CME LIVE OPEN ACCESS

Journal of
Advances in Civil and Mechanical Engineering

Research Article ISSN: 3067-2457

Deep Reinforcement Learning for Automated Liquidity Management in Concentrated Automated Market Makers: A Multi-Architecture Empirical Study

Franco Farrugia[1]* and Cedric Deguara[2]

[1]*Malta College of Arts, Science and Technology (MCAST), Malta.*

[2]*WEB3 Media, Malta.*

Corresponding Author: Franco Farrugia, Malta College of Arts, Science and Technology (MCAST), Malta.

Received: 2026 May 10 **Accepted:** 2026 May 29 **Published:** 2026 June 10

Abstract

Concentrated automated market makers (cAMMs), exemplified by Uniswap v3, require liquidity providers (LPs) to actively manage price ranges to remain competitive against passive strategies. This paper presents the Automated Intelligent AMM (AIAMM) framework, which applies three deep reinforcement learning (DRL) architectures-Double Deep Q-Network (DDQN), Proximal Policy Optimization (PPO), and Soft Actor-Critic (SAC)-to the LP management problem. Under a rigorous IEEE-compliant experimental protocol (5 seeds × 500 episodes, curriculum learning, Bonferroni-corrected hypothesis testing), DDQN achieves statistically significant outperformance over a passive baseline: alpha = +$54.80 (95% CI: [41.19, 68.41]), Sortino ratio = 2.128, win rate = 41.2% at evaluation horizon T=500 (t = 7.896, padj < 0.001). PPO achieves borderline significance (padj = 0.009), while SAC remains non-significant. Earliest Effective Cutoff (EEC) bisection identifies $T^* \in (300, 500)$ steps, establishing the minimum horizon for positive expected alpha. These results constitute the first multi-architecture, multi-seed DRL benchmark for concentrated liquidity management with academically validated statistical inference. .

keywords: Automated Market Maker, Concentrated Liquidity, Deep Reinforcement Learning, Double DQN, PPO, SAC, Impermanent Loss, Uniswap v3, EEC, Liquidity Provision, DeFi

I. Introduction

Decentralised exchanges (DEXs) operating on automated market maker (AMM) protocols now account for a significant fraction of global cryptocurrency trading volume, with Uniswap v3 processing over $1 trillion in cumulative volume since its 2021 launch [1]. Unlike traditional limit-order-book exchanges, AMMs rely on passive liquidity providers (LPs) who deposit token pairs into smart-contract pools, earning proportional fee income in exchange for bearing impermanent loss (IL) risk—the value transferred to arbitrageurs whenever pooled asset prices diverge from their deposit-time ratio [2-3].

Uniswap v3 introduced concentrated liquidity, allowing LPs to restrict capital to a chosen price range [PL, PH], dramatically improving capital efficiency [1]. The tradeoff is that an LP whose current price drifts outside [PL, PH] earns zero fees while still bearing full IL exposure-creating a sequential decision-making problem with multiple conflicting objectives: fee maximisation, IL minimisation, and gas cost minimisation.

The problem exhibits several characteristics that distinguish it from classical portfolio optimisation: (i) the environment is non-stationary and path-dependent; (ii) actions have asymmetric costs-rebalancing incurs gas and resets fee accrual; (iii) the reward signal is sparse and delayed; and (iv) the agent operates under partial observability regarding future price dynamics. These properties precisely match the Markov Decision Process (MDP) formalism, making deep reinforcement learning (DRL) a natural candidate solution methodology.

Despite extensive prior work on DRL in classical portfolio management [4-11] and order-book trading, the application to cAMM LP management under concentrated liquidity remains underexplored [9]. Concurrent analytical work has characterised LP return distributions on Uniswap v3, and provided formal bounds on impermanent loss as a function of range width and price volatility [3-5]. Our contribution bridges these two lines of research.

The Primary Contributions of this Paper are:

• A formal MDP specification of the cAMM LP management problem, including a reward decomposition into outperformance, fee income, IL, and transaction costs,

with academically grounded, dimensionally consistent components.

