 data into (8,8,8) patches, which indicate patches position (x, y, z), where a single patch is a 8^3 cube, and flatten to a 1D array and added to position embedding vectors. It then undergoes 4 heads in Multi-Head Attention, with a layer depth of 4. After attention layers, we apply the Multi-Layer Perceptron (MLP), which is one Fully Connected layer.

and V. The Q and K are used to calculate the attention scores between patches, while the V is used to compute the output patch embeddings. The attention scores are computed as the dot product between the Q and the K, scaled by the square root of the dimensionality of the K vectors and multiplied by V.

Next, we apply the Transformer Encoder, a series of Multi-Head Attention (MHA) layers, each of which utilizes multiple attention scores from a given (Queries, Keys, Values) set. First, a single attention score is defined as [38]

$$\text{Attention} = \text{softmax}\left(\frac{QK^{\text{T}}}{\sqrt{d_K}}\right) V \ , \tag{3.2}$$

where the superscript "T" denotes the matrix transpose, and d_K is the dimension of K-matrix. Then each MHA layer is defined as

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \ldots, \text{head}_h)W^O \ , \tag{3.3}$$

where $\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$ is the i-th attention score, and $\{W_i^Q, W_i^K, W_i^V\}$ and W^O are weight matrices that will be learned throughout training. This process of computing attention scores and generating output embeddings is performed multiple times, in parallel, by different ViT heads in order to capture different aspects of the image. The resulting output embeddings from all heads are then concatenated and passed through a feedforward neural network to produce the final output.

After the Transformer Encoder processes the input patches, we normalize the outputs to maintain the mean within each batch close to 0 and the standard deviation close to 1.

Subsequently, we apply global average pooling to obtain 1D layers from the Transformer outputs. After adding a 10% dropout layer, we apply a series of fully connected (FCN) layers with 2,048, 1,024, 512, 256, 128, 64, and 4 neurons. To ensure a fair comparison with the CNN model, we use the same structure after the Transformer layer. Note that we did not include any dropout layer inside the FCN layers, as this does not improve the results. This model has a total of 25,425,348 trainable parameters.

While training we use the Adam optimizer, a learning rate of 5e-06, and a batch size of 32 due to memory limitations. This took about 226 minutes on the same GPU as before and again we use the model with the minimum validation loss. In creating this model, we modified the publically available `Keras` example code[1].

3.3 Two-Point Correlation Function Neural Network

We train Fully Connected Layers (FCN) to utilize the 2pcf measurements done in Section 2.3 to estimate cosmological parameters, which we call "2pcf+FCN" hereafter. We first flatten our input data shape of (27,3), where 27 indicates 2pcf radial bins ($15 - 150\,\mathrm{Mpc}/h$) and 3 is the number of redshift shells ($0.3 < z < 0.5$, $0.5 < z < 0.65$, and $0.65 < z < 0.8$), giving us an input vector of 81 elements. We pass this into a 512-neuron hidden layer and then perform batch normalization and activated with a ReLU function. Next, we pass them through a 32-neuron hidden layer and linear activation. Lastly we pass the output from the previous layer to a 4-neuron output layer. We also tried more complex network structures and found that the current version performs the best.

For this running, we choose the learning rate of Adam optimizer to 5e-05 and the number of epochs to 500 to obtain stable results. Compared to our previous CNN and ViT models, the 2pcf+FCN model takes a negligible amount of time for training, while the calculation of 2pcf data takes more than an hour.

4 Analysis

In this section, we discuss the results obtained from two deep learning algorithms, CNN and ViT, and compare their performance with the 2pcf + FCN. Section 4.1 presents a comparison of the outcomes from the three methods and discuss the test results for different cosmological sets. We also examine the results from the systematic test dataset. In Section 4.2, we explore the correlation between the three methods using Pearson correlation and present the combined results from all three approaches. Finally, in Section 4.3, we try to understand the deep learning procedures by analyzing the critical information extracted from the data.

4.1 Model Comparison

We begin by comparing the results obtained from the three methods, CNN, ViT, and 2pcf+FCN, for each cosmological parameter ($\Omega_\mathrm{m}, \sigma_8, w_0, S_8$). In the context of different cosmological sets, we analyze the performance of these methods in terms of accuracy and consistency. Furthermore, we investigate the impact of varying initial random seeds on the outcomes by performing a single cosmological test with 100 variations. For our result, we use the output from 10 runs to reduce the inherent randomness in the initialization process of neural networks and to avoid relying on results from only one run which may yield conclusions that are sensitive to the specific initialization. By conducting 10 runs, we generate a spread of

[1] https://keras.io/examples/vision/image_classification_with_vision_transformer

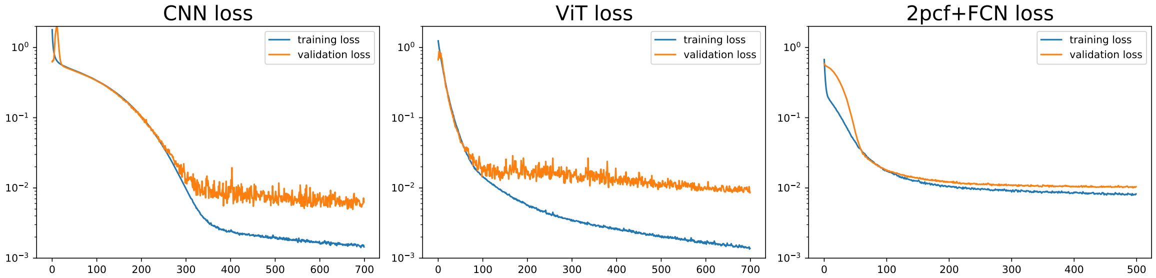

Figure 3: This is 10 averaged loss curve from 10 running sets of CNN (left), ViT (middle), and 2pcf+FCN (right) methods. The blue line is the training loss and the orange line is the validation loss.

outcomes that, when analyzed collectively, mitigate the uncertainties arising from these random initializations. This provides a more robust and credible understanding of the machine learning model's performance. In machine learning this is commonly referred to as model averaging.

Figure 3 presents the loss curves averaged value, at the same epoch, from 10 independent runs of our three methods. Among the training data, which accounts for 80% of the total dataset, we allocated 90% data to the training loss and 10% data to the validation loss. We checked the loss curves for each individual run and there were no outliers. By doing the averaging, we aimed to make comparisons on the same level as the other results presented here. Since we chose the model weights at the lowest validation loss for our final model, it is worth to compare their minimum validation loss. When we identify the lowest validation loss from each run and average the total of 10 values, the order is CNN (0.00404 ± 0.00032), ViT (0.00757 ± 0.00048), and 2pcf+FCN (0.00973 ± 0.00185), where standard deviation comes from 10 runs. The minimum validation loss aligns well with our final result.

Figure 4 presents the one-to-one comparison between truth and prediction values of $(\Omega_{\rm m}\,, \sigma_8, w_0, S_8)$ from the test data. Because we have four subcubes from one cosmological parameter set, we average the prediction from each subcube and use that for our final points. Also, to validate our deep learning model, we run the model 10 times under the same structure and plot the average value and standard deviation for error. All three models can predict all four cosmological parameters, while there are some differences between models and parameters. The predictions of $\Omega_{\rm m}$ and w_0 seem to be more scattered and have larger error bars than the other parameters in all three models, mainly because their allowed ranges are narrower than others.

