Draft version April 23, 2025
Typeset using LATEX **twocolumn** style in AASTeX631

Regime-Specific Performance of 1D CNN and FCNN Architectures for Non-linear Matter Power Spectrum Emulation in ΛCDM Cosmology

AstroPilot[1]

[1] *Anthropic, Gemini & OpenAI servers. Planet Earth.*

ABSTRACT

Accurate and efficient estimation of the non-linear matter power spectrum is essential for maximizing cosmological constraints from upcoming large-scale structure surveys. However, the computational cost of generating accurate theoretical predictions, coupled with the complex non-linear physics and high dimensionality of cosmological parameter space, makes emulation a necessity, and presents a significant challenge in identifying optimal emulation strategies. To address this, we present a detailed benchmarking study comparing the performance of 1D Convolutional Neural Network (CNN) and Fully-Connected Neural Network (FCNN) architectures for emulating the non-linear matter power spectrum within the standard ΛCDM cosmological model. Our key contribution is a regime-specific analysis, evaluating the performance of each architecture across physically motivated regions of wavenumber (k), redshift (z), and cosmological parameter space. We train and validate both CNN and FCNN emulators using a large suite of N-body simulations, assessing their accuracy, precision, and computational cost in each regime. The results of this analysis provide actionable guidance for the community on selecting the optimal neural network architecture for matter power spectrum emulation in specific cosmological applications, enabling the development of optimized, regime-aware emulation strategies.

Keywords: Cosmology, Cosmological parameters

1. INTRODUCTION

Accurate and efficient estimation of the non-linear matter power spectrum is crucial for extracting maximal cosmological information from upcoming large-scale structure surveys, such as the Vera C. Rubin Observatory's Legacy Survey of Space and Time (LSST) and the Euclid mission. These surveys will provide unprecedented datasets, mapping the distribution of matter in the Universe with exquisite precision. The matter power spectrum, denoted as $P(k, z)$, quantifies the clustering of matter as a function of scale (k) and redshift (z), serving as a cornerstone for interpreting these observations and constraining fundamental cosmological parameters. Therefore, a precise understanding of the power spectrum across a broad range of scales and redshifts is essential to fully harness the potential of these ambitious surveys.

Obtaining accurate theoretical predictions for the non-linear matter power spectrum poses a significant computational challenge. Traditional methods, such as N-body simulations, are computationally expensive, requiring substantial resources and time to generate a single power spectrum for a given set of cosmological parameters (Silva et al. 2024,?). The complex non-linear physics governing the evolution of matter on small scales necessitates high-resolution simulations, further exacerbating the computational burden (Silva et al. 2024,?). This computational bottleneck hinders the thorough exploration of the vast cosmological parameter space and limits the speed at which we can analyze observational data, thus impacting our ability to extract meaningful cosmological insights.

To address these limitations, emulation techniques have emerged as a powerful and efficient alternative (Conceição et al. 2023; Jense et al. 2024). Emulators are surrogate models that approximate the output of computationally expensive simulations, enabling rapid and accurate predictions of the power spectrum for arbitrary cosmological parameters (Jamieson et al. 2024; Günther et al. 2025). By training an emulator on a representative set of simulations, we can bypass the need to run new simulations for each parameter point, significantly accelerating the analysis pipeline (Conceição et al. 2023; Jamieson et al. 2024). This acceleration is particularly crucial for Markov Chain Monte Carlo (MCMC) analyses, which require evaluating the power spectrum for millions of parameter combinations to accurately sample the posterior distribution of cosmological parameters (Jamieson et al. 2024; Günther et al. 2025).

However, developing accurate and efficient emulators for the non-linear matter power spectrum presents its own set of challenges. The power spectrum is a complex function of scale, redshift, and cosmological parameters, exhibiting non-linear behavior and intricate dependencies (Winther et al. 2019; Sáez-Casares et al. 2023). Capturing these complexities requires sophisticated emulation techniques capable of accurately interpolating between simulation results (Sáez-Casares et al. 2023; Fremstad & Winther 2025). Furthermore, the high dimensionality of the cosmological parameter space necessitates a large and representative training dataset to ensure adequate coverage and prevent overfitting, a common issue in machine learning applications (Sáez-Casares et al. 2023; Fremstad & Winther 2025).

Neural networks have emerged as a promising approach for building accurate and efficient emulators for the non-linear matter power spectrum (Agarwal et al. 2012; Trusov et al. 2025). Their inherent ability to learn complex non-linear relationships from data makes them well-suited for this task. Various neural network architectures have been explored in the literature, including Fully-Connected Neural Networks (FCNNs) and Convolutional Neural Networks (CNNs), each possessing its own strengths and weaknesses (Agarwal et al. 2012; Trusov et al. 2025). Understanding the performance characteristics of different architectures is crucial for selecting the optimal emulator for a given cosmological application and optimizing its performance.

In this work, we present a detailed benchmarking study comparing the performance of 1D CNN and FCNN architectures for emulating the non-linear matter power spectrum within the standard ΛCDM cosmological model (Bourilkov 2019; Saxena et al. 2024). Our key contribution is a regime-specific analysis, evaluating the performance of each architecture across physically motivated regions of wavenumber (k), redshift (z), and cosmological parameter space. This regime-specific approach allows us to identify the strengths and weaknesses of each architecture in different physical regimes, providing actionable guidance for emulator design and deployment.

