Performance Evaluation of MaxViT in Cosmological Parameter Inference

Group 15 // Project 85 // Co-Instructor: Wenda Zhou

Seonhye Yang **Michael Healy** **Bhavna Thakar**

{sy3420, mbh425, bt2227}@nyu.edu

Shirley Ho **Chris Pedersen** **Bruno Regaldo Saint Blancard** **Michael Eickenberg**

Abstract

The goal of this work is to compare the performance of two models on the task of cosmological parameter inference using CAMELS, a large-scale dataset of cosmological simulations. The first model is a Learnable Scattering Transform [1]. The second model is a Multi-Axis Vision Transform (MaxViT), which consists in a combination of convolutional neural networks and vision transformers. In the work, the images used are two-dimensional CAMELS maps, which are slices taken from the original three-dimensional maps. The parameters are $\Omega_M, \sigma_8, A_{SN1}, A_{SN2}, A_{AGN1}, A_{AGN2}$. The first two are cosmological parameters and the next four are astrophysical parameters that control the efficiency of the supernova and Active Galactic Nuclei (AGN) feedback. In order to test the performance of MaxViT, the trainable parameters were optimized to estimate Ω_M and σ_8. The scattering network hybrid model utilizing wavelets we are trying to improve upon had an error of 2.199% for Ω_M and 1.670% for σ_8. Meanwhile, MaxViT had an error of 2.016% for Ω_M and 1.518% for σ_8- not improving upon the Wavelet model, but getting rather close with a possibility of surpassing with further tuning. Similar to the Wavelet model, the MaxViT model poorly constrained the four efficiency/feedback parameters, with further improvement possibly coming from a combination of a different algorithm specifically for those four inferences.

1 Introduction

In the world of deep learning for image analysis, convolutional neural networks (CNNs) are often regarded as the crux. Especially in the cosmological sphere, they have the ability to extract more information than their algorithmic predecessors. Despite their impressive performance, CNNs run into some potential problems. For one, they need enormous amounts of training data for optimization, which can quickly get computationally expensive. Another issue is that they have very low interpretability—their tiers of layers make for architecture that is essentially undecipherable to the human eye. This renders them "black boxes" which take input and produce output without comprehensible steps. To account for this lack of interpretability, prior works have outlined a model in which CNN layers are replaced by wavelets, making the results much easier to understand [1].

In this work we aim to compare the performance of this combination of wavelets and convolutional neural networks to that of a novel convolutional neural network-vision transformer hybrid called Multi-Axis Vision Transformer (MaxViT) for the inference of cosmological parameters. Both models are trained on a cosmological dataset called CAMELS, which simulate the large-scale structure of the Universe.

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

2 Related Work

The goal of this project is to build on work that has been previously done to address the pitfalls that CNNs tend to run into when used on cosmological data. The models outlined [1] utilize a cascade of analytic wavelet transforms followed by complex modulus to construct descriptors of the input signal that behave smoothly to small deformations and that can be made to exhibit known symmetries of the data. The learning scattering transform model proposed in this work is able to outperform a CNN on both large and small datasets, outperforming more appreciably on smaller sets. The goal of this work is to optimize MaxViT and compare its performance to the architecture of previous models.

3 Problem Definition and Algorithm

3.1 Task

The intention of this work is to compare the performance of a state-of-the-art vision transformer with a more traditional CNN, and a wavelet network with far fewer parameters. In previous iterations of this task, scattering transforms were used to create a model that replaced CNN layers with wavelets, which are short oscillations that are able to retrieve information from data. By using MaxViT, which is a hybrid CNN and vision transformer, we hope to also outperform CNN and compare results yielded by MaxViT to the original learnable wavelet model. This work aims to measure the success of this task through the inference of the following parameters: $\Omega_M, \sigma_8, A_{SN1}, A_{SN2}, A_{AGN1}, A_{AGN2}$. The first two are cosmological variables that correspond to the matter density parameter and the root-mean-square matter fluctuation parameter, respectively. The remaining four control the efficiency of the supernova and Active Galactic Nuclei (AGN) feedback. In addition, we looked at training and validation loss to assess convergence of our optimization and overfitting.