In Figure 5 we show the marginalised 2D contours from the single cosmology dataset with the models trained on the different cosmological parameters. As same with test result with different cosmological parameter sets, we used a averaged point from the four subcubes in one simulation box and draw 1,000 points from total 10 runs. We see that the 2pcf+FCN result has the largest error, while the CNN result has the smallest error and is also the least biased with respect to truth values.

When comparing the three methods, we found that the 3D CNN has the lowest standard deviation for each parameter, followed closely by the ViT, while the 2pcf+FCN had consistently the largest statistical scatter, as can be seen in Table 2. We can confirm that CNN, a more prevalent method, can generate accurate results with the lightcone-shaped data containing redshift information. Moreover, we found that ViT can also perform well in this

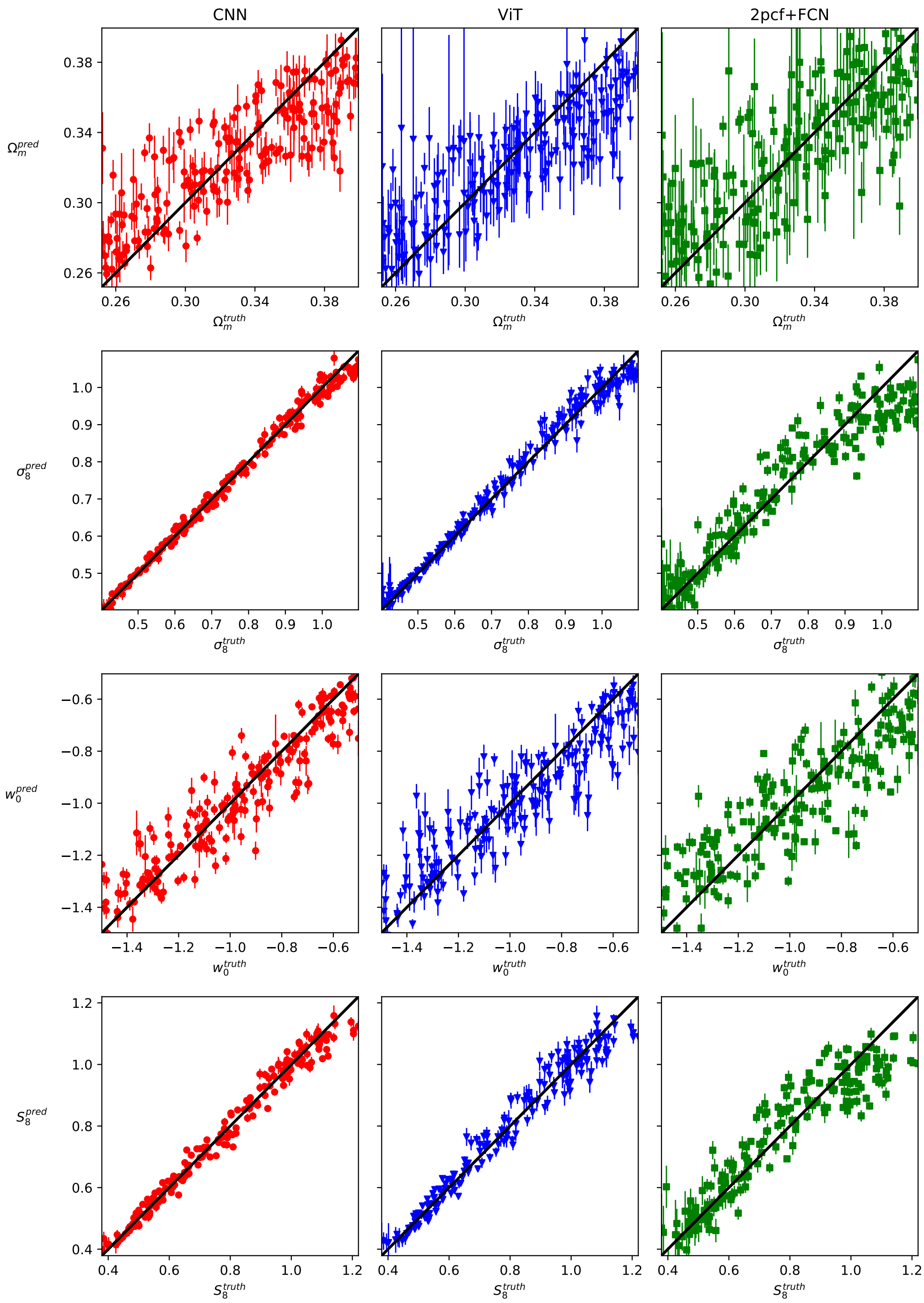

Figure 4: Comparison between truth values of $\Omega_{\rm m}$, σ_8, w_0, and S_8 (top to bottom) from test samples and their predictions from CNN (left), ViT (middle), and 2pcf+FCN (right). Dots and error bars correspond to average and standard deviation from 10 different runs, respectively. Black diagonal line shows where the prediction is same to the truth.