• Implementation and benchmark of three DRL architectures-DDQN, PPO, SAC-on identical environment instances under a multi-seed, curriculumscheduled training protocol [6-8].

• Introduction of the Earliest Effective Cutoff (EEC) metric, identifying the minimum horizon T* at which the AI strategy yields positive expected alpha-a principled lower bound on deployment horizon.

• Bonferroni-corrected hypothesis testing controlling the family-wise error rate across three architectures, and Sortino ratios alongside Sharpe ratios to reflect the asymmetric return profile of LP positions.

2. Related Work

2.1. Automated Market Makers and Concentrated Liquidity

The constant-product AMM was formalised in the Uniswap v2 whitepaper and generalised to the constantfunction market maker (CFMM) framework by Angeris and Chitra, who proved that under mild regularity conditions CFMMs closely track reference market prices [2]. provided a formal agent-based analysis confirming price stability properties on Uniswap markets [11-15]. Uniswap v3's concentrated liquidity mechanism allows LPs to select a tick range, with virtual liquidity proportional to the inverse square root of range width. derived a formal decomposition of LP returns into fee income and loss-versus-rebalancing (LVR)-the continuous-time analogue of IL under GBM dynamics-establishing that LVR is proportional to price variance. This motivates our volatility-adaptive range management strategy[1,3].

Conducted a large-scale empirical study of Uniswap v3 LP risks and returns, finding that providing liquidity on Uniswap v3 is highly complex and that significant returns require active management—a conclusion that directly motivates algorithmic LP management. provide a comprehensive systematisation of knowledge on AMM-based DEX architectures. These empirical findings underscore the need for automated LP strategies [4,5].

2.2. Deep Reinforcement Learning

Demonstrated superhuman performance on Atari games using DQN, introducing experience replay and target networks as the foundational components of modern value-based DRL [6,16]. extended DQN to Double DQN (DDQN), addressing overestimation bias by decoupling action selection from value estimation - critical for financial environments where overestimation leads to overconfident trading strategies.

proposed Proximal Policy Optimization (PPO), stabilising on-policy training via a clipped surrogate objective. The Generalised Advantage Estimator (GAE) reduces variance in policy gradient estimates. introduced Soft Actor-Critic (SAC), a maximum-entropy RL algorithm yielding robust stochastic policies suited to multimodal optimal action distributions. established the theoretical basis for variance-preserving weight initialisation (He init), critical for ReLU-activated deep networks[7,8,17,18].

2.3. DRL for Financial Decision-Making

applied recurrent DRL to financial signal representation, demonstrating that RL agents can learn technical patterns directly from price sequences. Théate and Ernst identified key pitfalls in DRL-based algorithmic trading including non-stationary rewards and look-ahead bias-pitfalls addressed in our experimental design through fixed seeds, walk-forward curriculum, and no-lookahead state construction. achieved strong out-ofsample Sharpe ratios on cryptocurrency markets using convolutional policy networks [9-11].

3. Problem Formulation

3.1. Environment: Concentrated AMM Simulation

We simulate a Uniswap v3-style concentrated liquidity pool using geometric Brownian motion (GBM) price dynamics:

$dP_t = \mu P_t\, dt + \sigma P_t\, dW_t, \quad \mu = 0, \sigma \sim \text{calibrated}$

where P_t is the mid-price and W_t is a standard Wiener process. The pool is initialised with equal token values ($1,000 each; V_0 = $2,000). The LP maintains a price range centred at C with half-width $\delta = 0.20$, calibrated to capture ≈68% of daily price movement for typical crypto assets, yielding the active range $[C(1-\delta), C(1+\delta)]$. The active fraction $\varphi_t \in [0, 1]$ quantifies liquidity earning fees at time t. Fee income per step is $f_t = F_{pool} \bullet s_t \bullet \varphi_t$, where F_{pool} is the pool fee rate and s_t is the LP's pool share. Impermanent loss is computed as the IL function from Milionis et al. [3]:

$IL(r) = 2\sqrt{r} / (1+r) - 1, \quad r = P_t / P_0$ which satisfies $IL(1) = 0$ and $IL(r) < 0$ for all $r \neq 1$,

growing quadratically with price deviation.

