DeFi Liquidity Provisioning with Reinforcement Learning and Dynamic Feature Selection

Benoît-Marie ROBAGLIA
Independent Researcher
Paris, France
benoitrobaglia@gmail.com

***Abstract*—This paper introduces a Deep Reinforcement Learning Algorithm for liquidity provision on Uniswap V3. This method enhances Proximal Policy Optimization with 1) Cross-Validated Recursive Feature Elimination to build a volatility-aware state representation and 2) Bayesian Online Change-Point Detection for real-time regime shifts. Experiments on the FinAI Contest 2025 Task 3 benchmark show that our proposed solution outperforms both the vanilla PPO baseline and a passive full-range strategy.**

***Index Terms*—Decentralized Finance (DeFi), Liquidity Provision, Uniswap v3, Reinforcement Learning**

I. Introduction

Decentralized Finance (DeFi) has rapidly emerged as a pillar of financial innovation, enabling permissionless, transparent, and non-custodial trading with Automated Market Makers (AMMs) like Uniswap v3 [1]. However, providing liquidity to these protocols remains complex, particularly for individual retail users as they face challenges such as impermanent loss, volatility, and the presence of adverse arbitrageurs [8].

Recent work by Xu et Brini [9] has explored applying deep reinforcement learning (RL) to optimize the placement and rebalancing of liquidity in concentrated AMMs. By modeling the LP's decision process as a sequential optimization problem, the authors have shown that the Proximal Policy Optimization (PPO) algorithm could help in mitigating losses and increase collected trading fees.

In this paper, we extend the liquidity provision framework of [9], addressing key implementation issues (see V) and introducing several practical enhancements. Specifically, we adopt a more flexible neural network architecture, apply Cross-Validated Recursive Feature Elimination (CVRFE) with a random forest regressor to improve volatility prediction, and integrate Bayesian online change point detection (BOCPD) to help the RL agent adapt to market regime shifts in real time. Our approach consistently outperforms the PPO baseline proposed by [9] in terms of cumulative rewards, with performance evaluated on the FinAI contest benchmark.

Related Work

A major research direction in DeFi explores how liquidity providers (LPs) can best balance risk and reward in Uniswap-style AMMs. Early methods focused on optimizing liquidity allocations to maximize fees while minimizing impermanent loss (IL) and transaction costs. For example, the authors in [3] uses neural networks to adjust liquidity intervals based on price dynamics and costs. Some approaches treat LP positions as option-like exposures, leveraging off-chain hedging with futures and deep RL [10]. In contrast, following [9], our work focuses on on-chain liquidity management, making strategies accessible to retail users without advanced off-chain execution.

Feature selection is essential in trading strategies to address the stochastic, non-stationary nature of financial returns. Recursive Feature Elimination (RFE) [4] is widely used for high-dimensional financial data and has been shown to improve forecasting across asset classes [6]. We adopt a cross-validated RFE step to retain only the most relevant volatility predictors, producing a compact, regime-adaptive state for the RL agent.

Finally, detecting market regime shifts is key for resilient trading, as financial time series statistics can change abruptly. Bayesian Online Change Point Detection (BOCPD) [1] offers a real-time, probabilistic method to track such changes. When included as a feature for reinforcement learning, BOCPD signals improve the agent's responsiveness to evolving markets, with proven benefits for adaptive financial strategies [2].

II. Problem Formulation: Liquidity Provisioning on Uniswap V3

We consider the task of optimal liquidity management for a liquidity provider (LP) in a concentrated automated market maker (AMM) like Uniswap v3. The LP dynamically decides how to allocate capital over dynamic price intervals in order to maximize cumulative trading fees while mitigating risks such as loss vs rebalancing (LVR) and adverse price movements.

Following [9], the LP's sequential decision process is modeled as a Markov Decision Process (MDP) $(\mathcal{S}, \mathcal{A}, P, r, \gamma)$, where:

- $\mathcal{S}$: The state at time t combines current and historical market data. Our state vector includes position features (price, tick, width, liquidity, volatility σ, liquidity ratio), market features (moving averages, Bollinger Band width, ADXR, momentum, aroon oscillator, and change point probability), and volatility-related predictors chosen via CVRFE.
- $\mathcal{A}$: The action is the choice of liquidity width (in ticks, centered at tick i_t) as in [9].
- $P(s_{t+1}|s_t, a_t)$: Transition dynamics follow market evolution and pool mechanics.

[1] https://uniswap.org/whitepaper-v3.pdf

- $r(s_t, a_t)$: Reward is calculated as

$$r(s_t, a_t) = \text{fees}_t - \text{LVR}_t - \text{gas}_t \tag{1}$$

with LVR estimated by [5], [9]:

$$\text{LVR}_t = L_t \frac{\sigma_t^2}{4} \sqrt{p_t} \tag{2}$$

and fees as in [3], [9].
- $\gamma \in (0, 1)$: Discount factor for future rewards.

The goal is to learn a policy $\pi(a_t|s_t)$ that maximizes the expected sum of future discounted rewards, selecting position ranges to optimize total fee collection and minimize LVR.