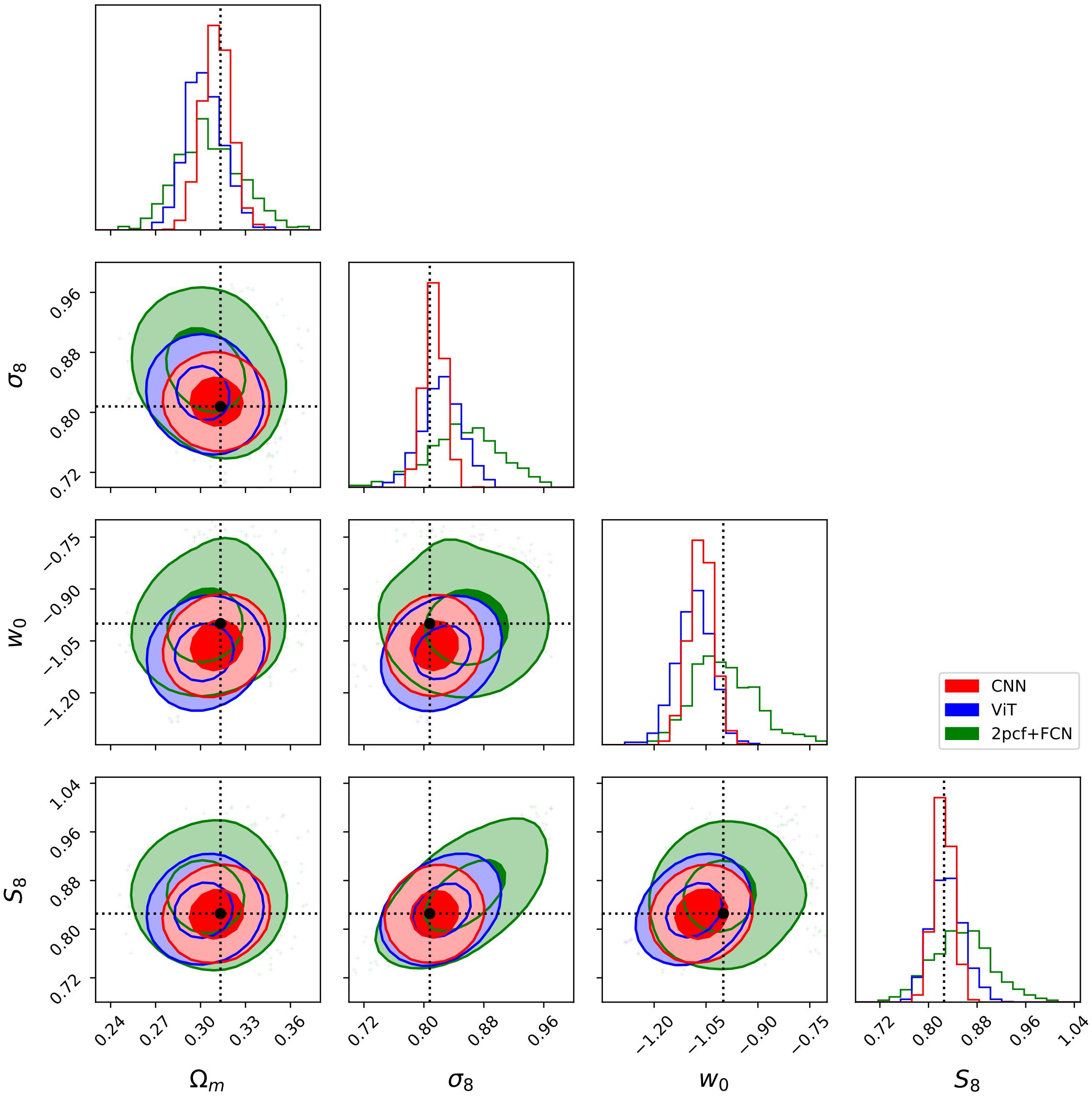

Figure 5: Single cosmology test with CNN (red), ViT (blue), and 2pcf+FCN (green). Black line indicates the truth value of our single cosmology parameter set: $(\Omega_{\rm m}\,, \sigma_8, w_0, S_8) = (0.3133, 0.8079, -1, 0.8256)$. The inner line represents 1σ and the outer line represents 2σ.

type of scenario. Additionally, the traditional two-point correlation function, combined with a simple dense neural network, performed well.

4.2 Correlation and Combined Results

To assess the relationship between the three methods, we employ Pearson correlation to the differences between the predictions and truth values of each of the four parameters from the different methods. We calculate the Pearson correlation using

$$p = \frac{\text{Mean}[(X-\mu_X)(Y-\mu_Y)]}{\sigma_X \sigma_Y}. \tag{4.1}$$

		CNN	ViT	2pcf+FCN
Ω_{m}	$\lvert\text{mean} - \text{truth}\rvert$	**0.0025**	0.0103	0.0072
	std	**0.0096**	0.0126	0.0213
σ_8	$\lvert\text{mean} - \text{truth}\rvert$	**0.0061**	0.0161	0.0451
	std	**0.0130**	0.0266	0.0489
w_0	$\lvert\text{mean} - \text{truth}\rvert$	0.0640	0.0845	**0.0060**
	std	**0.0359**	0.0516	0.0932
S_8	$\lvert\text{mean} - \text{truth}\rvert$	**0.0006**	0.0056	0.0301
	std	**0.0153**	0.0266	0.0495

Table 2: Absolute value of mean-truth and standard deviation of 100 single cosmology. Bold text is minimum value among 3 methods.

		2pcf+FCN	(CNN, ViT)	(CNN, ViT, 2pcf+FCN)
Ω_{m}	$\lvert\text{mean} - \text{truth}\rvert$	0.0072	**0.0054**	0.0056
	std	0.0213	0.0082	**0.0080**
σ_8	$\lvert\text{mean} - \text{truth}\rvert$	0.0451	**0.0081**	0.0101
	std	0.0489	0.0116	**0.0112**
w_0	$\lvert\text{mean} - \text{truth}\rvert$	**0.0060**	0.0707	0.0637
	std	0.0932	0.0316	**0.0312**
S_8	$\lvert\text{mean} - \text{truth}\rvert$	0.0301	**0.0009**	0.0029
	std	0.0495	0.0134	**0.0129**

Table 3: Same to Table 2, except showing 2pcf+FCN, the combined results from CNN and ViT, and the combined results from all three models. See texts for how to calculate the weighted mean.

where X and Y are groups of parameters we want to compare, and $\mu_{X,Y}$ and $\sigma_{X,Y}$ are their mean and standard deviation, respectively. When p is close to $+1$ or -1, it means that X and Y are linearly correlated or anti-correlated, respectively. On the other hand, when the Pearson correlation is close to 0, it means there is no linear correlation between X and Y. From Figure 6, we can observe that the results from CNN and ViT have relatively higher linear correlations than 2pcf+FCN has with others. This suggests that averaging the predictions from CNN/ViT with 2pcf+FCN could yield better results. Although the predictions for σ_8 between CNN and ViT are not strongly correlated and thus the two models may be using different parts of the density field to arrive at their results. Taken together, these results suggest that averaging all 3 models may provide the best cosmological constraints.

Next, we study the performance of combined results of two or more models by calculating the weighted mean

$$\bar{x}_{i,\mathcal{M}} = \frac{\sum\limits_{M \in \mathcal{M}} w_{i,M} x_{i,M}}{\sum\limits_{M \in \mathcal{M}} w_{i,M}} \,, \tag{4.2}$$

where $x_{i,M}$ is a prediction of the given cosmological parameter from a model M, which is a member of our combined model set $\mathcal{M}$, and $w_{i,M} \equiv \sigma_{i,M}^{-2}$ is its weight defined as the inverse of the parameter variance calculated from 10 runs of the single cosmology test. The combined