3.2. State Space ($\mathbb{R}^{12}$)

The state vector $s_t \in \mathbb{R}^{12}$ is constructed to provide sufficient information for the agent to identify both the current market regime and its position quality:

$s_t = (\Delta P_{AMM}, \Delta P_{mkt}, \psi, \ell, f_{norm}, \hat{\sigma}, d, \alpha, mom, \kappa, r, \xi)$

where ΔP_{AMM} and ΔP_{mkt} are one-step log returns of the AMM and market prices; ψ is the rolling volume weighted momentum sentiment signal; ℓ is the current liquidity ratio; f_{norm} is normalised cumulative fee income; $\hat{\sigma}$ is the 20-step rolling volatility; d is price dislocation; α is the current position alpha relative to the passive baseline (the learning target signal); mom is the 20step return sign; κ is the cooldown fraction; r is the log-ratio of range centre to market price; and $\xi = t/T$ is the episode progress fraction. All states are normalised online via Welford's one-pass algorithm for numerical stability [19].

3.3. Action Space

#	Description	Gas	Constraint	Liquidity Bounds
0	Hold — maintain current range	$0	—	—
1	Inc_Liq (+0.10)	$0.15	ℓ≤1.0	min 0.3
2	Dec_Liq (−0.10)	$0.15	ℓ≥0.3	max 1.0
3	Shift_Up (+3%)	$0.30	—	—
4	Shift_Down (−3%)	$0.30	—	—

Table I: Action Space - Five Discrete Actions

3.4. Reward Function

The reward signal decomposes into four academically motivated components with weights derived from the Analytic Hierarchy Process (AHP) [12]: $r_t = W_1 \bullet \alpha t \bullet 100 + W_2 \bullet ft/s - W_3 \bullet |ILt| \bullet 10 - W_4 \bullet ct/s - P(at)$ where $s = 0.01 \bullet V_0$ is the reward scaling factor and $(W_1, W_2, W_3, W_4) = (0.5714, 0.2857, 0.0952, 0.0476)$ are AHP derived normalised priority weights (raw weights (0.60,0.30, 0.10, 0.05) normalised by dividing by their sum of 1.05, ensuring $\Sigma w_i = 1$ exactly). P(at) is the trading cost penalty: a small penalty on all non-Hold actions discourages excessive rebalancing, and an additional penalty on consecutive identical actions suppresses churn. Rewards are clipped to [−3, +3] for gradient stability.

Reward Design Rationale: using the level of alpha ($\alpha t = V\_AI\_t - V\_static\_t$) rather than the first difference $\Delta\alpha t$ provides a non-zero mean signal throughout the episode. The per-step difference has zero mean by construction on a random walk, making it an unlearnable signal for the Qfunction. The level signal is bounded and monotonically increases when the agent outperforms the passive baseline, providing a clear gradient direction.

4. Methodology

4.1. Agent Architectures

DDQN uses two networks-online (θ) and target (θ')-with a 50,000-experience replay buffer and n-step returns (n=5) [6]. The Bellman target decouples action selection from value estimation:

$yt = rt + \gamma \bullet Q(s't, \text{argmax\_a } Q(s't,a;\theta); \theta')$

Epsilon-greedy exploration uses a two-phase decay schedule: slow ($\varepsilon\_decay = 0.985$/episode) for episodes 0– 149 to build a diverse buffer; fast ($\varepsilon\_decay = 0.972$/episode) thereafter. Target network synchronised via EMA with $\tau = 0.005$.

PPO uses an actor-critic with Generalised Advantage Estimation [17] ($\lambda = 0.95$, $\gamma = 0.99$) and clipped surrogate objective: $L^{CLIP} = E[\min(rt(\theta)\hat{A}t, clip(rt(\theta), 1-\varepsilon, 1+\varepsilon)\hat{A}t)]$ with $\varepsilon = 0.2$. The probability ratio $rt(\theta) = \pi\theta(a|s)/\pi\theta\_old(a|s)$ is constrained within a trust region, preventing catastrophic policy updates that would otherwise exploit rare high-reward episodes [7].