3.2 Algorithm

The model being tested in this work is MaxViT, which, as previously described, uses both components of CNNs and vision transformers. A transformer is a model that learns context and tracks sequential data—transformers have recently shown a lot of potential in computer vision. However, transformers alone tend to lack scalability, rendering them inefficient when dealing with large-scale image data, like that in CAMELS. What makes MaxViT exciting is that it combines a new attention model, multi-axis attention, with convolutions that allow MaxViT to discern through an entire network, making it suitable for large image data. The way multi-axis attention works is through the use of blocked local and dilated global attention, allowing for spatial interactions on inputs of linear complexity. Each MaxViT block is built by concatenating MBConv, an inverted linear bottleneck layer with depth-wise separable convolution, with multi-axis attention. By repeating this building block over a series of stages, MaxViT is able to successfully perform a wide variety of vision tasks.[3]
Figure 1 shows MaxViT structure as a whole and zoomed into a single block to show the combination of MBConv, block attention, and grid attention.

Initially a crude implementation of MaxViT was used that only took images of size 224 so we cropped our images by 16 on each side [1]. However, we decided to use Google's initial implementation but adapted for PyTorch (instead of its original TensorFlow build) which did not require cropping [2]. Some of the parameters we fine-tuned were the batch size, mixed precision, learning rate, and weight decay. Batch size refers to the number of batches being used and mixed precision is a simple Boolean parameter. For learning rate and weight decay, we aimed to find the maximal learning rate for batch size and the optimal weight decay value—around 0.0005. In addition, MaxViT comes in multiple sizes: tiny, small, base, large. These sizes correspond to the number of parameters and attention blocks used in the model, where larger models use more parameters and contain more attention blocks. The number of trainable parameters for tiny, small, base, and large were approximately 30m, 67m, 118m, and 210m, respectively.

[1] https://github.com/ChristophReich1996/MaxViT
[2] https://github.com/google-research/maxvit

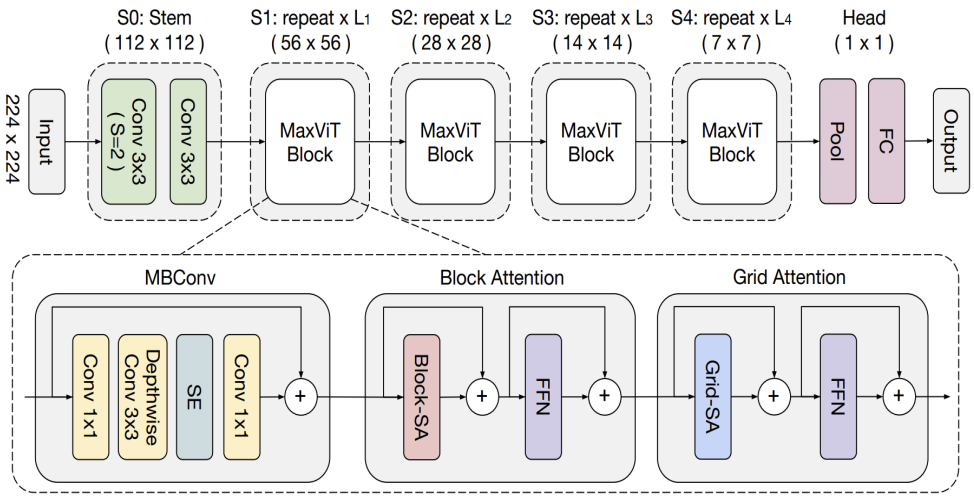

Figure 1: MaxViT architecture showing both a single block and the model as a whole[3]