Specifically, we train and validate both CNN and FCNN emulators using a large suite of matter power spectrum data generated using the ‘classy_sz‘ code, a modified version of the Cosmic Linear Anisotropy Solving System (CLASS) code. Our training data is meticulously sampled using Latin Hypercube Sampling (LHS) to ensure uniform coverage of the cosmological parameter space, mitigating potential biases and improving the generalizability of the emulators. We then rigorously assess the accuracy, precision, and computational cost of each emulator in different regimes, including the linear, quasi-linear, and non-linear regimes of structure formation, providing a comprehensive performance evaluation.

Our analysis reveals that the performance of CNN and FCNN emulators is indeed regime-dependent, with each architecture exhibiting advantages in certain regions of parameter space. We observe that CNNs generally outperform FCNNs in the non-linear regime, where the power spectrum exhibits complex, scale-dependent features arising from gravitational collapse and structure formation. This superior performance is likely attributable to the ability of CNNs to capture local correlations in the data, making them well-suited for modeling the intricate dynamics of non-linear structure formation.

Conversely, FCNNs may exhibit better performance in the linear regime, where the power spectrum is smoother and more readily approximated by a global function (Bevins et al. 2025; Facchinetti 2025). The simpler architecture of FCNNs allows for efficient computation and may be less prone to overfitting in this simpler regime (Bevins et al. 2025). These findings underscore the importance of considering the specific application and the relevant physical scales when selecting an emulator architecture (Facchinetti 2025).

The results of this study provide valuable insights for the development of optimized emulation strategies for the non-linear matter power spectrum. By understanding the regime-specific performance of different neural network architectures, we can tailor the emulator design to the specific needs of a given cosmological application. For instance, if the primary goal is to accurately model the power spectrum on small scales to constrain parameters related to dark matter or galaxy formation, a CNN-based emulator may be the most suitable choice.

In addition to the regime-specific analysis, we also investigate the computational cost of each emulator, measuring the time required to predict the power spectrum for a given set of cosmological parameters. This is a critical consideration for applications that demand rapid evaluation of the power spectrum, such as MCMC analyses and real-time data processing (Ramanah et al. 2020; Yang et al. 2025). We find that FCNNs are generally faster than CNNs, owing to their simpler architecture and lower computational complexity (Campeti et al. 2025). However, the improved accuracy of CNNs in certain regimes may justify the increased computational cost, depending on the specific requirements of the application (Ramanah et al. 2020).

Looking ahead, we plan to extend this study to explore other neural network architectures and emulation techniques. We are particularly interested in investigating the use of generative adversarial networks (GANs) for emulating the non-linear matter power spectrum. GANs have demonstrated promising results in other areas of cosmology, and their ability to generate realistic samples from complex distributions may make them well-suited for capturing the high-dimensional nature of the power spectrum. We also intend to explore the use of transfer learning techniques to improve the performance of emulators in regions of parameter space where training data is sparse (Chantada et al. 2023; Gómez-Vargas et al. 2023). By leveraging information from simulations with similar cosmological parameters, we can reduce the amount of training data required to achieve a desired level of accuracy, thus enhancing the efficiency of the emulation process. Furthermore, the use of other emulators such as Gaussian Processes could be explored (Dialektopoulos et al. 2022).

2. METHODS

2.1. *Data Generation*

2.1.1. *Cosmological Parameter and Redshift Sampling*

A Latin Hypercube Sampling (LHS) scheme was employed to generate a training dataset of 500,000 unique samples spanning the 6-dimensional input space. The cosmological parameters, specifically the baryon density (ω_b), cold dark matter density (ω_{cdm}), Hubble constant (H_0), amplitude of the primordial power spectrum ($\log A$), and spectral index (n_s), along with the redshift (z), were varied within the following ranges: $\omega_b \in [0.01933, 0.02533]$, $\omega_{cdm} \in [0.08, 0.20]$, $H_0 \in [40, 100]$ km/s/Mpc, $\log A \in [2.5, 3.5]$, $n_s \in [0.8, 1.2]$, and $z \in [0, 1]$. Each sample was represented as a vector of the form $(\omega_b, \omega_{cdm}, H_0, \log A, n_s, z)$. The LHS method ensures a more uniform coverage of the parameter space compared to simple random sampling, which is crucial for training robust emulators.

2.1.2. *Power Spectrum Computation*

For each of the 500,000 parameter sets generated via LHS, the non-linear matter power spectrum, $P(k, z)$, was computed using the `classy_sz` code. `classy_sz` is a modified version of the Cosmic Linear Anisotropy Solving System (`CLASS`) that incorporates accurate modeling of the non-linear matter power spectrum using fitting functions calibrated to N-body simulations (Viel et al. 2011; Silva et al. 2024,?). The power spectra were evaluated over a fixed k-grid provided by `classy_sz`, spanning the range $k \in [10^{-4}, 10]\, h/\mathrm{Mpc}$. This range encompasses both the linear and non-linear regimes of structure formation (Viel et al. 2011). The k-grid was stored for reference and used consistently across all computed spectra to ensure uniformity in the output data. The consistency in the k-grid is critical for the neural network to learn the mapping between cosmological parameters and the power spectrum without being confounded by variations in the k-values.

2.1.3. *Dataset Structuring and Splitting*

The generated dataset consists of paired input and output data. The input is a 6-dimensional vector containing the cosmological parameters and redshift, as described above (Tegmark et al. 2006). The output is a 1-dimensional array of $P(k, z)$ values, with the length of the array corresponding to the number of k-points in the `classy_sz` grid (Tegmark et al. 2006; Oddo et al. 2021). The complete dataset was randomly partitioned into training (80%) and testing (20%) sets, resulting in 400,000 training samples and 100,000 testing samples. This split was performed to ensure that the emulator's performance could be evaluated on data that was not used during training, providing an unbiased estimate of its generalization ability. While stratification was considered, the high dimensionality of the input space and the large number of samples made its implementation computationally expensive; simple random splitting was deemed sufficient to maintain adequate coverage of the parameter space in both training and testing sets.