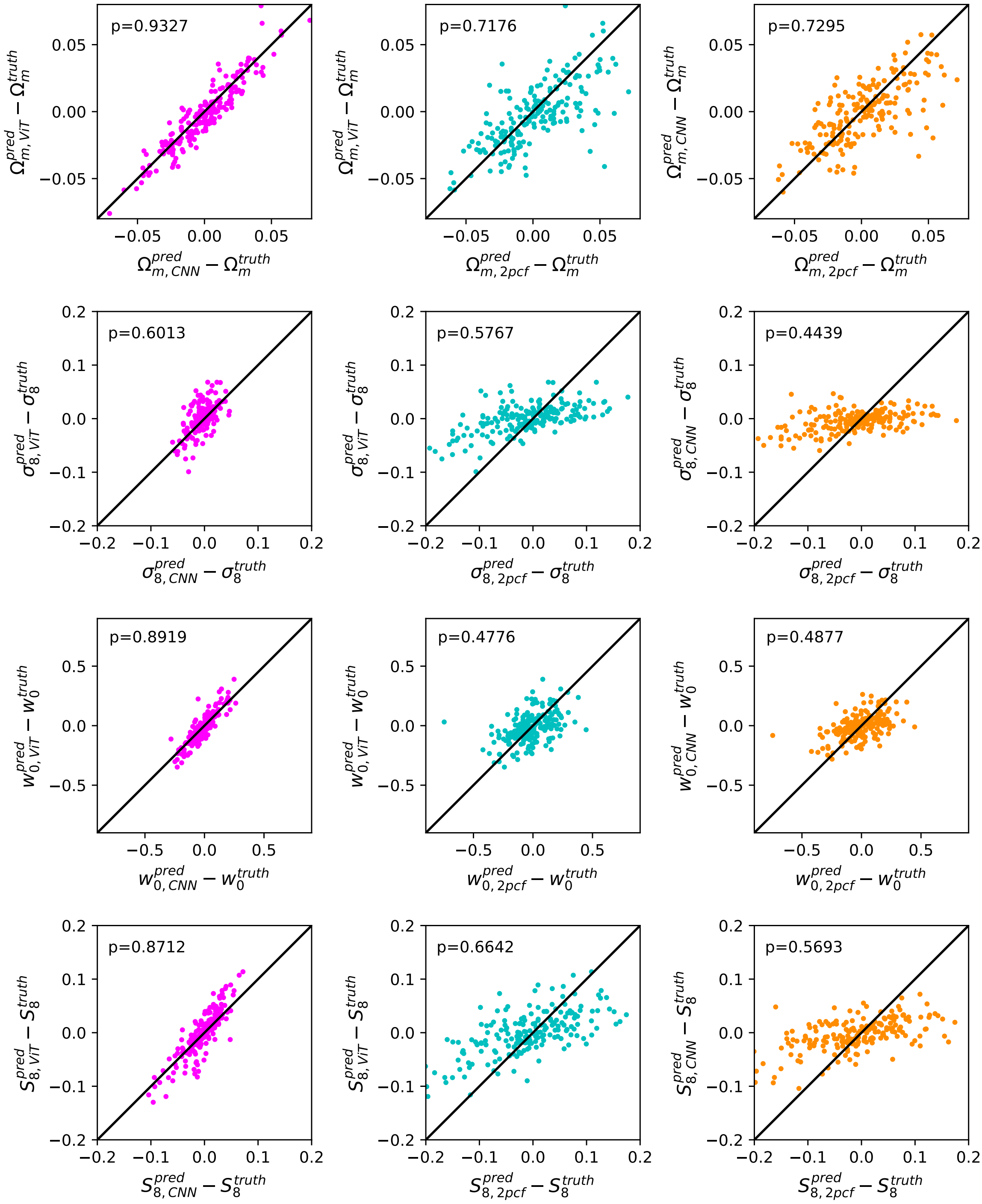

Figure 6: Relations between the difference between predictions and truth values from our three models. From top to bottom panels: $\Omega_{\rm m}$, σ_8, w_0, and S_8. From left to right panels: CNN vs. ViT, 2pcf+FCN vs. ViT, and 2pcf+FCN vs. CNN. The Pearson coefficient, as computed from Eq. 4.1, for the data is displayed in the top left of each panel.

result of CNN, ViT, and 2pcf+FCN is displayed in Figure 7 compared with the 2pcf+FCN

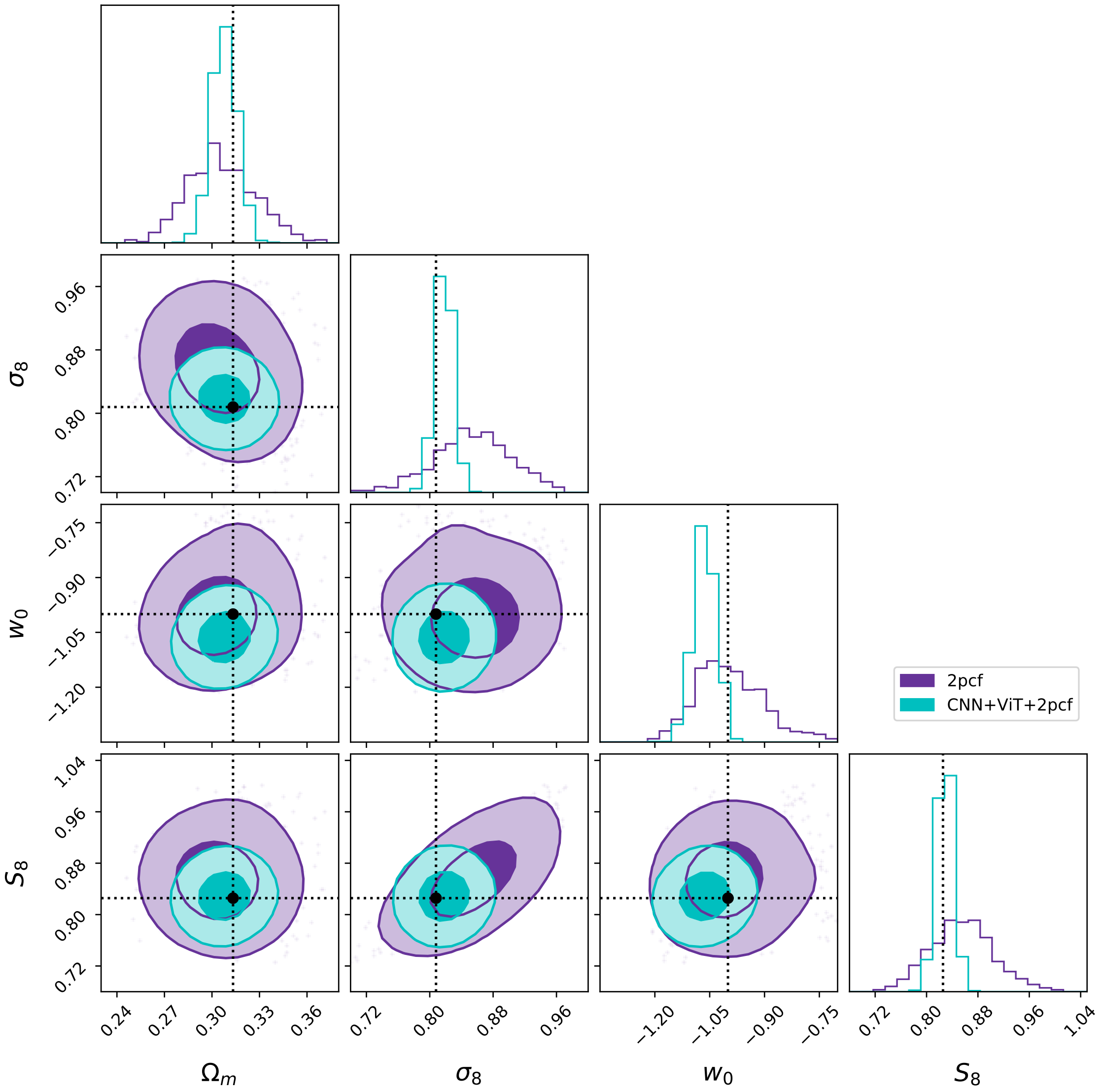

Figure 7: Same as Figure 5, except 2pcf+FCN (purple) and combined results from CNN, ViT, and 2pcf+FCN (cyan) are drawn.

alone. In Table 3, we show the absolute deviation and 1-σ error for each parameter for the 2pcf+FCN alone and for the combined models of (CNN, ViT) and (CNN, ViT, 2pcf+FCN). As expected, the combination of all three models, (CNN, ViT, 2pcf+FCN), provide the best parameter estimation for most cases, where the standard deviation is reduced $1-4\%$ from (CNN, ViT)-combination and $60-77\%$ from 2pcf+FCN alone. Also, mostly thanks to CNN, (CNN, ViT, 2pcf+FCN)-model predict σ_8 and S_8 parameter without noticeable degeneracy (see σ_8-S_8 panel in Figure 7).

4.3 Importance Maps

Understanding the inner workings of machine learning algorithms is crucial when applying deep learning techniques in scientific research. One popular method for this introspection is Gradient-weighted Class Activation Mapping (Grad-CAM) [56]. Grad-CAM computes weights by utilizing gradients through backpropagation between the last layer of the FCN and the last convolutional layer.

