V. Experiments and Results

5.1 Training Process:

5.1.1 Training and Validation Curves:

[Placeholder for plots of training and validation loss vs. epoch]. The plot should show both the training loss and the validation loss as a function of the training epoch. The analysis should discuss the following:

- Overall trend of the training loss: Does it decrease consistently, indicating successful learning?
- Overall trend of the validation loss: Does it decrease and then plateau, or does it start to increase after a certain point (indicating overfitting)?
- Comparison of training and validation loss: Is there a significant gap between the two curves, suggesting overfitting? If so, what measures could be taken to address this (e.g., regularization, dropout, reducing network complexity)?
- Optimal number of epochs: Based on the validation loss, what is the optimal number of epochs to train the network?

5.1.2 Hyperparameter Optimization Results:

[Placeholder for a table summarizing the hyperparameter optimization results]. The table should include:

- A list of the hyperparameters that were tuned (learning rate, batch size, optimizer).
- The search space for each hyperparameter.
- The best hyperparameter values found by Optuna.
- The corresponding validation loss (MSE) for the best hyperparameter combination.
- A brief discussion of the results: Were the optimal hyperparameters within the expected ranges? Did the choice of optimizer significantly affect performance?

5.1.3 Training Time:

Report the total training time for the best hyperparameter combination. Specify the hardware used (e.g., number of GPUs and their type).

5.2 Emulator Performance:

5.2.1 Evaluation Metrics:

The following metrics were used to evaluate the emulator's performance on the held-out test set:

- **Mean Squared Error (MSE):** The average squared difference between the predicted and true log-transformed, normalized power spectra.
- **Mean Absolute Percentage Error (MAPE):** The average absolute percentage difference between the predicted and true *untransformed* power spectra. This metric provides a more intuitive measure of the error in terms of percentage deviation. The formula is:

```
MAPE = (100/N) * ∑ |(P<sub>true</sub> - P<sub>predicted</sub>) / P<sub>true</sub>|
```

where N is the number of data points, P_{true} is the true power spectrum from CAMB, and $P_{predicted}$ is the emulator's prediction. Note: We calculate MAPE on the *untransformed* P(k) values, not the log-transformed ones.

- **R-squared (coefficient of determination):** A measure of how well the emulator's predictions explain the variance in the true power spectra. An R-squared value of 1 indicates a perfect fit.
- **Maximum Error:** The maximum absolute percentage error observed on the test set.

5.2.2 Quantitative Results:

[Placeholder for a table summarizing the performance metrics on the test set]. The table should present the values of MSE, MAPE, R-squared, and Maximum Error. Discuss the results: Does the emulator meet the target accuracy (e.g., MAPE < 1%)? Is the R-squared value close to 1? Is the maximum error acceptable?

5.2.3 Visual Comparison:

[Placeholder for plots comparing the emulated P(k) with the CAMB-generated P(k) for several randomly selected parameter combinations]. These plots should show:

- The true P(k) from CAMB (e.g., as a solid line).
- The emulated P(k) (e.g., as a dashed line).
- The percentage error as a function of k (e.g., in a separate panel below the main plot).
- The cosmological parameters for each plot should be clearly indicated.
- The plots should cover a diverse range of cosmological parameters to demonstrate the emulator's performance across different scenarios. Include examples where the emulator performs well and, if any, examples where it performs less well.

5.2.4 Error Distribution:

[Placeholder for a histogram of the errors (e.g., percentage errors)]. The histogram should show the distribution of the percentage errors across the entire test set. Analyze the distribution: Is it approximately Gaussian? Are there any significant outliers? Is the distribution skewed?

5.2.5 Performance vs. Cosmological Parameters:

[Placeholder for plots showing the error as a function of different cosmological parameters]. These plots should investigate whether the emulator's performance is dependent on specific parameter values. For each cosmological parameter, create a scatter plot where:

- The x-axis represents the value of the cosmological parameter.
- The y-axis represents the MAPE (or another error metric) for that particular prediction.
- Each point represents a single prediction from the test set.
- Analyze the plots: Are there any trends or correlations between the error and any of the cosmological parameters? Are there regions of the parameter space where the emulator performs significantly better or worse?

VI. Discussion

6.1 Interpretation of Results:

Provide an overall interpretation of the emulator's performance based on the results presented in Section V. Discuss whether the emulator meets the desired accuracy and performance goals. Relate the findings to the theoretical background: Does the emulator accurately capture the key features of the matter power spectrum (turnover, BAO, damping tail)?

6.2 Challenges and Limitations:

Discuss any challenges encountered during the project, such as difficulties in training the network, achieving the desired accuracy, or dealing with computational limitations. Identify the limitations of the current emulator:

- Range of validity: The emulator is only valid within the range of cosmological parameters used for training. Extrapolating beyond this range may lead to inaccurate predictions.
- Specific parameter combinations: Are there any specific combinations of cosmological parameters where the emulator performs poorly? Why might this be the case?
- Linear power spectrum: The current emulator is trained on the *linear* matter power spectrum. It does not account for non-linear effects, which become important at small scales.

6.3 Comparison with Theoretical Expectations:

Compare the emulator's predictions with theoretical expectations from linear perturbation theory. For example, discuss how well the emulator captures the expected scaling of P(k) with cosmological parameters (e.g., the dependence of the amplitude on σ_8). Discuss any discrepancies and potential explanations (e.g., limitations of linear theory, inaccuracies in the emulator).

6.4 Comparison with other emulators/methods:

If other emulators or methods for calculating the matter power spectrum exist, compare the performance of the developed FCNN emulator with them. Discuss advantages and disadvantages in terms of accuracy, speed, and complexity.

VII. Conclusion

7.1 Summary of Work:

Provide a concise summary of the project, reiterating the main findings. Summarize the methodology, implementation, and results. State whether the project objectives were achieved.

7.2 Future Directions:

Suggest potential future research directions, including:

- **Exploring different network architectures:** Investigate other neural network architectures, such as convolutional neural networks (CNNs) or recurrent neural networks (RNNs), which might be better suited for capturing the spatial correlations in the matter density field.
- **Incorporating more cosmological parameters:** Extend the emulator to include additional cosmological parameters, such as the neutrino mass or parameters describing dark energy models beyond a cosmological constant.
- **Extending the emulator to higher redshifts:** Train the emulator on power spectra at higher redshifts to enable studies of the evolution of structure formation.
- **Developing emulators for other cosmological observables:** Develop emulators for other cosmological observables, such as the CMB power spectrum or weak lensing shear power spectrum.
- **Using the emulator for parameter inference:** Integrate the emulator into a Bayesian inference framework (e.g., Markov Chain Monte Carlo) to constrain cosmological parameters from observational data.

- **Investigating methods for uncertainty quantification:** Develop methods to quantify the uncertainty in the emulator's predictions. This could involve using Bayesian neural networks or ensemble methods.
- **Non-linear matter power spectrum:** Extend the emulator to the non-linear matter power spectrum.

7.3 Final Remarks:

Offer concluding remarks on the significance of the work and its potential impact on cosmological research. Highlight the potential of machine learning techniques for accelerating cosmological simulations and enabling faster and more efficient exploration of the cosmological parameter space.