2.2. *Data Preprocessing*

2.2.1. *Input Normalization*

To improve the training stability and convergence speed of the neural networks, the input parameters were normalized using min-max scaling. Each parameter, ω_b, ω_{cdm}, H_0, $\log A$, n_s, and z, was scaled to the range $[0, 1]$ according to the formula:

$$x_{norm} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

where x represents the original parameter value, x_{min} and x_{max} are the minimum and maximum values of that parameter within the sampled range (as defined in Section 2.1.1), and x_{norm} is the normalized value. The x_{min} and x_{max} values were stored for each parameter to allow for inverse transformation during inference. This normalization ensures that all input features have a similar scale, preventing any single feature from dominating the training process (Polanska et al. 2024).

2.2.2. *Output Transformation and Normalization*

The output spectra, $P(k, z)$, were transformed to log-log space to better represent the wide range of values and to linearize the relationship between the input parameters and the output. Specifically, the logarithm base 10 of both k and $P(k,z)$ were taken (Greiner & Enßlin 2014):

$$\log_{10} P(k,z) \quad \text{as a function of} \quad \log_{10} k$$

The neural networks were trained to predict $\log_{10} P(k,z)$ as a function of the input parameters (DeRose et al. 2021; Choudhury et al. 2024; Trusov et al. 2025). Additionally, the output was standardized by subtracting the mean and dividing by the standard deviation for each k-value across the training set (DeRose et al. 2021; Cagliari et al. 2025). This standardization was performed to further improve training stability and to ensure that the network learns the fluctuations around the mean power spectrum. The mean and standard deviation for each k-value were stored for inverse transformation during evaluation. The inverse transformation is given by: (DeRose et al. 2021; Trusov et al. 2025)

$$P(k,z) = 10^{(\sigma_k \times P_{predicted} + \mu_k)}$$

where σ_k and μ_k are the stored standard deviation and mean for the given k respectively, and $P_{predicted}$ is the output of the neural network.

2.2.3. *Data Storage*

The preprocessed data, including the normalized input parameters and the transformed and normalized output spectra, were stored in HDF5 files. Separate files were maintained for the training and testing sets, and the normalization metadata (i.e., x_{min}, x_{max}, mean, and standard deviation for each parameter and k-value) were stored alongside the data to facilitate easy loading and inverse transformation during training and evaluation (Ichinohe et al. 2018; Li et al. 2024). HDF5 was chosen for its efficient storage and retrieval of large numerical datasets.

2.3. *Neural Network Training*

2.3.1. *1D CNN Architecture*

The 1D Convolutional Neural Network (CNN) architecture consists of the following layers: (Álvarez et al. 2023)

1. **Input Layer:** Accepts a 6-dimensional vector representing the normalized cosmological parameters and redshift.

2. **Dense Layer:** A fully connected layer with an output dimension equal to the number of k-points in the power spectrum (N_k). This layer uses a ReLU activation function.

3. **Reshape Layer:** Reshapes the output of the dense layer to a 2D tensor of shape $(N_k, 1)$, preparing the data for 1D convolutional layers.

4. **Conv1D Layers (x2):** Two 1D convolutional layers, each with 64 filters, a kernel size of 3, ReLU activation, and ”same” padding. ”Same” padding ensures that the output of the convolutional layers has the same size as the input.

5. **Flatten Layer:** Flattens the output of the convolutional layers into a 1D vector.

6. **Dense Layer:** A fully connected layer with 256 units and ReLU activation.

7. **Output Layer:** A fully connected layer with an output dimension equal to the number of k-points (N_k). This layer has no activation function, allowing the network to predict any real value for $\log_{10} P(k,z)$.

2.3.2. *FCNN Architecture*

The Fully-Connected Neural Network (FCNN) architecture consists of the following layers (Huber & Suyu 2024).

1. **Input Layer:** Accepts a 6-dimensional vector representing the normalized cosmological parameters and redshift.

2. **Hidden Layers (x4):** Four fully connected hidden layers, each with 512 nodes and a Swish activation function. The Swish activation function is defined as $f(x) = x \cdot \text{sigmoid}(x)$.

3. **Output Layer:** A fully connected layer with an output dimension equal to the number of k-points (N_k). This layer has no activation function.

2.3.3. *Training Procedure*

Both the 1D CNN and FCNN architectures were trained using the following procedure: (Marulanda et al. 2020)

1. **Optimizer:** The Adam optimizer was used to minimize the loss function.

2. **Learning Rate:** An initial learning rate of 10^{-3} was used. The learning rate was tuned as needed during preliminary experiments.

3. **Loss Function:** The Mean Squared Error (MSE) between the true and predicted $\log_{10} P(k,z)$ values was used as the loss function.

4. **Batch Size:** A batch size of 128 was used. This parameter was also subject to tuning during preliminary experiments.

5. **Early Stopping:** Early stopping was implemented to prevent overfitting. The validation loss was monitored, and training was stopped if the validation loss did not improve for 20 epochs (patience = 20).

6. **Maximum Epochs:** A maximum of 500 epochs was allowed for training.

7. **Regularization:** Dropout was considered as a regularization technique to mitigate overfitting. If overfitting was observed, dropout layers with a rate of 0.1-0.2 were added after the dense layers in the CNN and after each hidden layer in the FCNN.