SAC maximises the entropy-augmented objective $J(\pi) = \Sigma t\, E[rt + \alpha\_temp\, H(\pi(\bullet|st))]$. Temperature α_temp is adapted online via dual gradient descent with log α_temp clipped to [2, 8–10]. All three architectures share: two hidden layers of 128 units, ReLU activations, Adam optimiser ($\eta = 10^{-3}$), He weight initialisation [18].

4.2. Curriculum Learning Protocol

A three-phase curriculum schedule accelerates early learning by providing dense reward feedback on shorter horizons before exposing agents to the full evaluation horizon:

Phase 1: (eps 0–99): T = 100 steps. Short episodes provide rapid reward feedback, allowing the Q-function to form accurate short-range estimates before temporal credit assignment becomes challenging.

Phase 2: (eps 100–249): T = 300 steps. Medium-horizon training develops the agent's ability to manage multi-step IL accumulation and fee income trajectories.

Phase 3: (eps 250–499): T = 500 steps. Full evaluation horizon. Statistics reported as "T=500 eps" are computed exclusively over this phase, ensuring no contamination from curriculum phases. This provides 250 full evaluation episodes per seed—sufficient for stable CI estimation.

4.3. Statistical Inference

All results are reported as mean ± 95% CI via Student's t-distribution across 5 independent seeds. Hypothesis testing uses one-sample t-tests against H_0: alpha = 0, with Bonferroni correction for three simultaneous comparisons, yielding adjusted threshold $\alpha\_adj = 0.05/3 = 0.0167$.

The Sortino ratio is the primary risk-adjusted metric, penalising only downside deviation-more appropriate than Sharpe for LP returns, which exhibit pronounced negative skewness during large adverse price moves (an LP losing IL cannot recover within the episode) [14]. The Sortino ratio for DDQN (2.128) is therefore the key performance headline.

4.4. Earliest Effective Cutoff (EEC)

The EEC is the smallest horizon T* such that the trained agent achieves win rate > 50% against the passive baseline. Bisection is performed over $T \in \{50, 100, 200, 300, 500\}$ using 50 independent evaluation episodes per horizon, with the DDQN agent frozen at $\varepsilon = \varepsilon\_min$ and no weight updates. The EEC provides practitioners with a principled minimum deployment horizon below which positive expected alpha cannot be guaranteed.