III. Proximal Policy Optimization with Dynamic Feature Selection

In this section, we introduce our main contributions. We build them on top of the PPO framework [7], [9].

A. Cross-Validated Recursive Feature Elimination (CVRFE)

Our first main contribution is the application of CVRFE to volatility forecasting, which is essential to risk adjustment and LVR estimation. CVRFE builds upon the well-established principle of recursive feature elimination [4], where features contributing least to predictive performance are successively removed, and improves robustness by using K-fold cross-validation to assess the predictive power of each candidate feature subset at every step.

We also experimented with using RFE to predict future token prices with the hope of directly optimizing expected fee collection. However, due to the noise in the price data and the limited predictive signal in available features, results were poor. Additionally, as our environment only allows symmetric liquidity ranges centered at the current mid-price (precluding directional bets), we decided not to continue price prediction-based strategies further.

B. Bayesian Online Change Point Detection (BOCPD)

Our second main contribution is the integration of Bayesian Online Change Point Detection (BOCPD) [1] to provide real-time regime awareness. Specifically, BOCPD maintains a probability distribution over the "run length" (time since last change point). We include the resulting change point probability as a feature in the RL agent's state, enabling the policy to adapt its behavior when market regimes are likely to shift. Our implementation follows the standard recursive strategy for efficiency and robustness, as detailed in [1].

C. Methodology

We employ a walk-forward, chunked training and testing procedure to robustly evaluate the LP strategy, following [9]. The full methodology is described in Algorithm 1.

Key implementation details: - *Volatility:* For each window, CVRFE selects optimal volatility features on training data, and the predicted $\widehat{\sigma}$ is added to the feature set. - *Regime:* BOCPD computes change point probabilities (`cp_prob`), which are used as regime-awareness features. - *Model tuning:* RL hyperparameters are optimized in two Optuna[2] stages per fold, preventing lookahead bias.

Algorithm 1 Walk-Forward Training, Feature Selection, and RL Optimization

Require: Full dataset D (chronologically ordered), chunk size $N_c = 1500$. Features are computed at initialization.

Partition D into sequential, non-overlapping chunks: $D_1, D_2, \ldots, D_K$, each of length N_c

for window $i = 1$ to $K - 5$ **do**

  Step 1: Data Splitting

  Define training data: $D_{\text{train}} = D_i \cup D_{i+1} \cup \ldots \cup D_{i+4}$

  Define testing data: $D_{\text{test}} = D_{i+5}$

  Step 2: Volatility Feature Selection (CVRFE)

  Let $\mathcal{F}_0$ be the set of all features in D_{train}

  Use CVRFE with a random forest regressor to select optimal volatility predictors:

    $\mathcal{F}_{\text{vol}} = \text{CVRFE}(D_{\text{train}}, \mathcal{F}_0, \text{future volatility target})$

  Fit f_{vol} on $D_{\text{train}}[\mathcal{F}_{\text{vol}}]$; predict $\widehat{\sigma}$ for both train and test sets

  Add $\widehat{\sigma}$ and selected features to $D_{\text{train}}, D_{\text{test}}$

  Step 3: Feature Set Construction

  Combine base market features, $\mathcal{F}_{\text{vol}}$, predicted volatility ($\widehat{\sigma}$), and other position state feature into a single feature vector for RL.

  Step 4: Reinforcement Learning – PPO Optimization

  Use Optuna to perform two-step hyperparameter optimization, following [9]

  Step 5: Backtest Evaluation

  Evaluate the best PPO policy on D_{test}; record performance for this window

end for

return Out-of-sample performance metrics over all folds

IV. Empirical Studies

A. Experimental Setup

Our experiments use hourly WETH/USDC data from Uniswap v3 (0.05% fee tier), covering May 5, 2021 (v3 launch) to January 29, 2024. Following the protocol of Xu et al. [9], we apply walk-forward backtesting with non-overlapping train/test splits: each fold includes 7,500 training hours (10 months) and 1,500 testing hours (2 months), shifting the data window by 1,500 hours per iteration. RL hyperparameters are re-optimized before each fold using Bayesian Optimization.

We compare our approach to the baselines from [9]:

- **Passive Provision:** a periodic strategy that modifies LP positions at fixed 500-hour intervals (approx 20 days).
- **Original PPO:** Proximal Policy Optimization using only static market features, as implemented in [9].

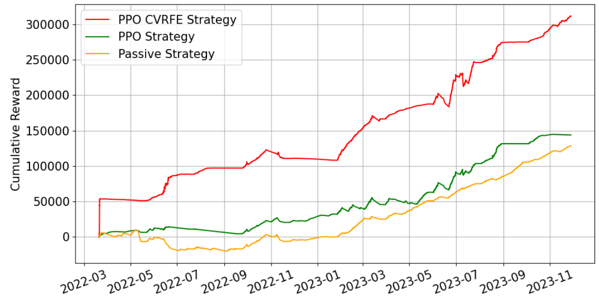

Fig. 1. Cumulative rewards during the out-of-sample periods corresponding to the testing windows.