We can obtain weights α_k^c for each kernel, k, and class c by calculating the gradients of y_c, the classification score before softmax activation, with respect to A_{ij}^k, the feature maps of the last convolution layer where i, j are spatial indices of the 2D image. These weights are thus given by,

$$\alpha_k^c = \frac{1}{Z} \sum_i \sum_j \frac{\partial y_c}{\partial A_{ij}^k}, \tag{4.3}$$

where Z is a weighted sum of feature maps and the global average pooling is performed by adding up the gradients for each i, j component and then dividing the sum by Z. After obtaining α_k^c, they are multiplied by the feature maps of their corresponding convolutional layer. In classification problems, higher values of y_c indicate more accurate classification, so the ReLU function is applied to show only positive gradients as y_c increases. This calculation yields

$$L_{Grad-CAM}^c = ReLU(\sum_k \alpha_k^c A^k), \tag{4.4}$$

which is final Grad-CAM outputs.

However, in our problem, which is a regression task, some modifications are necessary before applying Grad-CAM. Our final layer represents the actual values instead of class probabilities. Therefore, solely considering high gradients is not appropriate for our objective. To address this, we referred to [57] where, in the case of a regression task, $d = 1/D$ was introduced to replace y_c, where $D = \sqrt{(x - x')^2}$ and indicates the amount of deviation between the prediction, x, and the truth value x'. Smaller D values indicate higher accuracy, and by taking the inverse, we can achieve the desired behavior where d increases as the accuracy improves. The equation 4.3, the gradient of y^c, changes to become the gradient of d,

$$\alpha_k^d = \frac{1}{Z} \sum_i \sum_j \frac{\partial d}{\partial A_{ij}^k}, \tag{4.5}$$

where $\frac{\partial d}{\partial A_{ij}^k} = \frac{\partial d}{\partial x} \frac{\partial x}{\partial A_{ij}^k}$, and $\frac{\partial d}{\partial x} = \frac{-1}{(x-x')^2}$ via the chain rule. Taking $(x - x')^2$ to be a constant, shows that we need only to consider negative gradients of the prediction to adequately amend the original Grad-CAM equation.

While Grad-CAM was originally designed for CNNs, its underlying principle of using gradients to understand feature importance can be adapted for various neural network architectures, including attention-based models like Vision Transformers. For CNN, we use last convolution layer after activation and for ViT, we utilize the last normalized layer after the transformer calculation is done to obtain the gradients. We use the weights trained by our training dataset.

Figures 8 and Figure 9 show a Grad-CAM heatmap of a sample single cosmology data for a CNN model and a ViT model, respectively. The red color which means high Grad-CAM score indicates that the corresponding region contains essential information, while blue color signifies less important parts.

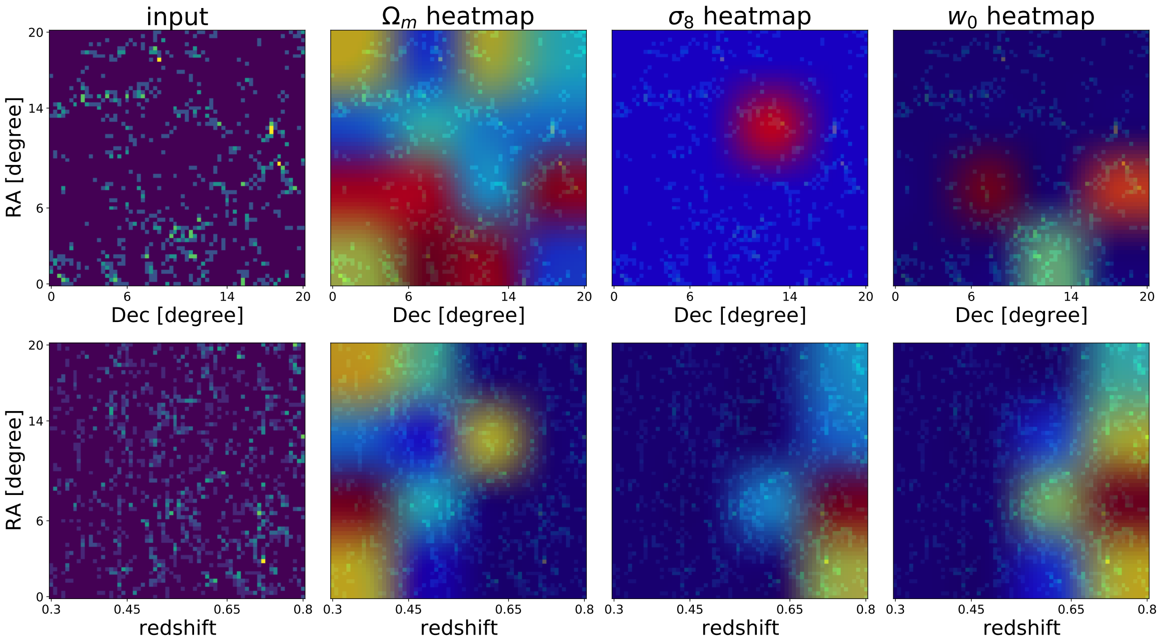

Figure 8: Grad-CAM map of Convolution Neural Network. (Top) Grad-CAM map from RA-Dec plane. (Bottom) Grad-CAM map from RA-redshift plane.

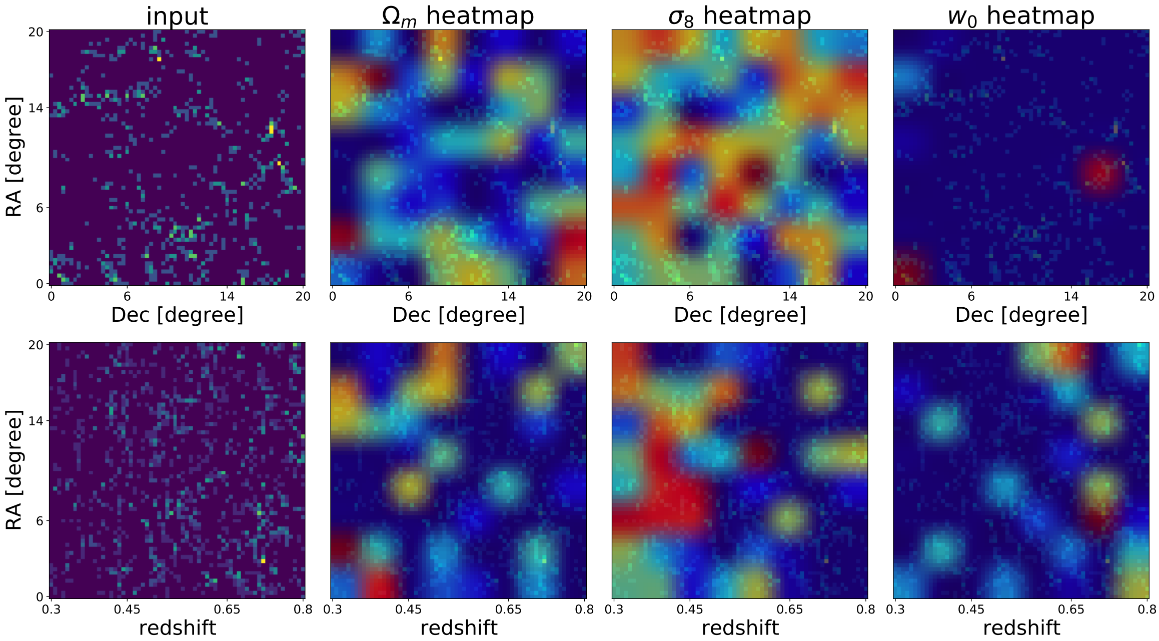

Figure 9: Grad-CAM map of Vision Transformer. (Top) Grad-CAM map from RA-Dec plane. (Bottom) Grad-CAM map from RA-redshift plane.