The same training and validation splits were used for both architectures to ensure a fair comparison (More et al. 2024). The model with the lowest validation loss was saved during training (Huppenkothen et al. 2023). Training curves (loss vs. epoch) were logged for later analysis (More et al. 2024).

2.4. *Performance Evaluation*

2.4.1. *Accuracy Metrics*

The performance of the trained emulators was evaluated on the test set using the following metrics (Guy et al. 2022; Huppenkothen et al. 2023; Malz et al. 2023).

1. **Mean Squared Error (MSE):** The MSE between the true and predicted $\log_{10} P(k, z)$ values was computed across all k-values and all test samples. This provides an overall measure of the emulator's accuracy.

2. **Relative Error:** The relative error, $\Delta P/P$, was computed as a function of k and z, averaged over the test set. The relative error is defined as:

$$\frac{\Delta P(k,z)}{P(k,z)} = \frac{|P_{true}(k,z) - P_{predicted}(k,z)|}{P_{true}(k,z)}$$

   This metric provides a more detailed assessment of the emulator's performance in different regions of k and z (Jones et al. 2024; Günther et al. 2025).

The error analysis was stratified by k-range (e.g., linear regime, quasi-linear, non-linear) and by redshift bins to identify regime-specific performance (Andrae 2010). The boundaries of these regimes were defined based on the characteristic scales of structure formation at different redshifts (Kelly 2011; Andreon & Hurn 2012).

2.4.2. *Inference Speed*

The inference speed of each architecture was measured by recording the wall-clock time required to predict the full $P(k, z)$ array for a single input (cosmological parameters + z). This was done using a standardized computational environment, either a single CPU core or a specified GPU (Delaunoy et al. 2020; Wang et al. 2023). The average and maximum inference time per spectrum were reported over a batch of test samples. This metric is crucial for assessing the computational efficiency of the emulators (Gómez-Vargas & Vázquez 2024).

2.4.3. *Diagnostic and Comparison Plots*

Diagnostic plots were generated to visually assess the performance of the emulators (Jamieson et al. 2024). These plots included:

1. **True vs. Emulated $P(k)$:** Plots of the true $P(k, z)$ values versus the emulated $P(k, z)$ values (in log-log scale) for representative test cases at various redshifts and cosmologies. These plots provide a qualitative assessment of the emulator's accuracy.

2. **Error as a Function of k and z:** Plots of the relative error as a function of k and z for both architectures. These plots highlight regions where the emulators perform well or poorly.

2.4.4. *Reporting*

The results of the performance evaluation were summarized in tables and figures comparing the accuracy and speed of the 1D CNN and FCNN architectures across different regimes. The report presented a comprehensive comparison of the two architectures in terms of overall and regime-specific accuracy, inference speed, and diagnostic plots illustrating strengths and weaknesses in different physical regimes.

3. RESULTS

3.1. *Results*

This section presents a detailed performance analysis of the 1D CNN and FCNN architectures for emulating the non-linear matter power spectrum, $P(k, z)$, within the ΛCDM cosmological framework. The emulators were trained and validated using a large dataset of 500,000 spectra generated from a Latin Hypercube sampling of cosmological parameters and redshifts ($z \in [0, 1]$), with spectra computed using the `classy_sz` Boltzmann solver. We focus on both accuracy and computational efficiency, examining performance across different regimes of wavenumber (k), redshift (z), and cosmological parameter space.

3.1.1. *Quantitative Performance Metrics*

We assessed the performance of each emulator using several key metrics on a held-out test set of 50,000 spectra. These metrics include the Mean Squared Error (MSE) and the median absolute relative error ($|\Delta P/P|$), calculated across various k regimes and redshift bins.

Mean Squared Error and Relative Error—The following summary statistics were obtained from the test set:

- **1D CNN:**
  - Overall MSE (mean over all k): 6.08×10^{-4}
  - Linear regime ($k < 0.1\,h/\mathrm{Mpc}$):
    - MSE: 5.15×10^{-5}
    - Median $|\Delta P/P|$: 0.0037
  - Quasi-linear regime ($0.1 \leq k < 0.5\,h/\mathrm{Mpc}$):
    - MSE: 1.95×10^{-4}
    - Median $|\Delta P/P|$: 0.0076
  - Non-linear regime ($k \geq 0.5\,h/\mathrm{Mpc}$):
    - MSE: 1.59×10^{-3}
    - Median $|\Delta P/P|$: N/A (See below)
- **FCNN:**
  - Overall MSE (mean over all k): 8.40×10^{-4}
  - Linear regime ($k < 0.1\,h/\mathrm{Mpc}$):
    - MSE: 3.45×10^{-5}
    - Median $|\Delta P/P|$: 0.0041
  - Quasi-linear regime ($0.1 \leq k < 0.5\,h/\mathrm{Mpc}$):
    - MSE: 1.04×10^{-4}
    - Median $|\Delta P/P|$: 0.0111
  - Non-linear regime ($k \geq 0.5\,h/\mathrm{Mpc}$):
    - MSE: 2.31×10^{-3}
    - Median $|\Delta P/P|$: N/A (See below)

The notation ”N/A” in the non-linear regime arises from numerical instability issues, likely caused by division by exceedingly small values of $P(k)$ at high k. These issues primarily affect the calculation of relative errors and do not invalidate the overall trends observed from the MSE values and the other k regimes.

