# Detailed Report of the Current Model: EEG Classification Using DCGRU with Dual Graph Inputs

## 1. Introduction

This report describes the current model for classifying EEG signals into two categories: Depressed and Healthy. The model leverages **Diffusion Convolutional GRU (DCGRU)**, a powerful graph-based deep learning model designed to capture both **spatial** and **temporal dependencies** in EEG signals. The model integrates **dual graph inputs** — a **distance graph** to represent spatial relationships between EEG electrodes and a **correlation graph** to represent functional connectivity between electrodes. The DCGRU architecture is utilized for **self-supervised pre-training** and fine-tuned for the **binary classification** task.

DCGRU combines diffusion convolution for modeling spatial relationships and GRU (Gated Recurrent Units) to capture the temporal dynamics in sequential data, making it ideal for analyzing the dynamic nature of EEG signals.

## 2. Data Collection and Preprocessing

**Objective:** Prepare EEG data in a suitable form for graph-based processing.

- **Data Acquisition:**
    - The model uses **labeled EEG datasets** such as DEAP, SEED, or clinical EEG recordings, which are annotated with two classes: **Depressed** and **Healthy**.
    - The datasets typically involve **64-channel** EEG setups using the **10-20 system** for electrode placement.
- **Signal Preprocessing:**
    - **Artifact Removal:** Techniques like **Independent Component Analysis (ICA)** or **Wavelet Decomposition** are used to remove noise from EEG signals (e.g., eye blinks, muscle activity).
    - **Frequency Filtering:** A **band-pass filter** (0.5–45 Hz) is applied to retain the relevant frequency bands (alpha, beta, theta) for brainwave analysis.
    - **Epoch Segmentation:** The continuous EEG signal is divided into smaller time segments, typically **2-5 seconds** long, to create input samples for the model.
- **Feature Extraction:**
    - **Power Spectral Density (PSD)** and **band power** (alpha, beta, theta) are computed as features.
    - Features are **normalized** across electrodes and samples to ensure consistency across the dataset.

### 3. Graph Construction

**Objective:** Construct graphs that represent the relationships between EEG electrodes.

- **Nodes:** Each EEG electrode is treated as a **node** in the graph.
- **Edges:**
    - **Spatial Graph (Distance Graph):** Edges are created using the **Euclidean distance** between electrode positions, typically derived from the **10-20 EEG system**.
    - **Functional Graph (Correlation Graph):** Edges are formed based on the **normalized cross-correlation** between preprocessed EEG signals at different electrodes, capturing dynamic connectivity.

For both graphs, the adjacency matrices are constructed:

- **Distance-based adjacency matrix:** Represents spatial adjacency based on electrode positions.
- **Correlation-based adjacency matrix:** Represents functional adjacency based on signal similarity.

### 4. Graph-Based Representation

**Objective:** Represent EEG data as a graph for input to the DCGRU model.

- **Node Features:**
    - Node features are computed for each electrode using the EEG features (e.g., band power, spectral entropy, etc.), representing the characteristics of the signals recorded at each electrode.
- **Adjacency Matrices:**
    - **Spatial Graph Adjacency Matrix (A_distance):** A weighted matrix capturing spatial relationships between electrodes.
    - **Functional Graph Adjacency Matrix (A_correlation):** A weighted matrix capturing functional relationships based on cross-correlation.
- **Dynamic Graphs (Optional):**
    - Separate graphs can be created for each **epoch** (time segment) to capture the temporal evolution of connectivity across EEG electrodes.

### 5. Self-Supervised Pre-Training

**Objective:** Learn meaningful spatiotemporal representations of EEG data in an unsupervised manner.

- **Input:**
    - Node features (EEG features per electrode) and adjacency matrices representing spatial and functional connectivity.
- **Model Architecture:**
    - **DCGRU Encoder:**
        - The encoder captures the spatiotemporal dependencies in EEG signals by utilizing **diffusion convolution** to model spatial relationships and **GRU layers** to capture temporal dynamics.

