GraphDTA Baseline Project Report

# Objective

To reproduce and extend the baseline GraphDTA model, which predicts the binding affinity between drug molecules and protein targets using:  
- A Graph Convolutional Network (GCN) to model drugs from their SMILES structure  
- A 1D Convolutional Neural Network (CNN) to model protein sequences  
- A multi-layer perceptron (MLP) to predict the affinity score  
This project serves as a foundation for an enhanced GraphDTA-3D model that will incorporate AlphaFold-predicted protein structures.

# Dataset Used

Source: Davis dataset

Files involved:  
- drugs.csv: Contains Drug\_Index and Canonical\_SMILES  
- proteins.csv: Contains Protein\_Index and Sequence  
- drug\_protein\_affinity.csv: Contains triplets (Drug\_Index, Protein\_Index, Affinity)

Target variable: Affinity value (continuous), typically log-transformed Kd.

# Drug Preprocessing – SMILES to Graph

Method:  
- SMILES strings are parsed using RDKit into molecule objects.  
- Molecules are converted into graphs (nodes = atoms, edges = bonds).  
- Node features = one-hot encoded atom types.

Reasoning: GNNs can leverage the molecular topology to learn chemical interactions.  
Output saved as davis\_drug\_graphs.pt.

# Protein Preprocessing – Sequence to Index Tensors

Method:  
- Sequences mapped into amino acid indices using a 21-character vocabulary.  
- Stored as variable-length index lists and saved as protein\_index\_sequences.pkl.

Reasoning: CNNs extract sequential motifs. Padding applied at batch-time.

# Dataset Construction

Method:  
- DTADataset retrieves preprocessed drug graphs and protein sequences.  
- Returns (drug\_graph, protein\_tensor, affinity).

Reasoning: Modular dataset allows flexibility in model input and extensions.

# DataLoader with Custom collate\_fn

Method:  
- Drug graphs batched with Batch.from\_data\_list()  
- Protein sequences padded with pad\_sequence()

Reasoning: Supports variable-length sequences and graph batching efficiently.

# Model Architecture – GraphDTA Baseline

Model:  
- Drug: 3-layer GCN  
- Protein: Embedding + 2x Conv1D + Pooling  
- Fusion: Concatenate + MLP

Reasoning: GCNs for structure, CNNs for sequential motifs, fusion for interaction.

# Training Loop and Evaluation

Training:  
- Optimizer: Adam, Loss: MSELoss  
- Epochs: 30–50

Metrics:  
- RMSE: prediction accuracy  
- CI: ranking consistency

Implemented fully in PyTorch (no NumPy).

# Regularization and Stabilization

Method:  
- Added Dropout(p=0.3) after dense layers.

Result:  
- Reduced overfitting  
- CI improved from ~0.54 → 0.57  
- RMSE stabilized ~0.9–1.3

# Training Performance Summary

Train Loss: ~0.002  
Validation RMSE: ~0.9–1.2  
CI: ~0.57

# Conclusion

The GraphDTA baseline pipeline is:  
- Modular  
- Documented  
- Reproducible  
- Accurate enough to serve as a launchpad for GraphDTA-3D.