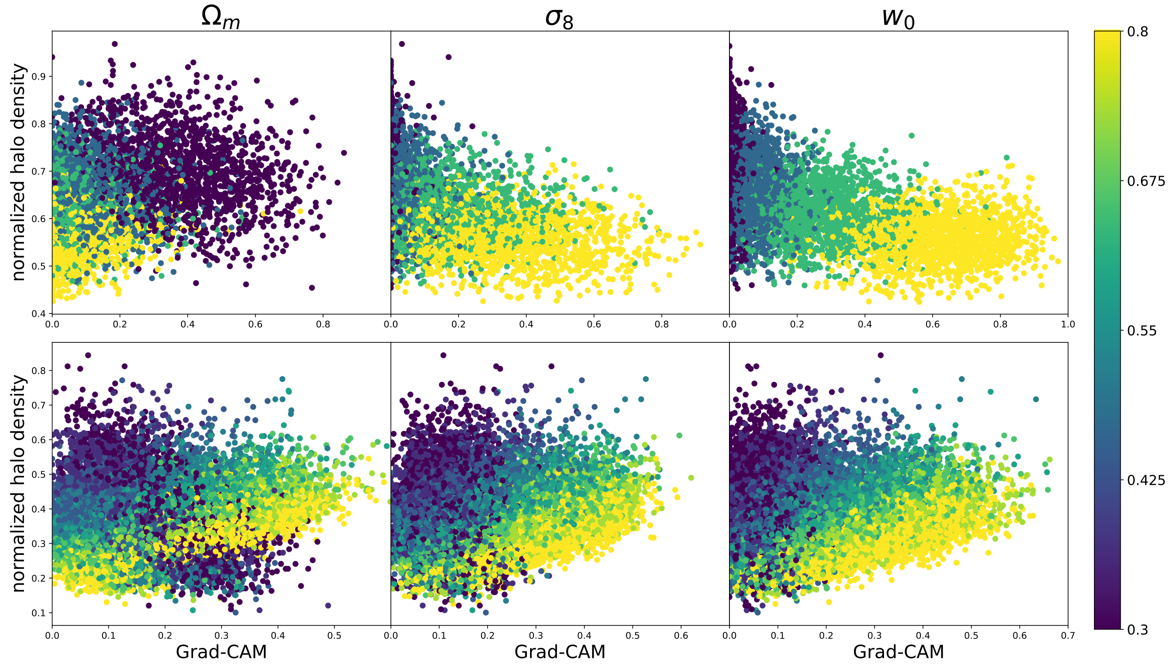

Figure 10: The correlation between normalized halo density and Grad-CAM value ofsingle cosmology catalogues, while using CNN (top) and ViT (bottom). The color bar indicates the redshift of the pixel.

The final Grad-CAM result for CNN has a shape of 4^3, which corresponds to the shape before the last average pooling, while ViT maintains the original patch size of 8^3 throughout the algorithm's calculations. This is why the colored box size in two plots look different.

A visual comparison between the Grad-CAM importance maps and the input data is difficult to interpret by eye. In figure 10 we match pixels importance to their input density value, while also colour coding for the pixel's input redshift. We generated this figure from the single cosmology data and used 100 subcubes for CNN and 40 subcubes for ViT since the final Grad-CAM output shapes of the two algorithms differ, which results in a different number of data points.

For CNN (top panels) we see that the model prediction of Ω_m is influenced most predominately by over densities at lower redshift, while σ_8 and w_0 are more related to under-dense and higher redshift pixels. In the case of ViT (lower panels), there is an upward trend diagonally across all three parameters. This indicates that as the halo density increases at high redshifts, Grad-CAM values also tend to increase. Specifically, for Ω_m, it is noticeable that the halo density at low redshifts has a more significant impact compared to other parameters.

We adopted this approach to gain a better understanding of where our algorithm was focusing within our input data. For example, in the Grad-CAM paper [56], when classifying a "tiger cat", Grad-CAM indicated a high concentration on the stripes of the cat's body, while for classifying a "dog", it focused on the dog's face. The reason we can interpret these findings in this manner is that as humans, we can easily differentiate between dogs and cats when viewing the images, and we can directly judge whether Grad-CAM's results are reasonable or not. However, in our case with a halo density field, it is challenging to immediately determine which areas to focus on for parameter prediction. Therefore, we examined the relationship between the normalized halo density and Grad-CAM.

5 Conclusions

Predicting cosmological parameters from large-scale structures is a crucial aspect of cosmology and a field that benefits greatly from machine learning and big data techniques. In this study, we estimated three cosmological parameters, Ω_m, σ_8, w_0, and one derived parameter, S_8, using two deep learning algorithms — Convolutional Neural Network (CNN) and Vision Transformer (ViT) — for the first time. We also compared these results with a statistical approach that combined standard two-point correlation functions and a simple neural network regression. Our comparison revealed that CNN currently yield the best results, while ViT also show significant potential when applied to predicting cosmological parameters.

Using the same data, we found that the combination of ViT, CNN, and 2pcf resulted in a 62% reduction in the error of Ω_m, a 77% reduction in the error of σ_8, a 66% reduction in the error of w_0, and a 74% reduction in the error of S_8, compared with the 2pcf+FCN alone.

We have shown that ViT could play a role in cosmological model constraints, but they may benefit from pre-training on other datasets or using transfer learning for better performance. Leveraging deep learning method makes us constrain cosmological parameters more tightly than using 2pcf with simple fully connected layers. This proof-of-concept work could be made applicable to current data by forward modeling observational systematics to mock particular cosmological surveys, e.g., SDSS, DESI, etc. We plan to undertake this approach in our future work.

To gain deeper insights into how our machine learning algorithms interpret large scale structure data, we utilized the Gradient-weighted Class Activation Mapping (Grad-CAM) technique. This method enabled us to identify which regions of the data were most informative for the models. Our analysis revealed that Convolutional Neural Networks (CNNs) and Vision Transformers (ViTs) extract valuable information from distinct areas within the dataset. Furthermore, we found that even within a single algorithm, the focus shifts depending on which cosmological parameter is being predicted. Upon statistical evaluation, we also noted variations in the Grad-CAM values across different redshifts and halo densities.

Distinguished among studies that employ machine learning for the prediction of cosmological parameters, our work represents an innovative effort that delves into the inner workings of these algorithms. Rather than presenting our results as definitive answers, our primary objectives were to compare information content between various methods and to gain a human perspective on how our these algorithms accentuated particular facets of the data to yield results. Our approach acknowledges that the outcomes are contingent upon the choice of data and the specific machine learning algorithms and architectures employed.

Acknowledgments

We would like to thank Dongsu Bak, Sangnam Park, Ian Watson and Hannah Jhee for helpful discussions and considerate comments. Also we thanks to ChatGPT for valuable feedback and editorial input.

