Outline for the Report on Emulation of the Matter Power Spectrum in Cosmology Using FCNN

I. Introduction (approx. 500 words)

- **1.1 Motivation:**
  - Briefly introduce the importance of understanding the large-scale structure of the universe.
  - Highlight the role of the matter power spectrum in characterizing this structure.
  - Explain the challenges of computationally expensive cosmological simulations.
  - Introduce the concept of emulation as a faster alternative.
  - State the need for accurate and efficient emulators for the matter power spectrum. This section motivates the entire project.
- **1.2 Objectives:**
  - Clearly state the primary goal: to develop a FCNN emulator for the matter power spectrum.
  - Specify the target accuracy and performance metrics.
  - Mention the use of CAMB for dataset generation and Jax for implementation.
  - List the key cosmological parameters considered.
- **1.3 Project Overview:**
  - Provide a high-level summary of the entire project.
  - Briefly describe the dataset, network architecture, training procedure, and evaluation methods.
  - Outline the structure of the report, summarizing each subsequent section.

II. Theoretical Background (approx. 1000 words)

- **2.1 Cosmology:**
  - **2.1.1 The Expanding Universe:** Describe the Friedmann-Lemaître-Robertson-Walker (FLRW) metric, Hubble's Law, and the concept of redshift.
  - **2.1.2 Cosmological Parameters:** Define and explain the key cosmological parameters that influence the matter power spectrum, such as:
    - Ω_m (matter density parameter)
    - Ω_b (baryon density parameter)
    - Ω_Λ (dark energy density parameter)
    - h (Hubble constant)
    - σ_8 (amplitude of matter fluctuations)
    - n_s (spectral index)

        - (Mention any other relevant parameters used in the dataset)
    - **2.1.3 Structure Formation:** Explain the process of structure formation from primordial density fluctuations, including gravitational instability and the growth of perturbations.
    - **2.1.4 Linear Perturbation Theory:** Introduce the mathematical framework for describing the evolution of small density perturbations. Derive the linear perturbation equations.
    - **2.1.5 Transfer Function:** Define the transfer function and its role in relating the initial conditions to the matter power spectrum at later times.
- **2.2 The Matter Power Spectrum:**
    - **2.2.1 Definition:** Provide a rigorous definition of the matter power spectrum, P(k), as the Fourier transform of the two-point correlation function of the density field. Explain its statistical interpretation.
    - **2.2.2 Importance:** Explain why the matter power spectrum is a crucial tool in cosmology, connecting theory with observations. Discuss how it can be used to constrain cosmological parameters and test different cosmological models.
    - **2.2.3 Features:** Describe the key features of the matter power spectrum, such as the turnover scale, the baryon acoustic oscillations (BAO), and the damping tail. Explain the physical origin of these features.
    - **2.2.4 CAMB:** Briefly introduce the Code for Anisotropies in the Microwave Background (CAMB) as a tool for calculating the matter power spectrum.
- **2.3 Neural Networks:**
    - **2.3.1 Introduction to Artificial Neural Networks:** Provide a general overview of artificial neural networks, their biological inspiration, and their applications in various fields.
    - **2.3.2 Fully Connected Neural Networks (FCNNs):** Describe the architecture of FCNNs, including layers, nodes, activation functions, weights, and biases. Explain the forward pass.
    - **2.3.3 Training Neural Networks:** Explain the concept of loss functions (e.g., mean squared error), optimization algorithms (e.g., Adam), backpropagation, and gradient descent.
    - **2.3.4 Hyperparameter Optimization:** Introduce the concept of hyperparameters and the need for their optimization. Briefly mention Optuna.
    - **2.3.5 Emulation with Neural Networks:** Explain the concept of using neural networks as emulators to approximate complex functions or simulations.

III. Methodology (approx. 1000 words)

- **3.1 Dataset Generation:**
    - **3.1.1 CAMB Configuration:** Describe the specific configuration of CAMB used to generate the dataset. List the cosmological parameters and their ranges. Explain the choice of these ranges.
    - **3.1.2 Sampling Strategy:** Explain the method used to sample the parameter space (e.g., Latin Hypercube Sampling). Justify the choice of sampling strategy.

- **3.1.3 Data Preprocessing:** Describe any preprocessing steps applied to the CAMB output, such as:
  - Normalization: Explain how the power spectra were normalized.
  - Logarithmic Scaling: Explain the use of logarithmic scaling for both k and P(k).
  - Data Splitting: Describe how the dataset was split into training, validation, and testing sets. Specify the proportions used.
- **3.1.4 Data Storage:** Describe how and where the data is stored.