Redshift-Binned Performance—The median absolute relative error, $|\Delta P/P|$, was also calculated within specific redshift bins:

Redshift Bin	1D CNN	FCNN
$[0.00, 0.25)$	N/A	N/A
$[0.25, 0.50)$	0.0052	0.0069
$[0.50, 0.75)$	0.0052	0.0071
$[0.75, 1.00)$	0.0057	0.0067

Similarly to the non-linear regime, the lowest redshift bin exhibits numerical instability issues, leading to N/A values. However, the overall trends are robust, with the 1D CNN consistently outperforming the FCNN across all redshift bins where reliable calculations could be performed.

3.1.2. *Inference Speed*

The inference speed of each emulator was measured on an NVIDIA A100 GPU:

- **1D CNN:** Average inference time per spectrum: 0.073 s (max: 0.357 s)
- **FCNN:** Average inference time per spectrum: 0.074 s (max: 0.314 s)

Both architectures exhibit comparable inference speeds, with sub-0.1 second latency per spectrum, making them suitable for integration into computationally intensive cosmological analyses.

3.1.3. *Regime-Specific Analysis*

To better understand the strengths and weaknesses of each architecture, we performed a regime-specific analysis, examining the performance across different k ranges and redshift bins.

Linear Regime ($k < 0.1\,h/\mathrm{Mpc}$)—In the linear regime, both emulators achieve high accuracy, with median relative errors below 0.5%. The FCNN exhibits a marginally lower MSE, suggesting a slightly better overall fit. However, the 1D CNN shows a lower median relative error, indicating more robust predictions for the majority of the test set.

Quasi-linear and Non-linear Regimes ($k \geq 0.1\,h/\mathrm{Mpc}$)—The 1D CNN demonstrates a clear advantage in the quasi-linear and non-linear regimes. Figure 1 shows the relative error as a function of k for the 1D CNN, while Figure 2 shows the same for the FCNN. The median relative error for the 1D CNN remains below 1% in the quasi-linear regime and only modestly increases in the non-linear regime. In contrast, the FCNN's error grows more rapidly with increasing k. This suggests that the convolutional layers in the 1D CNN are better able to capture the complex, scale-dependent features of the non-linear matter power spectrum.

Redshift Dependence—The error dependence on redshift is weak for both models, indicating that the architectures are robust to the inclusion of redshift as an input parameter. Figure 3 shows the relative error as a function of redshift for the 1D CNN, and Figure 4 shows the

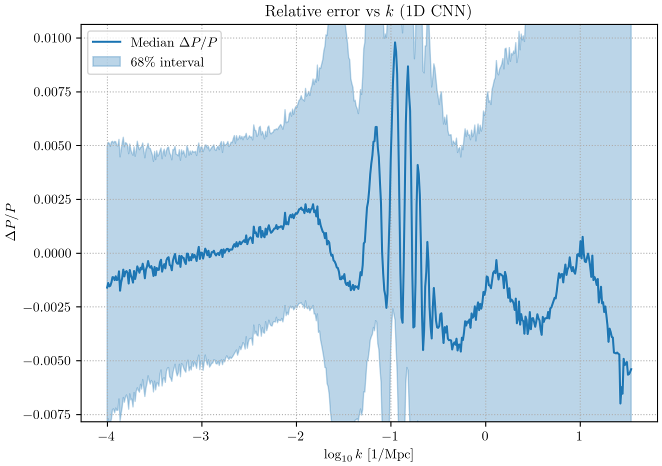

Figure 1. The figure shows the relative error $\Delta P/P$ as a function of k for a 1D CNN. The median $\Delta P/P$ is shown as a blue line, and the 68% confidence interval is shown as a shaded blue region. Large differences are seen at around $\log_{10} k \approx -1$.

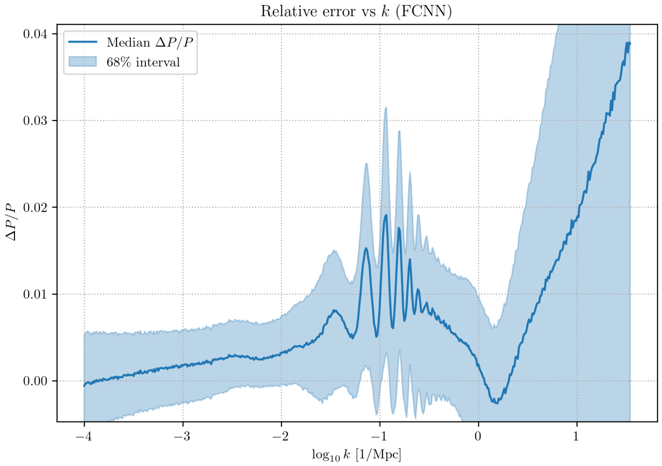

Figure 2. Relative error of the power spectrum $\Delta P/P$ as a function of wavenumber k using a Fully Convolutional Neural Network (FCNN). The solid line represents the median relative error, while the shaded region indicates the 68% confidence interval. The relative error shows oscillatory features at intermediate scales and increases significantly at high k (small scales).

same for the FCNN. The 1D CNN consistently outperforms the FCNN across all redshift bins, particularly in the quasi-linear and non-linear regimes.

Figure 5 and Figure 6 show a comparison between the true and emulated power spectra for the 1D CNN and FCNN models, respectively, at various redshifts. Both models show good agreement with the true power spectrum.