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (2018R1A6A1-A06024977). C.G.S is also support via the Basic Science Research Program from the National

Research Foundation of South Korea (NRF) funded by the Ministry of Education (2020R1-I1A1A01073494). S.E.H. was supported by the project 우주거대구조를 이용한 암흑우주 연구 ("Understanding Dark Universe Using Large Scale Structure of the Universe"), funded by the Ministry of Science. Our research was partially carried out using the University of Seoul UBAI big data cluster. The authors are especially grateful to KISTI for providing the KREONET network.

We would like to acknowledge the Korean Astronomy and Machine Learning (KAML) meeting series for providing a forum for fruitful discussions on the intersection of astronomy and machine learning.

References

[1] P.J.E. Peebles, *The large-scale structure of the universe* (1980).

[2] M. Davis, G. Efstathiou, C.S. Frenk and S.D.M. White, *The evolution of large-scale structure in a universe dominated by cold dark matter*, *Astrophys. J.* **292** (1985) 371.

[3] J.R. Bond, L. Kofman and D. Pogosyan, *How filaments of galaxies are woven into the cosmic web*, *Nature* **380** (1996) 603 [astro-ph/9512141].

[4] C.G. Sabiu, D.F. Mota, C. Llinares and C. Park, *Probing scalar tensor theories for gravity in redshift space*, *Astron. Astrophys.* **592** (2016) A38 [1603.05750].

[5] C.G. Sabiu, B. Hoyle, J. Kim and X.-D. Li, *Graph Database Solution for Higher-order Spatial Statistics in the Era of Big Data*, *Astrophys. J. Suppl. Ser.* **242** (2019) 29 [1901.00296].

[6] O.H.E. Philcox, J. Hou and Z. Slepian, *A First Detection of the Connected 4-Point Correlation Function of Galaxies Using the BOSS CMASS Sample*, *arXiv e-prints* (2021) arXiv:2108.01670 [2108.01670].

[7] D.J. Eisenstein, I. Zehavi, D.W. Hogg, R. Scoccimarro, M.R. Blanton, R.C. Nichol et al., *Detection of the baryon acoustic peak in the large-scale correlation function of SDSS luminous red galaxies*, *The Astrophysical Journal* **633** (2005) 560.

[8] S. Cole, W.J. Percival, J.A. Peacock, P. Norberg, C.M. Baugh, C.S. Frenk et al., *The 2dF Galaxy Redshift Survey: power-spectrum analysis of the final data set and cosmological implications*, *Mon. Not. R. Astron. Soc.* **362** (2005) 505 [astro-ph/0501174].

[9] F. Beutler, C. Blake, M. Colless, D.H. Jones, L. Staveley-Smith, L. Campbell et al., *The 6dF Galaxy Survey: baryon acoustic oscillations and the local Hubble constant*, *Mon. Not. R. Astron. Soc.* **416** (2011) 3017 [1106.3366].

[10] X. Luo, Z. Wu, M. Li, Z. Li, C.G. Sabiu and X.-D. Li, *Cosmological Constraints from the Redshift Dependence of the Alcock-Paczynski Effect: Fourier Space Analysis*, *Astrophys. J.* **887** (2019) 125 [1908.10593].

[11] H. Park, C. Park, C.G. Sabiu, X.-d. Li, S.E. Hong, J. Kim et al., *Alcock-Paczynski Test with the Evolution of Redshift-space Galaxy Clustering Anisotropy*, *Astrophys. J.* **881** (2019) 146 [1904.05503].

[12] Z. Zhang, G. Gu, X. Wang, Y.-H. Li, C.G. Sabiu, H. Park et al., *Nonparametric Dark Energy Reconstruction Using the Tomographic Alcock-Paczynski Test*, *Astrophys. J.* **878** (2019) 137 [1902.09794].

[13] X.-D. Li, C.G. Sabiu, C. Park, Y. Wang, G.-b. Zhao, H. Park et al., *Cosmological Constraints from the Redshift Dependence of the Alcock-Paczynski Effect: Dynamical Dark Energy*, *Astrophys. J.* **856** (2018) 88 [1803.01851].

[14] X.-D. Li, C. Park, C.G. Sabiu, H. Park, C. Cheng, J. Kim et al., *Cosmological Constraints from the Redshift Dependence of the Volume Effect Using the Galaxy 2-point Correlation Function across the Line of Sight*, *Astrophys. J.* **844** (2017) 91 [1706.09853].

[15] X.-D. Li, C. Park, C.G. Sabiu, H. Park, D.H. Weinberg, D.P. Schneider et al., *Cosmological Constraints from the Redshift Dependence of the Alcock-Paczynski Effect: Application to the SDSS-III BOSS DR12 Galaxies*, *Astrophys. J.* **832** (2016) 103 [1609.05476].

[16] X.-D. Li, C. Park, C.G. Sabiu and J. Kim, *Cosmological constraints from the redshift dependence of the Alcock-Paczynski test and volume effect: galaxy two-point correlation function*, *Mon. Not. R. Astron. Soc.* **450** (2015) 807 [1504.00740].

[17] N. Kaiser, *Clustering in real space and in redshift space*, *Mon. Not. R. Astron. Soc.* **227** (1987) 1.

[18] S. Alam, M. Ata, S. Bailey, F. Beutler, D. Bizyaev, J.A. Blazek et al., *The clustering of galaxies in the completed SDSS-III Baryon Oscillation Spectroscopic Survey: cosmological analysis of the DR12 galaxy sample*, *Mon. Not. R. Astron. Soc.* **470** (2017) 2617 [1607.03155].

[19] J.E. Bautista, R. Paviot, M. Vargas Magaña, S. de la Torre, S. Fromenteau, H. Gil-Marín et al., *The completed SDSS-IV extended Baryon Oscillation Spectroscopic Survey: measurement of the BAO and growth rate of structure of the luminous red galaxy sample from the anisotropic correlation function between redshifts 0.6 and 1*, *Mon. Not. R. Astron. Soc.* **500** (2021) 736 [2007.08993].

[20] A. de Mattia, V. Ruhlmann-Kleider, A. Raichoor, A.J. Ross, A. Tamone, C. Zhao et al., *The completed SDSS-IV extended Baryon Oscillation Spectroscopic Survey: measurement of the BAO and growth rate of structure of the emission line galaxy sample from the anisotropic power spectrum between redshift 0.6 and 1.1*, *Mon. Not. R. Astron. Soc.* **501** (2021) 5616 [2007.09008].

[21] R. Neveux, E. Burtin, A. de Mattia, A. Smith, A.J. Ross, J. Hou et al., *The completed SDSS-IV extended Baryon Oscillation Spectroscopic Survey: BAO and RSD measurements from the anisotropic power spectrum of the quasar sample between redshift 0.8 and 2.2*, *Mon. Not. R. Astron. Soc.* **499** (2020) 210 [2007.08999].

[22] F. Villaescusa-Navarro, C. Hahn, E. Massara, A. Banerjee, A.M. Delgado, D.K. Ramanah et al., *The Quijote Simulations*, *Astrophys. J. Suppl. Ser.* **250** (2020) 2 [1909.05273].