4 Experimental Evaluation

4.1 Data

The data being used in the project is the Cosmology and Astrophysics with Machine Learning Simulations (CAMELS) dataset. This dataset consists of hundreds of thousands of two dimensional maps and three dimensional grids that represent cosmic regions that span 100 million light years. Each two dimensional map denotes a region of area $(25h^{-1}\text{Mpc})^2$ and each three dimensional grid denotes a region of volume $(25h^{-1}\text{Mpc})^3$[3]. The data contains a myriad of cosmological properties including properties of cosmic gas, dark matter, and stars from over 2000 simulated universes over a variety of cosmic times. All simulations in the dataset share a set of parameters and are varied over two distinct cosmological parameters, Ω_M and σ_8. The full dataset encompasses over 70 Terabytes however in this work we are only using the two dimensional maps. The training data consists of 1,000 simulations drawn from a Latin hypercube in 6 parameters: 2 cosmological, and 4 describing astrophysical feedback effects. Figure 2 is an example of what a map from CAMELS looks like.

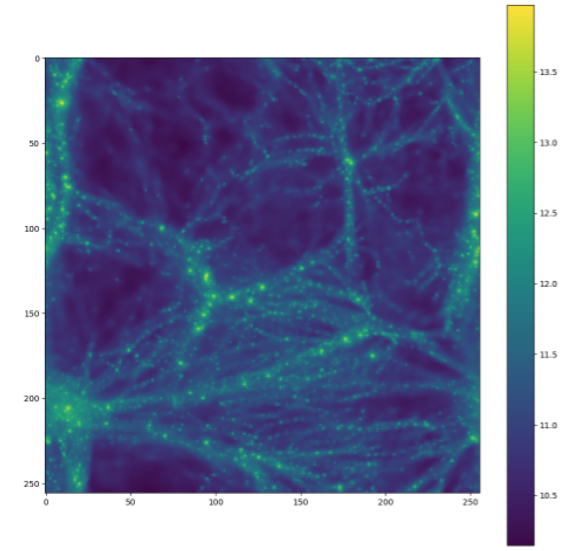

Figure 2: Example of CAMELS 2D map

4.2 Methodology

The bulk of this project was spent on training and optimizing MaxViT for the CAMELS dataset. To begin, we started training on a smaller, space-efficient model using the same parameters used for the scattering network hybrid model—batch size, number of workers, learning rate for scattering layers, learning rate for weight decay, and the number of epochs. This was done on a subset of the training data that includes 1000 maps from the 15000 maps in the full CAMELS dataset. This smaller dataset is about 200 MB and the full dataset it about 3000 MB.

To fine-tune parameters, we varied a few parameters and plotted the difference in loss between training and validation data to evaluate the convergence of the optimization and overfitting. Once the parameters were optimized, we reran MaxViT on the validation data using the updated parameters. After determining parameters, we scaled up to a larger model—MaxViT comes in a number of sizes ranging from tiny to large. Once we successfully ran validation on a base model, we scaled up to using the full CAMELS dataset and got a final test performance.

After running MaxViT on the CAMELS data as is, we performed mutations on the data. Often, adding quality white noise improves the generalization of deep learning models[5]. The way that we transformed the data was by reflecting the image on the y-axis. Generally, visual transformers benefit from using transformed training data to improve model performance. Other ways to transform data include shifting, brightness, cropping, or random augmentations.

4.3 Results

Figure 3 illustrates the loss in both training and validation data over 150 epochs (100 for MaxViT using transformed data due to time constraints). We experimented with different epoch numbers and decided on 150 because this is when the loss began to plateau in the validation data. The models we compared were MaxViT base using the original CAMELS data, MaxViT base using the transformed CAMELS data, the scattering network model using wavelets, and the convolutional neural network model.

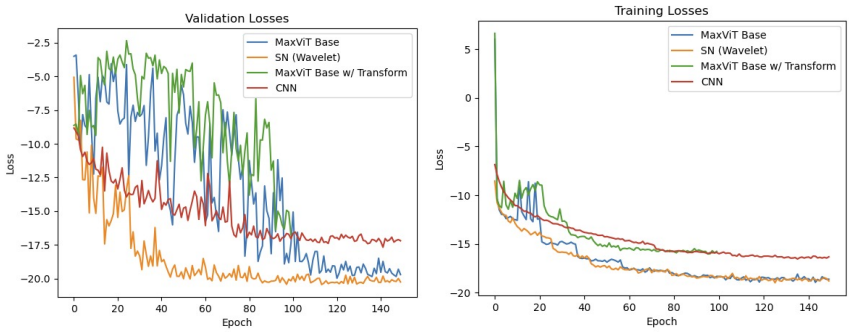

Figure 3: Validation and Training losses for the MaxViT base, scattering network (SN), MaxViT base with augmented dataset (trained for 100 epochs due to time constraints), CNN models

