# AETHERIX - Technical Overview : CGNVX

## Overview

The RL-Based Drug Discovery Model is an advanced system designed to recommend personalized drugs based on patient-specific blood reports. It leverages Reinforcement Learning (RL) to simulate multi-step treatment processes, integrating deep learning for feature extraction from text-based blood reports. The model dynamically adjusts patient states (e.g., hemoglobin, glucose) over time, optimizing drug recommendations to achieve ideal health parameters while considering side effects and patient-specific factors like age and weight.

## How It Works

### 1. Data Input and Feature Extraction

- Input: Patient blood reports containing Hemoglobin (g/dL), Glucose (mg/dL), Age, and Weight (kg) are processed.  
- Feature Extraction:   
 - BERT (`bert-base-uncased`) transforms text into a 768-dimensional embedding capturing semantic content.  
 - Additional features (Hemoglobin, Glucose, Age, Weight) are extracted via regex, adding 4 dimensions.  
 - Total state vector: 772D (768 from BERT + 4 scalar values), cast to `float32`.

### 2. Multi-Step RL Environment

- Environment: `PatientDrugEnv` simulates a multi-step treatment process (up to 5 steps).  
- State: 772D vector representing patient condition.  
- Actions: Selection from 20 possible drugs, each with predefined effects on hemoglobin, glucose, and side effect severity.  
- Dynamics:   
 - Drug effects adjust hemoglobin and glucose, scaled by patient weight (e.g., effect \* weight/70).  
 - Side effects scale with age (e.g., severity \* age/50).  
- Reward Function:   
 - Negative error from ideal ranges (Hemoglobin: 12-16 g/dL, Glucose: 70-110 mg/dL).  
 - Penalty for side effects.  
 - Bonus (+1.0) if error < 0.5, encouraging near-ideal states.  
- Termination: Ends when reward > 0 (ideal state) or after 5 steps.

### 3. Policy Learning

- Algorithm: Proximal Policy Optimization (PPO) learns an optimal drug recommendation policy.  
- Custom Policy Network:   
 - `CustomFeatureExtractor` combines BERT embeddings and patient data using a Transformer encoder.  
 - Output: 256D feature vector fed into PPO’s actor-critic network.  
- Training:   
 - Collects 2048 steps per rollout, updates over 10 epochs.  
 - Total 100,000 timesteps, optimizing on GPU.  
- Prediction: Selects the drug with the highest probability, simulating treatment steps.

### 4. Output Generation

- Recommendation: Final drug selected after multi-step simulation.  
- Details: SMILES string mapped to drug name and molecular formula via RDKit and PubChemPy.  
- Confidence: Computed from policy logits using softmax.  
- Visualization: Plots hemoglobin and glucose trajectories over steps, showing progress toward ideal ranges.

## Frameworks Used

- gymnasium: Provides the RL environment framework for defining states, actions, and rewards.  
- stable-baselines3: Implements PPO, enabling efficient RL training and policy management.  
- transformers: Supplies BERT (`bert-base-uncased`) for text feature extraction from blood reports.  
- torchvision: Provides ResNet-18 for potential image processing (simulated here).  
- torch: Powers GPU-accelerated tensor operations for all neural networks.  
- rdkit: Parses SMILES strings to derive molecular formulas.  
- pubchempy: Maps SMILES to drug names via PubChem database queries.  
- tdc: Sources the initial drug dataset (ADME `Caco2\_Wang`) with 20 SMILES entries.  
- matplotlib: Generates patient state trajectory plots for visualization.