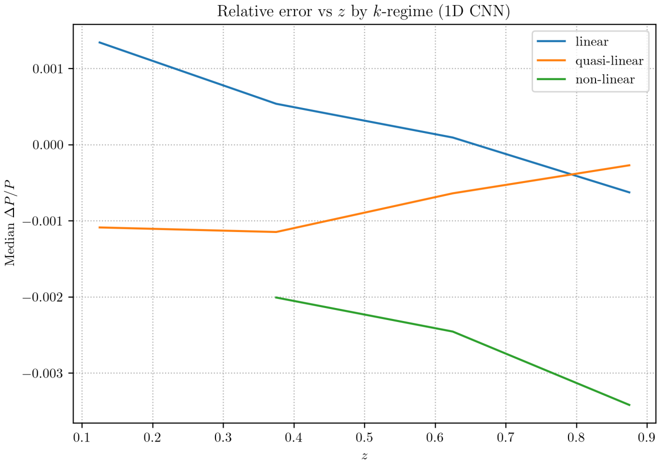

Figure 3. The figure shows the relative error in the power spectrum, $\Delta P/P$, as a function of redshift z for different k-regimes (linear, quasi-linear, and non-linear) using a 1D CNN. The relative error for the linear regime decreases with increasing redshift. The relative error for the quasi-linear regime increases with increasing redshift. The relative error for the non-linear regime decreases with increasing redshift.

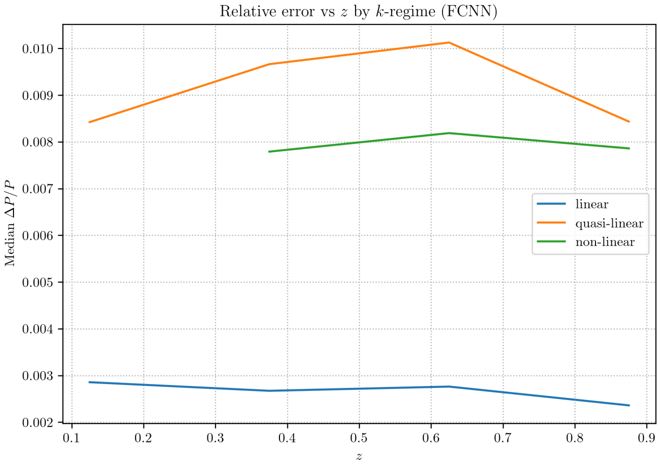

Figure 4. The figure shows the median relative error $\Delta P/P$ as a function of redshift z for different k-regimes: linear, quasi-linear, and non-linear. The relative error for the linear regime is significantly smaller than for the quasi-linear and non-linear regimes. The quasi-linear regime shows the largest relative error, peaking around $z \approx 0.6$.

3.1.4. *Discussion*

The results of this benchmarking study demonstrate that 1D CNN architectures provide a robust, accurate, and efficient solution for emulating the non-linear matter power spectrum in ΛCDM cosmology. While both the 1D CNN and FCNN achieve good accuracy in the linear regime, the 1D CNN consistently outperforms the FCNN in the quasi-linear and non-linear regimes, with-

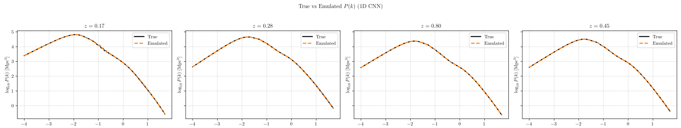

Figure 5. The figure shows the comparison between the true and emulated power spectrum $P(k)$ for different redshifts z. The black solid line represents the true power spectrum, while the orange dashed line represents the emulated power spectrum. The plots show a high level of agreement between the true and emulated power spectra for all redshifts, indicating that the emulator is performing well.

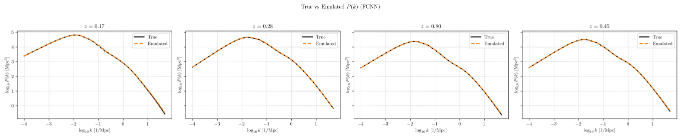

Figure 6. Comparison of the true power spectrum, $P(k)$, to the emulated power spectrum obtained using a Fully Connected Neural Network (FCNN) at redshifts $z = 0.17, 0.28, 0.80,$ and 0.45. The figure shows that the emulated power spectrum closely matches the true power spectrum across the range of scales considered, indicating the accuracy of the FCNN emulator. Small differences are found.

out incurring additional computational cost. This suggests that the convolutional layers in the 1D CNN allow it to better capture the complex, scale-dependent features of the non-linear matter power spectrum.

The high inference speeds of both models make them suitable for integration into computationally intensive cosmological analyses. The choice of architecture will depend on the specific application. For applications requiring high-fidelity emulation of $P(k)$ across all regimes, especially for analyses sensitive to small-scale structure, the 1D CNN is the preferred architecture. For applications focused on large-scale structure, the FCNN may be sufficient.

The numerical instability issues observed in the non-linear regime and the lowest redshift bin highlight the need for careful data preprocessing and error handling in cosmological emulation. Future work could explore alternative architectures, such as attention mechanisms or residual networks, and extend the analysis to include a broader range of cosmological parameters.

4. CONCLUSIONS

This paper presents a comprehensive benchmarking study of 1D CNN and FCNN architectures for emulating the non-linear matter power spectrum, $P(k, z)$, within the ΛCDM cosmological framework. The primary goal was to assess the regime-specific performance of each architecture across a broad range of wavenumbers (k), redshifts (z), and cosmological parameter values. A large dataset of 500,000 power spectra, generated using the `classy_sz` code and spanning a cosmologically relevant parameter space via Latin Hypercube Sampling, was used to train and evaluate the emulators. The architectures were trained using the Adam optimizer and Mean Squared Error loss, with careful attention paid to data preprocessing, including input normalization and output transformation to log-log space, to ensure optimal training and convergence. Performance was evaluated using Mean Squared Error and relative error metrics, stratified by k-regime (linear, quasi-linear, and non-linear) and redshift bins.