Table 1 outlines the overall results of this report, showing MaxViT Base, MaxViT base trained on the transformed dataset, the SN+Wavelet model, and CNN as a baseline. The results provided in Figure 4 outline the estimations of the two parameters (Ω_M, σ_8) that we decided to focus on. We chose to ignore the other four parameters for this project because we found that MaxViT failed to estimate them well (see Figure 6), producing relatively high errors. It did, however, come very close to the scattering networks model in estimating the first two. The scattering networks model yielded errors of 1.654 for Ω_M and 1.592 for σ_8 while MaxViT yielded errors of 2.199% for Ω_M and 1.670% for σ_8. Figure 4 shows MaxViT on the original dataset, while Figure 5 shows MaxViT on the transformed dataset. Contrary to the generalization that using transformed data improves performance in visual transformers, we found that the estimation error increased remarkably for both parameters. The transformed data yielded errors of 4.081% for Ω_M and 2.917% for σ_8

Model	Ω_M **Error (%)**	σ_8 **Error (%)**	**Train. Loss**	**Val. Loss**
MaxViT Base (150 epochs)	2.199	1.670	-18.624	-19.713
With Transformed Data (100 epochs)	4.081	2.917	-15.765	-16.757
SN Hybrid (Wavelet) (150 epochs)	2.016	1.518	-18.797	-20.261
CNN (150 epochs)	3.979	2.039	-16.323	-17.202

Table 1: Results for Omega and Sigma error (lower is better), as well as for Training and Validation loss (lower is better).

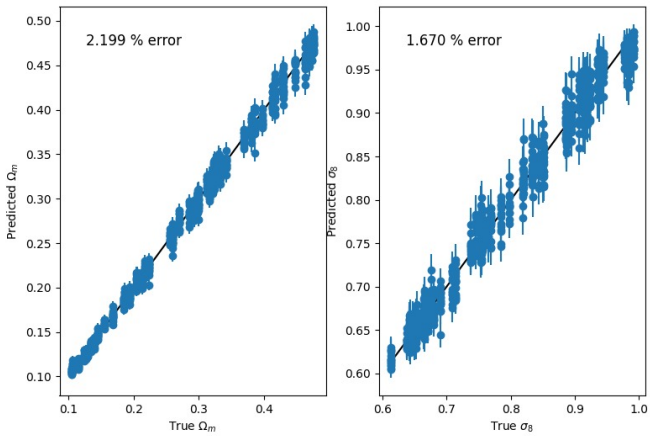

Figure 4: Inference results on a test set for the MaxViT base model using original CAMELS dataset.

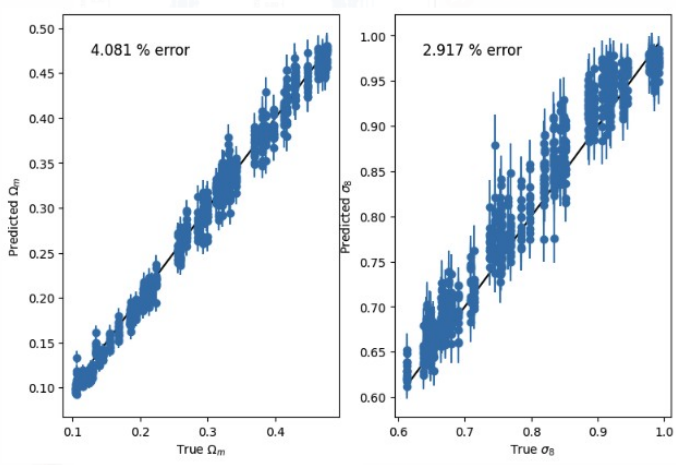

Figure 5: Inference results on a test set for the MaxViT base model using transformed CAMELS dataset.