[23] S. Yuan, L.H. Garrison, D.J. Eisenstein and R.H. Wechsler, *Stringent σ_8 constraints from small-scale galaxy clustering using a hybrid MCMC + emulator framework*, *Mon. Not. R. Astron. Soc.* **515** (2022) 871 [2203.11963].

[24] S. Yuan, B. Hadzhiyska and T. Abel, *Full forward model of galaxy clustering statistics with ABACUSSUMMIT light cones*, *Mon. Not. R. Astron. Soc.* **520** (2023) 6283 [2211.02068].

[25] T. Kacprzak, J. Fluri, A. Schneider, A. Refregier and J. Stadel, *CosmoGridV1: a simulated wCDM theory prediction for map-level cosmological inference*, *J. Cosmol. Astropart. Phys.* **2023** (2023) 050 [2209.04662].

[26] C. Hahn, M. Eickenberg, S. Ho, J. Hou, P. Lemos, E. Massara et al., *Simbig: mock challenge for a forward modeling approach to galaxy clustering*, *Journal of Cosmology and Astroparticle Physics* **2023** (2023) 010.

[27] Y. Kobayashi, T. Nishimichi, M. Takada and H. Miyatake, *Full-shape cosmology analysis of the SDSS-III BOSS galaxy power spectrum using an emulator-based halo model: A 5% determination of σ_8*, *Phys. Rev. D* **105** (2022) 083517 [2110.06969].

[28] Y. LeCun, Y. Bengio and G. Hinton, *Deep learning*, *Nature* **521** (2015) 436.

[29] I.J. Goodfellow, Y. Bengio and A. Courville, *Deep Learning*, MIT Press, Cambridge, MA, USA (2016).

[30] S. Ravanbakhsh, J. Oliva, S. Fromenteau, L.C. Price, S. Ho, J. Schneider et al., *Estimating cosmological parameters from the dark matter distribution*, 2017. 10.48550/ARXIV.1711.02033.

[31] A. Mathuriya, D. Bard, P. Mendygral, L. Meadows, J. Arnemann, L. Shao et al., *Cosmoflow: Using deep learning to learn the universe at scale*, 2018. 10.48550/ARXIV.1808.04728.

[32] S. Pan, M. Liu, J. Forero-Romero, C.G. Sabiu, Z. Li, H. Miao et al., *Cosmological parameter estimation from large-scale structure deep learning*, 2019. 10.48550/ARXIV.1908.10590.

[33] A. Lazanu, *Extracting cosmological parameters from n-body simulations using machine learning techniques*, *Journal of Cosmology and Astroparticle Physics* **2021** (2021) 039.

[34] M. Ntampaka, D.J. Eisenstein, S. Yuan and L.H. Garrison, *A hybrid deep learning approach to cosmological constraints from galaxy redshift surveys*, *The Astrophysical Journal* **889** (2020) 151.

[35] C.G. Sabiu, K. Kadota, J. Asorey and I. Park, *Probing ultra-light axion dark matter from 21 cm tomography using Convolutional Neural Networks*, *J. Cosmol. Astropart. Phys.* **2022** (2022) 020 [2108.07972].

[36] T. Kacprzak and J. Fluri, *Deeplss: breaking parameter degeneracies in large scale structure with deep learning analysis of combined probes*, 2022. 10.48550/ARXIV.2203.09616.

[37] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner et al., *An image is worth 16x16 words: Transformers for image recognition at scale*, 2020. 10.48550/ARXIV.2010.11929.

[38] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez et al., *Attention is all you need*, 2017. 10.48550/ARXIV.1706.03762.

[39] J.Y.-Y. Lin, S.-M. Liao, H.-J. Huang, W.-T. Kuo and O.H.-M. Ou, *Galaxy morphological classification with efficient vision transformer*, 2021. 10.48550/ARXIV.2110.01024.

[40] S. Wei, Y. Li, W. Lu, N. Li, B. Liang, W. Dai et al., *Unsupervised galaxy morphological visual representation with deep contrastive learning*, *Publications of the Astronomical Society of the Pacific* **134** (2022) 114508.

[41] K.-W. Huang, G.C.-F. Chen, P.-W. Chang, S.-C. Lin, C.-J. Hsu, V. Thengane et al., *Strong gravitational lensing parameter estimation with vision transformer*, 2022. 10.48550/ARXIV.2210.04143.

[42] L. Rustige, J. Kummer, F. Griese, K. Borras, M. Brüggen, P.L.S. Connor et al., *Morphological classification of radio galaxies with wgan-supported augmentation*, 2022. 10.48550/ARXIV.2212.08504.

[43] Planck Collaboration, N. Aghanim, Y. Akrami, M. Ashdown, J. Aumont, C. Baccigalupi et al., *Planck 2018 results. VI. Cosmological parameters*, *Astron. Astrophys.* **641** (2020) A6 [1807.06209].

[44] P. Monaco, T. Theuns and G. Taffoni, *The pinocchio algorithm: pinpointing orbit-crossing collapsed hierarchical objects in a linear density field*, *Monthly Notices of the Royal Astronomical Society* **331** (2002) 587.

[45] W.H. Press and P. Schechter, *Formation of Galaxies and Clusters of Galaxies by Self-Similar Gravitational Condensation*, *Astrophys. J.* **187** (1974) 425.

[46] R.K. Sheth and G. Tormen, *Large-scale bias and the peak background split*, *Mon. Not. R. Astron. Soc.* **308** (1999) 119 [astro-ph/9901122].

[47] A. Jenkins, C.S. Frenk, S.D.M. White, J.M. Colberg, S. Cole, A.E. Evrard et al., *The mass function of dark matter haloes*, *Mon. Not. R. Astron. Soc.* **321** (2001) 372 [astro-ph/0005260].

[48] J. Tinker, A.V. Kravtsov, A. Klypin, K. Abazajian, M. Warren, G. Yepes et al., *Toward a Halo Mass Function for Precision Cosmology: The Limits of Universality*, *Astrophys. J.* **688** (2008) 709 [0803.2706].

[49] C. Sabiu, "KSTAT: KD-tree Statistics Package." Astrophysics Source Code Library, record ascl:1804.026, Apr., 2018.

[50] S.D. Landy and A.S. Szalay, *Bias and Variance of Angular Correlation Functions*, *Astrophys. J.* **412** (1993) 64.

[51] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro et al., *TensorFlow: Large-scale machine learning on heterogeneous systems*, 2015.

[52] S. Ioffe and C. Szegedy, *Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift*, *arXiv e-prints* (2015) arXiv:1502.03167 [1502.03167].

[53] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever and R. Salakhutdinov, *Dropout: a simple way to prevent neural networks from overfitting*, *The journal of machine learning research* **15** (2014) 1929.

[54] X. Glorot, A. Bordes and Y. Bengio, *Deep sparse rectifier neural networks*, in *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 315–323, JMLR Workshop and Conference Proceedings, 2011.

[55] D.P. Kingma and J. Ba, *Adam: A method for stochastic optimization*, *arXiv preprint arXiv:1412.6980* (2014) .

[56] R.R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh and D. Batra, *Grad-CAM: Visual explanations from deep networks via gradient-based localization*, *International Journal of Computer Vision* **128** (2019) 336.

[57] D. Kanda, S. Kawai and H. Nobuhara, *Visualization method corresponding to regression problems and its application to deep learning-based gaze estimation model*, *Journal of Advanced Computational Intelligence and Intelligent Informatics* **24** (2020) 676.