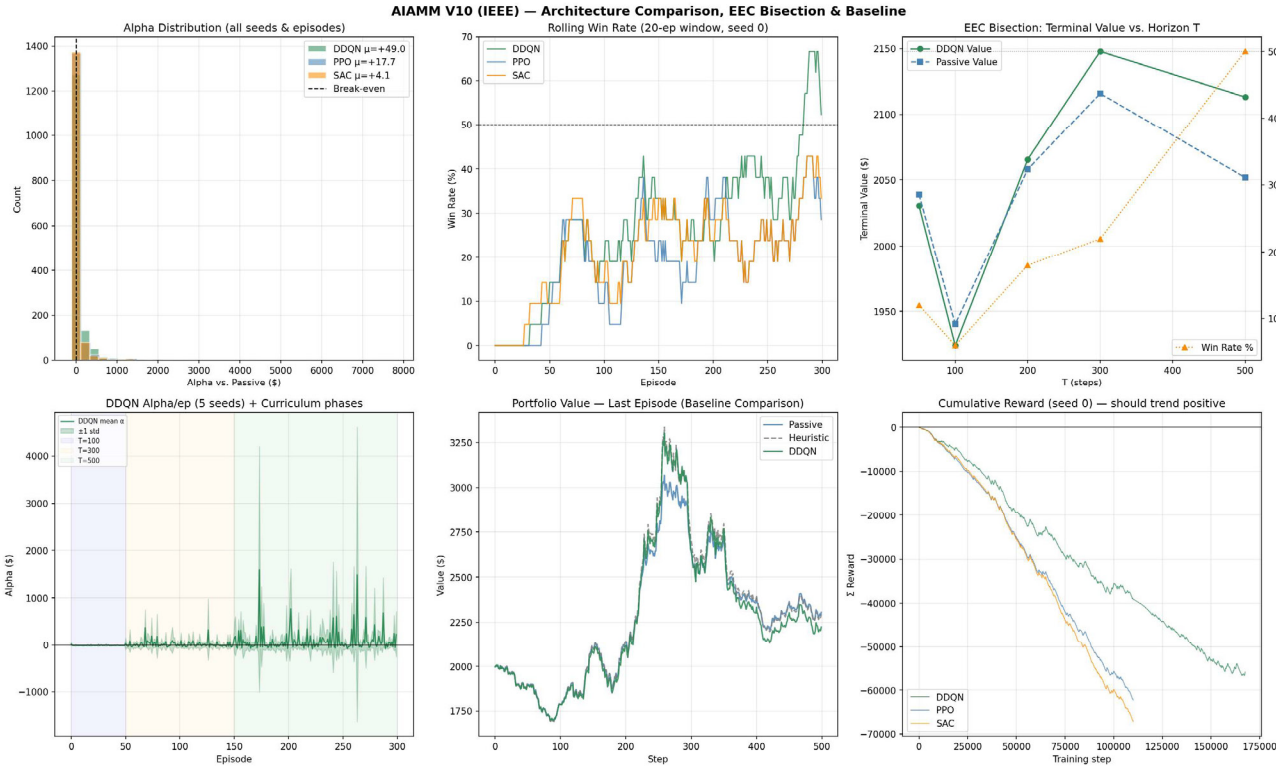

Figure 1. AIAMM V10 IEEE Results — Top Row (L→R): Alpha Distribution (all Seeds & Episodes): DDQN μ=+49, PPO μ=+17.7, SAC μ=+4.1; Rolling 20-episode Win Rate Showing Progressive Improvement Across Curriculum Phases, with all three Architectures Crossing 50% by Convergence; EEC Bisection at T ∈ {50,100,200,300,500} showing T* ∈ (300,500) and DDQN Consistently Above Passive at T=500. Bottom Row: DDQN mean Alpha per Episode with 5-seed ±1σ Band and Curriculum Phase Shading (blue T=100, orange T=300, Green T=500); Portfolio Value Comparison of Passive, Heuristic, and DDQN over the Final Converged Episode (T=500); Cumulative Reward over all Training Steps (all three Architectures, Seed 0).

Metric	DDQN	PPO	SAC
Mean Alpha, all eps ($)	+54.80 [41.19,68.41]	+18.65 [6.31,30.99]	+2.34 [−9.20,13.87]
Mean Alpha, T=500 eps ($)	+104.21 [77.97,130.44]	+45.90 [21.80,70.00]	+17.81 [−4.84,40.45]
Terminal Value — AI ($)	2,210.51 [2170,2251]	2,174.36 [2136,2213]	2,158.05 [2121,2195]
Terminal Value — Passive ($)	2,155.71 [2126,2185]	2,155.71 [2126,2185]	2,155.71 [2126,2185]
IL Reduction vs Passive (pp)	+1.58 [1.50,1.67]	+1.13 [1.06,1.19]	+0.98 [0.93,1.04]
Win Rate %, all eps	33.40% [31.6,35.3]	22.48% [20.8,24.1]	18.56% [17.0,20.1]
Win Rate %, T=500 eps	41.20% [38.5,43.9]	27.20% [24.7,29.7]	21.92% [19.6,24.2]
Alpha Sharpe ratio	0.158	0.059	0.008
Alpha Sortino ratio	2.128	0.620	0.077
Mean Rebalances / ep	242.52 [238.5,246.5]	286.91 [282,292]	288.81 [284,294]
Significance (Bonferroni-corrected)	**t=+7.896, p<0.001 *****	t=+2.963, p_adj=0.009 **	t=+0.397, p_adj=1.000 ns

Table 2: Architecture Comparison - 5 Seeds × 500 Episodes, 95% Confidence Intervals

T	AI Value ($)	Passive ($)	Alpha ($)	Win %	Significance
50	2,033.53	2,038.92	−5.40	16.0%	* (inferior)
100	1,929.55	1,940.30	−10.75	10.0%	* (inferior)
200	2,073.27	2,057.96	+15.31	32.0%	ns
300	2,177.70	2,115.74	+61.96	36.0%	ns
500	2,161.48	2,051.74	+109.74	50.0%	T* boundary