We decided to only consider Ω_M and σ_8 because the feedback parameters $A_{SN1}, A_{SN2}, A_{AGN1}, A_{AGN2}$ were not able to be estimated accurately. Figure 6 shows the errors in estimation for these four parameters.

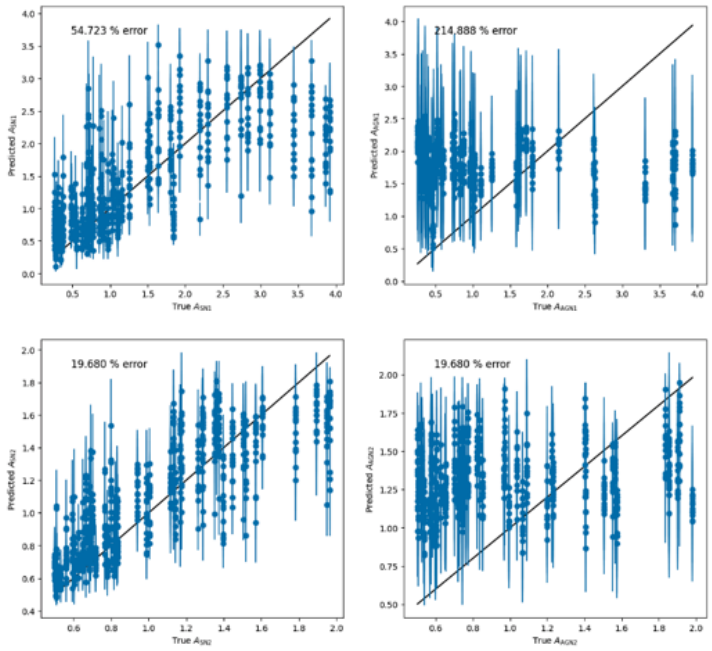

Figure 6: Estimating Active Galactic Nuclei feedback parameters

These figures were created used Weights and Biases (WandB), which allowed us to keep track of our experiments and the hyperparameters we were attempting to fine-tune.

4.4 Discussion

MaxViT is not currently outperforming the scattering networks model in either training and validation loss, nor in either considered parameter error; however, the results of MaxViT are only slightly worse than SN+Wavelet, and well outperform the CNN baseline. MaxViT using transformed data performed worse than the CNN, however. Nevertheless, in order to match or improve upon the results of the Wavelet model, there is still work to be done to increase MaxViT's performance on all accounts. We would like to use Optuna—an automated hyperparameter optimization framework—to further fine-tune hyperparameters to determine whether it can finally surpass the scattering network hybrid model in regards to both Ω_M and σ_8 parameter inference. For the AGN feedback parameters, the results were not satisfactory for MaxViT—or the other considered models for that matter—and could simply be an issue in just using MaxViT alone. Utilizing a combination of MaxViT—for Ω_M and σ_8—and another algorithm to infer AGNI feedback would be the necessary next step for inferring all parameters in CAMELS simulation data.

The way we trained was using moment loss. Equation 1 defines the loss function [1].

$$\mathcal{L} = \sum_{i=1}^{6} \log\left(\sum_{j \in batch} (\theta_{i,i} - \mu_{i,j})^2\right) + \sum_{i=1}^{6} \log\left(\sum_{j \in batch} \left((\theta_{i,i} - \mu_{i,j})^2 - \sigma_{i,j}^2\right)^2\right) \tag{1}$$

It may be useful to train instead with mean squared error loss, which is calculated by taking an average of the sum of the squared difference between predicted and actual values as shown in Equation 2. In this equation, n refers to the number of data points, Y_i refers to the actual values, and $\hat{Y}_i$ refers to the predicted values.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2 \tag{2}$$

Also, we would like to work further on transforming the CAMELS dataset. As of now, we have reflected each map across the y-axis to, but other ways of transforming include random cropping, brightness, shifting, and implementing RandAugment [5]. In this project, we could only train over 100 epochs due to lack of time. We would like to dedicate more epochs and try other transform methods. Because MaxViT is a visual transformer, the number of trainable parameters is very high, resulting in running out of GPU memory for larger variations of MaxViT, while changing batch size to a small number was necessary when using MaxViT base with transformed datawwwwwwwwwwwwww. The immense time, memory, and data required to train MaxViT models makes it difficult to fully utilize all the visual transformer parameters.