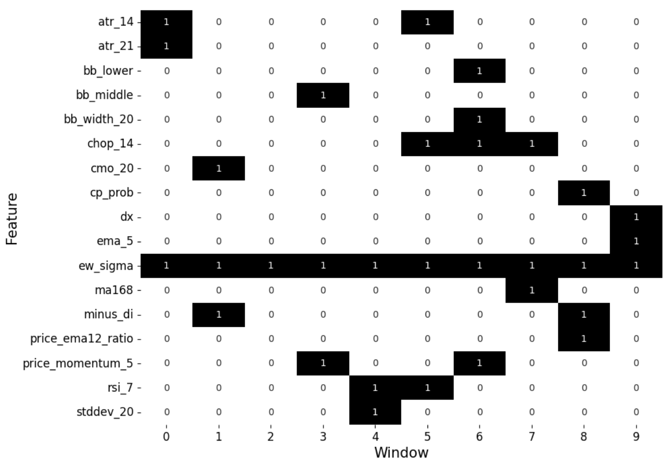

Fig. 2. Feature Selection Matrix Across all Windows, with feature importance $> 1e-3$.

B. Simulation Results

Figure 1 presents the cumulative reward across all out-of-sample windows for all strategies. It shows that our RL-based strategy with CVRFE (PPO-CVRFE) significantly outperforms both baselines across all test windows. This improved performance can be explained by three upgrades: (1) regime-adaptive volatility prediction via CVRFE, (2) a larger neural network architecture (up to 3 layers and 256 neurons), and (3) more extensive Bayesian hyperparameter optimization (20 trials versus 5).

Besides, the feature selection matrix (Fig. 2) shows that prior volatility (`ew_sigma`) is the only feature consistently selected. This is aligned with financial research saying that recent volatility is the strongest single predictor of future volatility. Other frequently selected features are also volatility or regime-related (e.g., Choppiness Index, ATR, Bollinger Band metrics), while technical indicators like RSI, momentum, or change point probability are only sporadically selected. This pattern highlights the robustness and interpretability of our CVRFE-based methodology, dynamically focusing on regime-relevant predictors.

[2]https://optuna.org/

V. Conclusion

This work extends the FinAI PPO baseline for Uniswap v3 liquidity provision with three key enhancements: a more expressive neural network, adaptive feature selection via cross-validated RFE for volatility prediction, and a BOCPD-based regime-awareness signal. Our PPO-CVRFE approach consistently outperforms both the original PPO and passive strategies in empirical tests.

Due to time and computational resources limitations, we were unable to conduct a full ablation study to quantify the contribution of each component, nor to analyze per-window performance. Future work could include such analyses. In addition, BOCPD could also be used as an automated regime detection tool, replacing arbitrary fixed data splits. Finally, enabling asymmetric price range selection in the environment would make possible directional and price-prediction-driven strategies, potentially improving both fee income and risk-adjusted returns.

References

[1] Ryan Prescott Adams and David JC MacKay. Bayesian online changepoint detection. *arXiv preprint arXiv:0710.3742*, 2007.

[2] Matias Altamirano, François-Xavier Briol, and Jeremias Knoblauch. Robust and scalable bayesian online changepoint detection. In *International Conference on Machine Learning*, pages 642–663. PMLR, 2023.

[3] Zhou Fan, Francisco Marmolejo-Cossio, Daniel J Moroz, Michael Neuder, Rithvik Rao, and David C Parkes. Strategic liquidity provision in uniswap v3. *arXiv preprint arXiv:2106.12033*, 2021.

[4] Isabelle Guyon, Jason Weston, Stephen Barnhill, and Vladimir Vapnik. Gene selection for cancer classification using support vector machines. *Machine learning*, 46(1):389–422, 2002.

[5] Jason Milionis, Ciamac C Moallemi, Tim Roughgarden, and Anthony Lee Zhang. Automated market making and loss-versus-rebalancing. *arXiv preprint arXiv:2208.06046*, 2022.

[6] Hakan Pabuccu and Adrian Barbu. Feature selection with annealing for forecasting financial time series. *Financial Innovation*, 10(1):87, 2024.

[7] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[8] Tim Weingärtner, Fabian Fasser, Pedro Reis Sá da Costa, and Walter Farkas. Deciphering defi: A comprehensive analysis and visualization of risks in decentralized finance. *Journal of risk and financial management*, 16(10):454, 2023.

[9] Yuhang Xu and Eliseo Brini. Improving defi accessibility through efficient liquidity provisioning with deep reinforcement learning. *arXiv preprint arXiv:2501.07508*, 2024.

[10] Haochen Zhang, Xi Chen, and Lin F Yang. Adaptive liquidity provision in uniswap v3 with deep reinforcement learning. *arXiv preprint arXiv:2309.10129*, 2023.

Appendix A: Corrections and Contributions to the FinAI Starter Kit

We addressed two critical issues in the official FinAI starter kit implementation.[3].

- **Environment Reset:** In `custom_env.py`, the `self.x` (asset X quantity) was not reset between episodes, causing state contamination and unstable RL training.
- **Volatility Misalignment:** The `ew_sigma` (exponentially weighted volatility) feature lagged other data by up to 168 steps, leading to outdated volatility inputs and incorrect LVR calculations.

[3]Github Pull Request link