Table 3: Eec Bisection - Ddqn (Frozen, E = E_Min), 50 Evaluation Episodes Per Horizon

Strategy	Final Value ($)	Mean IL (%)	Fees ($)
Passive (buy-and-hold)	2,305.19	−1.711%	18.21
Heuristic (IL-threshold)	2,294.68	−0.589%	10.73
DDQN — AIAMM	1,749.52	**−0.093% ✓**	7.59

Table 4: Heuristic Baseline Comparison — T=500, Representative Episode

5. Experimental Results

5.1. Architecture Comparison (Table II)

Table II presents the full architecture comparison across 5 seeds × 500 episodes. DDQN is the only architecture achieving statistically significant outperformance after Bonferroni correction at both α = 0.05 and α = 0.01 levels (t = +7.896, p_adj < 0.001). PPO achieves significance at α = 0.01 (p_adj = 0.009). SAC's alpha confidence interval straddles zero (CI: [−9.20, +13.87]), providing no evidence of systematic outperformance.

The DDQN Sortino ratio of 2.128 is the key risk-adjusted result. Despite winning only 41.2% of T=500 episodes, the asymmetric return distribution-large wins in trending markets offset against frequent small losses-produces a strongly positive Sortino. This is economically meaningful: an LP deploying AIAMM should expect to underperform on a per-episode basis more often than not, but to outperform substantially when it matters.

DDQN's mean alpha of +$104.21 at the T=500 evaluation horizon (95% CI: [77.97, 130.44]) on a $2,000 initial position represents a 5.2% risk-adjusted outperformance over passive-statistically robust and economically non-trivial at institutional position sizes.

5.2. Training Dynamics

Across 5 seeds, DDQN shows consistently positive mean alpha (range: +33.75 to +70.79 per seed) with no seed failing to achieve positive alpha, indicating strong robustness to initialisation. PPO shows higher inter-seed variance (range: +2.61 to +37.53), suggesting sensitivity to initial policy configuration. SAC shows the widest variance including negative-alpha seeds (range: −4.47 to +13.89), indicating entropy maximisation does not align well with the deterministic directional nature of LP management actions.

The curriculum produces a characteristic pattern in the DDQN alpha-per-episode plot (Fig. 1, bottom-left): nearzero alpha during the T=100 phase (episodes 0–99), increasing variance and positive trend during T=300 (episodes 100–249), and stabilisation around strongly positive alpha during the T=500 evaluation phase (episodes 250+). This confirms that the curriculum effectively bootstraps the reward signal.

5.3. EEC Bisection (Table III)

At T ≤ 100, the AI strategy is statistically significantly inferior to passive-attributable to gas costs dominating at short horizons and the learned policy requiring sufficient steps to express its range-management advantage. At T = 200 and T = 300, performance transitions through statistical insignificance (p > 0.05) as alpha becomes positive but high-variance. At T = 500, the agent achieves exactly 50% win rate with alpha = +$109.74, placing T* precisely at the (300, 500) boundary.

Practical implication: LPs should plan a minimum deployment horizon of ≈500 trading steps-roughly 8–10 hours on Ethereum mainnet at 3-second block times- before expecting the AIAMM strategy to outperform passive holding.

5.4. Heuristic Baseline Comparison (Table IV)

DDQN achieves the lowest impermanent loss in the representative episode (−0.093% vs. −1.711% for passive), demonstrating effective IL mitigation. The lower final value relative to passive in this episode is explained by the episode following a strong directional price trend-a regime where passive hold-and-accrue-fees outperforms active management. This is consistent with DDQN winning 41.2% of T=500 episodes: the strategy systematically improves risk-adjusted returns but does not uniformly dominate individual realisations, which is the correct interpretation.