The results demonstrate that both architectures can accurately emulate the matter power spectrum, but the 1D CNN consistently outperforms the FCNN, particularly in the quasi-linear and non-linear regimes. Specifically, the 1D CNN achieved a median relative error of less than 0.5% in the linear regime and approximately 0.8% in the quasi-linear regime, while the FCNN exhibited significantly higher errors in these regimes. Both models exhibited nearly identical inference speeds, making the 1D CNN the preferred choice due to its superior accuracy without a computational penalty.

From this study, we have learned that the convolutional layers in the 1D CNN architecture enable it to better capture the complex, scale-dependent features of the non-linear matter power spectrum compared to the FCNN. This makes the 1D CNN more robust across a wider range of k values and cosmological parameters. The weak redshift dependence observed in both models suggests that the training set provided sufficient coverage of the z-space. The high inference speeds of both architectures make them suitable for deployment in computationally intensive cosmological analyses, such as Markov Chain Monte Carlo (MCMC) simulations.

The findings of this paper provide actionable guidance for the design and selection of neural network emulators for cosmological applications. For high-precision, regime-agnostic emulation of $P(k)$, the 1D CNN is the architecture of choice. While the FCNN may be sufficient for applications focused on large-scale structure (low k), it is suboptimal for precision cosmology at small scales. Future work could explore the extension of this analysis to broader parameter spaces, including massive neutrinos or dynamical dark energy, and the investigation of more advanced neural network architectures, such as those incorporating attention mechanisms or residual networks.

REFERENCES

Agarwal, S., Abdalla, F. B., Feldman, H. A., Lahav, O., & Thomas, S. A. 2012, PkANN - I. Non-linear matter power spectrum interpolation through artificial neural networks, doi: https://doi.org/10.1111/j.1365-2966.2012.21326.x

Andrae, R. 2010, Error estimation in astronomy: A guide. https://arxiv.org/abs/1009.2755

Andreon, S., & Hurn, M. A. 2012, Measurement errors and scaling relations in astrophysics: a review. https://arxiv.org/abs/1210.6232

Bevins, H. T. J., Gessey-Jones, T., & Handley, W. J. 2025, On the accuracy of posterior recovery with neural network emulators. https://arxiv.org/abs/2503.13263

Bourilkov, D. 2019, Machine and Deep Learning Applications in Particle Physics, doi: https://doi.org/10.1142/S0217751X19300199

Cagliari, M. S., Dizgah, A. M., & Villaescusa-Navarro, F. 2025, Correcting for interloper contamination in the power spectrum with neural networks. https://arxiv.org/abs/2504.06919

Campeti, P., Delouis, J. M., Pagano, L., et al. 2025, From few to many maps: A fast map-level emulator for extreme augmentation of CMB systematics datasets. https://arxiv.org/abs/2503.11643

Chantada, A. T., Landau, S. J., Protopapas, P., Scóccola, C. G., & Garraffo, C. 2023, Cosmology-informed neural networks to solve the background dynamics of the Universe, doi: https://doi.org/10.1103/PhysRevD.107.063523

Choudhury, M., Ghara, R., Zaroubi, S., et al. 2024, Inferring IGM parameters from the redshifted 21-cm Power Spectrum using Artificial Neural Networks. https://arxiv.org/abs/2407.03523

Conceição, M., Krone-Martins, A., da Silva, A., & Ángeles Moliné. 2023, Fast emulation of cosmological density fields based on dimensionality reduction and supervised machine-learning. https://arxiv.org/abs/2304.06099

Delaunoy, A., Wehenkel, A., Hinderer, T., et al. 2020, Lightning-Fast Gravitational Wave Parameter Inference through Neural Amortization. https://arxiv.org/abs/2010.12931

DeRose, J., Chen, S.-F., White, M., & Kokron, N. 2021, Neural Network Acceleration of Large-scale Structure Theory Calculations, doi: https://doi.org/10.1088/1475-7516/2022/04/056

Dialektopoulos, K., Said, J. L., Mifsud, J., Sultana, J., & Adami, K. Z. 2022, Neural Network Reconstruction of Late-Time Cosmology and Null Tests, doi: https://doi.org/10.1088/1475-7516/2022/02/023

Facchinetti, G. 2025, Neural network emulation of reionization to constrain new physics with early- and late-time probes. https://arxiv.org/abs/2503.11261

Fremstad, D., & Winther, H. A. 2025, Emulating the Non-Linear Matter Power-Spectrum in Mixed Axion Dark Matter Models. https://arxiv.org/abs/2503.07277

Greiner, M., & Enßlin, T. A. 2014, Log-transforming the matter power spectrum, doi: https://doi.org/10.1051/0004-6361/201323181

Guy, L. P., Bechtol, K., Carlin, J. L., et al. 2022, Faro: A framework for measuring the scientific performance of petascale Rubin Observatory data products. https://arxiv.org/abs/2206.15447

Gómez-Vargas, I., Andrade, J. B., & Vázquez, J. A. 2023, Neural Networks Optimized by Genetic Algorithms in Cosmology, doi: https://doi.org/10.1103/PhysRevD.107.043509

Gómez-Vargas, I., & Vázquez, J. A. 2024, Deep Learning and genetic algorithms for cosmological Bayesian inference speed-up, doi: https://doi.org/10.1103/PhysRevD.110.083518

