**Plan & Research: Reinforcement Learning for Crypto Trading with Generative Simulation**

## **1. Tech Stack & Tools**

### **Core Stack:**

* **Language**: Python (ML/data science ecosystem).

### **ML Frameworks:**

* **PyTorch/TensorFlow**: For building TimeGAN and RL models.
* **Stable Baselines3 (RL)**: Pre-implemented algorithms (PPO, DQN) with GPU support.
* **Gymnasium**: Custom trading environment (OpenAI Gym fork).

### **Generative Models:**

* **TimeGAN**: For synthetic time-series data – Reference Implementation.

### **Data Tools:**

* **Pandas/NumPy**: Data preprocessing.
* **Binance API/CCXT**: Historical price/trade data.
* **Glassnode API**: On-chain metrics (mining hash rate, wallet activity).

### **Visualization:**

* **Plotly, Matplotlib**.

### **Deployment:**

* **Streamlit**: Dashboard.
* **Docker**: Containerization.

### **Optional/Niche Tools:**

* **Monte Carlo Tree Search (MCTS)**: Use alpha-zero-general GitHub repo for chess-inspired planning.
* **Sentiment Analysis**: Hugging Face Transformers (e.g., FinBERT for crypto news).

## **2. Existing Solutions & Best Practices**

### **A. Crypto Trading with RL**

#### **Prior Work:**

* **Deep Reinforcement Learning for Cryptocurrency Trading**: Uses PPO with technical indicators.
* **Practical Limitations of Deep Q-Learning in Trading**: Highlights overfitting risks.

#### **Best Practices:**

* **Reward Shaping**: Use Sharpe Ratio or Sortino Ratio instead of raw returns.
* **Feature Engineering**: Include volatility, RSI, mining hash rate, and on-chain metrics.

### **B. Generative Market Data**

* **TimeGAN**: Proven for stock data synthesis; adapt for crypto’s 24/7 volatility.

#### **Validation Metrics:**

* **Dynamic Time Warping (DTW)**: Compare synthetic/historical sequences.
* **Fréchet Inception Distance (FID)**: Measure distribution similarity.

### **C. Mining Data Integration**

#### **Key Metrics:**

* Hash rate, miner revenue, miner outflow (Glassnode’s “Miner Position Index”).

#### **Studies:**

* Miner Activity & Bitcoin Price Correlation.

## **3. Project Roadmap**

### **Timeline: 12 Weeks**

| Week | Milestone | Key Task |
| --- | --- | --- |
| 1-2 | Data Pipeline & EDA | Collect data from Binance perform ETL & Publish EDA report |
| 3-4 | TimeGAN Implementation | Train TimeGAN on BTC/ETH hourly data Validate with DTW/FID scores Augment training dataset with synthetic data |
| 5-6 | RL Environment & Agent | Build Gymnasium environment (state=price, volume, mining data) Train PPO agent on synthetic data |
| 7-8 | Mining Module | Add mining hash rate as a feature  Test agent's response to simulated miner sell pressure |
| 9-10 | Backtesting & optimization | Benchmark against buy-and-hold, SMA Tune hyperparameters (e.g., discount factor, risk penalty) |
| 11-12 | Deployment & Final Report | Build Streamlit dashboard, Write thesis + GitHub repo documentation |

### **4. Research Insights**

### **A. Risks & Mitigations**

* **Data Quality**: Binance API rate limits → Use CCXT with multiple exchanges.
* **TimeGAN Training Instability**: Start with small synthetic sequences (e.g., 24-hour windows).
* **RL Overfitting**: Train on synthetic data, test on unseen historical data.

### **B. Innovation Opportunities**

* **Chess Parallel**: Use MCTS to simulate “what-if” trading paths (like chess move trees).
* **Mining Predictions**: Model miner behavior as a game (e.g., miners vs. traders).

### **C. Key Papers**

* **TimeGAN**: Time Series Synthesis with GANs.
* **RL for Portfolio Management**.
* **Miner-Driven Bitcoin Price Dynamics**.