5 Conclusions

The current results show that the MaxViT transformer is not able to outperform the scattering networks model for either the Ω_M or σ_8 parameter, but does achieve errors that are only marginally greater—with room for improvement as MaxViT is tuned more. The errors on fitting parameters indicate that for Ω_M and σ_8, scattering networks performs better and for $A_{SN1}, A_{SN2}, A_{AGN1}$, and A_{AGN2}, neither model is able to accurately constrain. Both, however, are able to outperform CNN by a large margin in all parameters. Our attempt at using transformed data had poor results, but further use of different transformations may be able to achieve better results.

We are very excited about these results as they perform very well especially for the limited time we had to optimize MaxViT. We find that MaxViT has tremendous potential in inferring cosmological parameters and with further hyperparameter tuning, we hope it has the ability to surpass the scattering networks model in estimating both parameters.

6 Lessons Learned

From start to finish, this project was a wonderful learning experience. We really enjoyed learning about cosmology and the different methods used to model and estimate cosmological parameters. One of the biggest issues we ran into was implementing a version of MaxViT model that was set up to use GPUs. The first model we tried to use was only partially set up to use GPUs and so we had to search for another version that works: lesson being do not trust everything you find on the internet/GitHub to work properly or as intended; we have learned caution and the ability to locate and deal with erroneous models. A new skill set developed through both Capstone Bootcamp and in initializing our model was creating stable PyTorch environments able to use kernels, as well as gaining familiarity with PyTorch, WandB, and models we have not had hands on experience with before such as CNN and SN—besides MaxViT itself. In addition, we ran into high training costs where running MaxViT on large sets of data took up to 50 hours, developing a consideration for time cost that was not present previously since we just used simple, relatively fast, models from previous classes. This consideration of time cost forced a greater importance on teamwork and coordination as managing training and tuning multiple algorithms at a time made division of work a necessity. Overall, we really enjoyed learning about new topics and methods while working on this project, and finding solutions and developing future consideration for problems that we ran into.

7 Student Contributions

All group members worked on setting up the environment for running the model and initial model training and testing. Seonhye Yang and Michael Healy worked on optimizing MaxViT and computing results; Bhavna Thakar worked on the poster and report. All members contributed equally.

8 Acknowledgements

We would like to thank and acknowledge Shirley Ho, Chris Pedersen, Bruno Regaldo Saint Blancard, and Michael Eickenberg for their invaluable mentorship and all the wonderful work, ideas, and time they put into helping us achieve success in this project.

References

[1] Pedersen, C., Eickenberg M., Ho S. (2022) Learnable wavelet neural networks for cosmological inference *ICML 2022 Workshop on Machine Learning for Astrophysics.* https://ml4astro.github.io/icml2022/assets/40.pdf

[2] Villaescusa-Navarro, F., Genel, S., Anglés-Alcázar, D., Thiele, L., Dave, R., Narayanan, D., Vogelsberger, M. (2022). The CAMELS Multifield Data Set: Learning the Universe's Fundamental Parameters with Artificial Intelligence. *The Astrophysical Journal Supplement Series*, 259(2), 61. doi:10.3847/1538-4365/ac5ab0

[3] Tu, Z., Talebi, H., Zhang, H., Yang, F., Milanfar, P., Bovik, A., Li, Y. (2022). MaxViT: Multi-Axis Vision Transformer. doi: 10.48550/ARXIV.2204.01697

[4] Zarka, J., Guth, F., Mallat, S., (2020). Separation and Concentration in Deep Networks. doi: 10.48550/arXiv.2012.10424

[5] Cubuk, E., Zoph, B., Shlens, J., Le, Q.. (2019). RandAugment: Practical automated data augmentation with a reduced search space.