        - Two separate graphs (distance and correlation) are fed into the encoder using different branches, enabling the model to process spatial and functional connectivity simultaneously.
    - **Task-Specific Decoder:**
        - **Masked Signal Prediction:** Predicts missing EEG signal features from specific electrodes.
        - **Next-Step Prediction:** Predicts the next time segment of EEG signals to capture temporal dependencies.
        - **Contrastive Learning:** Maximizes the agreement between augmented EEG signals and their original representations.
- **Loss Functions:**
    - **Mean Squared Error (MSE):** Used for reconstruction tasks.
    - **Contrastive Loss:** Used for representation learning tasks to ensure meaningful embeddings.
- **Output:**
    - The output is a set of **latent graph embeddings** that capture the underlying spatiotemporal patterns in the EEG data.

### 6. Binary Classification Task

**Objective:** Fine-tune the pre-trained DCGRU model for classifying EEG signals into Depressed or Healthy classes.

- **Input:**
    - The **graph embeddings** from the self-supervised pre-training phase, which contain rich spatiotemporal representations.
- **Model Architecture:**
    - **DCGRU Layers:** The pre-trained DCGRU layers are used to extract features from the graph representations.
    - **Graph Pooling:** Node-level features are aggregated into a **graph-level** representation using methods like **global mean pooling** or **attention pooling**.
    - **Classification Head:**
        - The aggregated graph features are passed through fully connected layers to produce a **binary output** (Depressed or Healthy).
        - A **sigmoid activation** function is applied to get the final probability for classification.
- **Loss Function:**
    - **Binary Cross-Entropy Loss:** Used to optimize the model for binary classification.
- **Output:**
    - A binary label (1 for Depressed, 0 for Healthy), representing the classification result.

### 7. Model Interpretability

**Objective:** Understand which features, nodes, or connections are most relevant for classification.

- **Attention Mechanisms:**
    - **Graph Attention Networks (GATs)** can be used to highlight important electrodes or connections in the graph that contribute significantly to the classification task.
- **Explainability Tools:**
    - **Grad-CAM** or **saliency maps** can be used to visualize which parts of the graph (electrodes, connections, or features) are most important for the classification decision.
- **Insights:**
    - The model can reveal which regions of the brain or frequency bands are most relevant for distinguishing between Depressed and Healthy classes, providing valuable insights for clinical decision-making.

### 8. Validation and Testing

**Objective:** Evaluate the performance of the model on unseen data.

- **Evaluation Metrics:**
    - **Accuracy, Precision, Recall, F1-score,** and **ROC-AUC** to evaluate the model's classification performance.
    - **Sensitivity (True Positive Rate)** and **Specificity (True Negative Rate)** are used to assess the model’s ability to identify both classes.
- **Validation Techniques:**
    - **K-fold cross-validation** (e.g., 5-fold, 10-fold) is used to assess the robustness and generalizability of the model.
    - Testing is done on **independent datasets** to ensure the model performs well on unseen data.

### 9. Deployment

**Objective:** Integrate the trained model into a practical diagnostic pipeline.

- **Interface:**
    - Develop a user-friendly interface where EEG data can be uploaded, and predictions can be made (Depressed or Healthy).
- **Integration:**
    - The trained model is packaged for clinical or research use, providing not only the classification result but also **confidence scores** and visual outputs like **saliency maps** or **Grad-CAM visualizations** to explain the predictions.

### 10. Pipeline Summary

1. **Data Collection and Preprocessing:** Prepare and preprocess EEG signals and construct spatial and functional graphs.
2. **Self-Supervised Pre-Training:** Use the DCGRU model to learn spatiotemporal representations through masked signal prediction, next-step prediction, and contrastive learning.
3. **Binary Classification:** Fine-tune the pre-trained DCGRU model for the binary classification task (Depressed vs. Healthy).
4. **Interpretability:** Use attention mechanisms and explainability tools to analyze the important features for classification.
5. **Validation and Deployment:** Evaluate the model using appropriate metrics, validate its generalization, and deploy it for practical use.

## Conclusion

The current model utilizes **dual graph inputs** (distance and correlation graphs) to capture both the **spatial** and **functional** relationships between EEG electrodes. By leveraging **DCGRU** and incorporating **self-supervised pre-training**, the model effectively learns rich representations of EEG data, which are then fine-tuned for **binary classification**. The incorporation of **model interpretability techniques** ensures that the model's decision-making process is transparent, which is critical for real-world applications in **clinical settings**. Future steps could include expanding the model for other neurological disorders or testing it with larger datasets for better generalizability.