**AUC (Area Under the Curve)**

A number that measures how well a model can tell the difference between two classes (like "disease" vs. "no disease"). The closer the AUC is to 1, the better the model is at making predictions.

**Bipartite Graph**

A graph with two separate groups of nodes, where connections (edges) can only happen between nodes in different groups (e.g., diseases and genes).

**Decoder**

Part of a model that takes simplified information (like embeddings) and predicts or reconstructs the original data, such as edges in a graph.

**Disease-Gene Association (Host Response Aspect)**

A relationship between a disease and genes involved in how the body reacts to that disease, often studied to understand underlying biological processes.

**Edge List**

A way to represent a graph by listing all the connections (edges) between nodes. For example, if nodes A and B are connected, the edge list will include (A, B).

**Embeddings**

A small set of numbers that represent a node or item in a simplified way, keeping important information about its features and relationships.

**Encoder**

Part of a model that takes data (like a graph) and simplifies it into embeddings, capturing the most important information.

**Features**

Information or properties about nodes in a graph. For example, in a gene-disease graph, features for a gene might include its expression level or role in a biological process.

**Gene Propagation**

A method for spreading information about genes across a graph to predict or identify new relationships, like genes associated with a disease.

**Graph**

A structure made of **nodes** (e.g., diseases, genes) and **edges** (connections between nodes), used to represent relationships.

**Graph Autoencoder (GAE)**

A special type of model that learns simplified representations (embeddings) of nodes in a graph and uses them to predict or reconstruct edges (connections).

**Hyperparameter**

A setting that you choose before training a model (e.g., learning rate or number of layers) to control how the model learns.

**Latent Space**

A "hidden space" where data is represented in a simplified and compressed form, often created by an encoder.

**Ontology**

A structured framework for organizing information, often used in biology to classify genes, diseases, or proteins and their relationships.

**Overfitting**

When a model learns too much detail from the training data, including noise, and performs poorly on new, unseen data.

**Parameters**

Values in a model (like weights in a neural network) that are learned during training to make better predictions.

**Parameter vs. Hyperparameter**

* **Parameters**: Learned by the model during training (e.g., weights).
* **Hyperparameters**: Set manually before training (e.g., learning rate).

**Performance Metrics**

Numbers or scores that measure how well a model is working, like accuracy, AUC, or precision.

**PPI Network (Protein-Protein Interaction Network)**

A map showing how proteins interact with each other, often represented as a graph to study relationships and biological functions.

**Precision**

The proportion of correct positive predictions out of all the positive predictions made by the model. A measure of how accurate the model’s "yes" predictions are.

**PyTorch Data Object**

A way to organize graph data (like nodes, edges, and features) in PyTorch for use in machine learning models.

**ROC Curve (Receiver Operating Characteristic Curve)**

A graph showing how well a model can distinguish between two groups (e.g., disease vs. no disease) at different thresholds. It plots true positives vs. false positives.

**Train, Test, and Validation Split**

Breaking data into three parts:

* **Training**: Data the model learns from.
* **Validation**: Data to check how well the model is learning.
* **Test**: Data to evaluate the model’s final performance on unseen information.