- **3.2 Neural Network Architecture:**
  - **3.2.1 FCNN Design:** Provide a detailed description of the chosen FCNN architecture:
    - Input Layer: Specify the number of input nodes (corresponding to the number of cosmological parameters).
    - Hidden Layers: Specify the number of hidden layers (4) and the number of nodes in each layer (1024). Justify this choice.
    - Output Layer: Specify the number of output nodes (corresponding to the number of k-values at which P(k) is evaluated).
    - Activation Functions: Specify the activation functions used in each layer (e.g., ReLU, sigmoid, tanh). Justify these choices.
    - [Placeholder for a diagram of the network architecture]
- **3.3 Hyperparameter Tuning:**
  - **3.3.1 Optuna:** Introduce Optuna as the hyperparameter optimization framework.
  - **3.3.2 Search Space:** Define the search space for the hyperparameters, including:
    - Learning Rate: Specify the range and type of distribution (e.g., logarithmic).
    - Batch Size: Specify the range.
    - Optimizer: Specify the optimizers considered (e.g., Adam, SGD).
    - (Include any other hyperparameters that were tuned)
  - **3.3.3 Optimization Process:** Describe the optimization process, including the number of trials, the objective function (e.g., validation loss), and the pruning strategy.

IV. Implementation Details (approx. 750 words)

- **4.1 Jax Framework:**
  - **4.1.1 Introduction to Jax:** Briefly introduce Jax and its advantages for numerical computation and machine learning.
  - **4.1.2 Data Loading and Preprocessing in Jax:** Describe how the dataset is loaded and preprocessed using Jax. [Placeholder for pseudocode/code snippets]
  - **4.1.3 FCNN Implementation in Jax:** Describe the implementation of the FCNN architecture in Jax. [Placeholder for pseudocode/code snippets showing the network definition]
  - **4.1.4 Loss Function and Optimizer in Jax:** Describe the implementation of the loss function and optimizer in Jax. [Placeholder for pseudocode/code snippets]
  - **4.1.5 Training Loop in Jax:** Describe the training loop, including forward pass, loss calculation, backpropagation, and parameter updates. [Placeholder for pseudocode/code

snippets]

- **4.2 Computational Resources:**
    - **4.2.1 Cambridge HPC:** Describe the Cambridge HPC resources used, including GPU and CPU allocations.
    - **4.2.2 Software Environment:** Specify the software environment, including Jax version, Python version, and other relevant libraries.
- **4.3 Data Interpolation:**
    - **4.3.1 Need for Interpolation:** Explain why interpolation is needed (to evaluate P(k) at a fixed set of k-values for all parameter combinations).
    - **4.3.2 Interpolation Method:** Describe the chosen interpolation method (e.g., linear interpolation, cubic spline interpolation). Justify the choice.
    - **4.3.3 Implementation in Jax:** Describe the implementation of the interpolation method in Jax. [Placeholder for pseudocode/code snippets]

V. Experiments and Results (approx. 1000 words)

- **5.1 Training Process:**
    - **5.1.1 Training and Validation Curves:** [Placeholder for plots of training and validation loss vs. epoch]. Analyze these curves, looking for signs of overfitting or underfitting.
    - **5.1.2 Hyperparameter Optimization Results:** [Placeholder for a table summarizing the hyperparameter optimization results]. Report the best hyperparameters found.
    - **5.1.3 Training Time:** Report the total training time.
- **5.2 Emulator Performance:**
    - **5.2.1 Evaluation Metrics:** Define the metrics used to evaluate the emulator's performance, such as:
        - Mean Squared Error (MSE)
        - Mean Absolute Percentage Error (MAPE)
        - R-squared (coefficient of determination)
        - Maximum Error
    - **5.2.2 Quantitative Results:** [Placeholder for a table summarizing the performance metrics on the test set]. Present the results for each metric.
    - **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]. Visually assess the accuracy of the emulator.
    - **5.2.4 Error Distribution:** [Placeholder for a histogram of the errors (e.g., percentage errors)]. Analyze the distribution of errors.
    - **5.2.5 Performance vs. Cosmological Parameters:** [Placeholder for plots showing the error as a function of different cosmological parameters]. Investigate whether the emulator's performance is dependent on specific parameter values.

VI. Discussion (approx. 500 words)

- **6.1 Interpretation of Results:**
  - Discuss the overall performance of the emulator.
  - Interpret the results in the context of the theoretical background.
  - Compare the achieved accuracy with the requirements and expectations.
- **6.2 Challenges and Limitations:**
  - Discuss any challenges encountered during the project.
  - Identify the limitations of the current emulator (e.g., range of validity, specific parameter combinations where it performs poorly).
- **6.3 Comparison with Theoretical Expectations:**
  - Compare the emulator's predictions with theoretical expectations from linear perturbation theory.
  - Discuss any discrepancies and potential explanations.
- **6.4 Comparison with other emulators/methods:**
  - Compare the performance with other existing emulators or methods for calculating the matter power spectrum.

VII. Conclusion (approx. 250 words)

- **7.1 Summary of Work:**
  - Provide a concise summary of the project, reiterating the main findings.
- **7.2 Future Directions:**
  - Suggest potential future research directions, such as:
    - Exploring different network architectures (e.g., convolutional neural networks, recurrent neural networks).
    - Incorporating more cosmological parameters.
    - Extending the emulator to higher redshifts.
    - Developing emulators for other cosmological observables.
    - Using the emulator for parameter inference.
    - Investigating methods for uncertainty quantification.
- **7.3 Final Remarks:**
  - Offer concluding remarks on the significance of the work and its potential impact on cosmological research.

This outline provides a comprehensive framework for the 5000-word report. Each section and subsection can be expanded upon with detailed explanations, derivations, and results. The placeholders indicate where specific figures, tables, and code snippets should be inserted.