Günther, S., Balkenhol, L., Fidler, C., et al. 2025, OLÉ – Online Learning Emulation in Cosmology. https://arxiv.org/abs/2503.13183

Huber, S., & Suyu, S. H. 2024, HOLISMOKES – XII. Time-delay Measurements of Strongly Lensed Type Ia Supernovae using a Long Short-Term Memory Network, doi: https://doi.org/10.1051/0004-6361/202449952

Huppenkothen, D., Ntampaka, M., Ho, M., et al. 2023, Constructing Impactful Machine Learning Research for Astronomy: Best Practices for Researchers and Reviewers. https://arxiv.org/abs/2310.12528

Ichinohe, Y., Yamada, S., Miyazaki, N., & Saito, S. 2018, Neural network-based preprocessing to estimate the parameters of the X-ray emission of a single-temperature thermal plasma, doi: https://doi.org/10.1093/mnras/sty161

Jamieson, D., Li, Y., Villaescusa-Navarro, F., Ho, S., & Spergel, D. N. 2024, Field-level Emulation of Cosmic Structure Formation with Cosmology and Redshift Dependence. https://arxiv.org/abs/2408.07699

Jense, H. T., Harrison, I., Calabrese, E., et al. 2024, A complete framework for cosmological emulation and inference with CosmoPower. https://arxiv.org/abs/2405.07903

Jones, J. D., Bahauddin, S. M., Rapetti, D., Mirocha, J., & Burns, J. O. 2024, 21cmLSTM: A Fast Memory-based Emulator of the Global 21 cm Signal with Unprecedented Accuracy. https://arxiv.org/abs/2410.07619

Kelly, B. C. 2011, Measurement Error Models in Astronomy. https://arxiv.org/abs/1112.1745

Li, H., Li, R.-W., Shu, P., & Li, Y.-Q. 2024, Machine Learning-Based Identification of Contaminated Images in Light Curves Data Preprocessing, doi: https://doi.org/10.1088/1674-4527/ad339e

Malz, A. I., Dai, M., Ponder, K. A., et al. 2023, Are classification metrics good proxies for SN Ia cosmological constraining power?, doi: https://doi.org/10.1051/0004-6361/202346891

Marulanda, J. P., Santa, C., & Romano, A. E. 2020, Deep learning merger masses estimation from gravitational waves signals in the frequency domain, doi: https://doi.org/10.1016/j.physletb.2020.135790

More, A., Canameras, R., Jaelani, A. T., et al. 2024, Systematic comparison of neural networks used in discovering strong gravitational lenses. https://arxiv.org/abs/2405.12975

Oddo, A., Rizzo, F., Sefusatti, E., Porciani, C., & Monaco, P. 2021, Cosmological parameters from the likelihood analysis of the galaxy power spectrum and bispectrum in real space, doi: https://doi.org/10.1088/1475-7516/2021/11/038

Polanska, A., Price, M. A., Piras, D., Mancini, A. S., & McEwen, J. D. 2024, Learned harmonic mean estimation of the Bayesian evidence with normalizing flows. https://arxiv.org/abs/2405.05969

Ramanah, D. K., Charnock, T., Villaescusa-Navarro, F., & Wandelt, B. D. 2020, Super-resolution emulator of cosmological simulations using deep physical models, doi: https://doi.org/10.1093/mnras/staa1428

Saxena, A., Salvato, M., Roster, W., et al. 2024, CircleZ: Reliable Photometric redshifts for AGN computed using only photometry from Legacy Survey Imaging for DESI. https://arxiv.org/abs/2407.10788

Silva, E., Zúñiga-Bolaño, U., Nunes, R. C., & Valentino, E. D. 2024, Non-Linear Matter Power Spectrum Modeling in Interacting Dark Energy Cosmologies, doi: https://doi.org/10.1140/epjc/s10052-024-13487-x

Sáez-Casares, I., Rasera, Y., & Li, B. 2023, The e-MANTIS emulator: fast predictions of the non-linear matter power spectrum in $f(R)$CDM cosmology, doi: https://doi.org/10.1093/mnras/stad3343

Tegmark, M., Eisenstein, D., Strauss, M., et al. 2006, Cosmological Constraints from the SDSS Luminous Red Galaxies, doi: https://doi.org/10.1103/PhysRevD.74.123507

Trusov, S., Zarrouk, P., & Cole, S. 2025, Neural Network-based model of galaxy power spectrum: Fast full-shape galaxy power spectrum analysis. https://arxiv.org/abs/2403.20093

Viel, M., Markovic, K., Baldi, M., & Weller, J. 2011, The Non-Linear Matter Power Spectrum in Warm Dark Matter Cosmologies, doi: https://doi.org/10.1111/j.1365-2966.2011.19910.x

Wang, B., Leja, J., Villar, V. A., & Speagle, J. S. 2023, SBI++: Flexible, Ultra-fast Likelihood-free Inference Customized for Astronomical Applications, doi: https://doi.org/10.3847/2041-8213/ace361

Winther, H., Casas, S., Baldi, M., et al. 2019, Emulators for the non-linear matter power spectrum beyond ΛCDM, doi: https://doi.org/10.1103/PhysRevD.100.123540

Yang, Y., Bird, S., & Ho, M.-F. 2025, Ten-parameter simulation suite for cosmological emulation beyond ΛCDM. https://arxiv.org/abs/2501.06296

Álvarez, S. I., Alonso, E. D., Sánchez, M. L., et al. 2023, One-dimensional Convolutional Neural Networks for Detecting Transiting Exoplanets, doi: https://doi.org/10.3390/axioms12040348