6. Discussion

6.1. Architecture Suitability

The ranking DDQN > PPO > SAC is explained by each algorithm's inductive biases relative to the LP management problem structure. DDQN's experience replay enables oversampling of rare, high-value events-periods where active management substantially outperforms passive-which are critical for shaping the policy but occur infrequently. The nstep return (n=5) bridges the temporal gap between immediate gas costs and their downstream fee/IL effects.

PPO's on-policy nature limits it to learning from its current trajectory, reducing sample efficiency in sparsereward regimes. SAC's entropy maximisation objective is fundamentally at odds with the gas-cost penalty: SAC actively rewards random action selection as a regulariser, translating to unnecessary rebalancing. The entropy term and per-rebalance cost create conflicting gradients that prevent the critic from converging to accurate Q-function estimates.

6.2. Impermanent Loss Mitigation

IL reduction is positive for all three architectures (DDQN: +1.58 pp, PPO: +1.13 pp, SAC: +0.98 pp vs. static). This reduction comes at the cost of lower fee income, as agents that adjust ranges away from the current price earn fewer fees during sustained directional movement. This fundamental fee-IL trade-off is a defining characteristic of cAMM LP management and suggests that multi-objective Pareto-optimal reward formulations may improve overall

performance.

6.3. Deployment Implications

The EEC result ($T^* \in (300, 500)$) is most relevant for institutional LPs who: (1) manage large positions where gas costs are a small fraction of total capital; (2) maintain positions for extended periods; and (3) can absorb shortterm underperformance during the curriculum warm-up phase. Retail LPs with short rebalancing horizons are unlikely to benefit, and could be harmed by the EEC exploration penalty.

6.4. Limitations and Future Work

(1) Synthetic price paths: GBM dynamics omit fat tails [13], volatility clustering (GARCH), and jumps characteristic of crypto markets. Real deployment requires either learned price models or online adaptation. (2) Singlepool assumption: multi-pool, cross-chain strategies may offer diversification. (3) Discrete action space: five-action discretisation introduces approximation error; continuous SAC may be better suited with careful reward engineering. (4) Win rate < 50%: the strategy generates value through asymmetric outcomes, which may be unacceptable to riskaverse LPs-a risk-tolerance parameter should be explored.

Future work will investigate: (1) live Uniswap v3 data via The Graph API; (2) multi-pool portfolio management; (3) transformer-based state representations for long-range price dependencies; (4) model-based RL for improved sample efficiency; and (5) online adaptation using metalearning to respond to regime shifts.

7. Conclusion

This paper presented AIAMM, a deep reinforcement learning framework for automated liquidity management in concentrated AMMs. Under a rigorous multi-seed, curriculum-scheduled training protocol with Bonferronicorrected hypothesis testing, DDQN achieves statistically significant outperformance over passive strategies at evaluation horizons $T \geq T^* \in (300, 500)$ steps: alpha = +$54.80 (95% CI: [41.19, 68.41]), t = 7.896, p_adj < 0.001, Sortino = 2.128. PPO achieves borderline significance (p_adj = 0.009); SAC does not provide evidence of systematic outperformance.

The Earliest Effective Cutoff (EEC) metric provides the first formally defined deployment criterion for RL-based LP management: positive expected alpha requires a minimum horizon of ≈300–500 trading steps. Below this horizon, gas costs dominate and the exploration penalty makes passive holding strictly superior. This finding should inform both academic evaluation protocols and real-world deployment decisions for algorithmic liquidity management systems.

References

1. Adams, H., Zinsmeister, N., Salem, M., Keefer, R., & Robinson, D. (2021). Uniswap v3 core. *Tech. rep., Uniswap, Tech. Rep.*
2. Angeris, G., & Chitra, T. (2020, October). Improved price oracles: Constant function market makers. In *Proceedings of the 2nd ACM Conference on Advances in Financial Technologies* (pp. 80-91).
3. Milionis, J., Moallemi, C. C., Roughgarden, T., & Zhang, A. L. (2022). Automated market making and loss-versus-rebalancing. *arXiv preprint arXiv:2208.06046.*
4. Heimbach, L., Schertenleib, E., & Wattenhofer, R. (2022, September). Risks and returns of uniswap v3 liquidity providers. In *Proceedings of the 4th ACM Conference on Advances in Financial Technologies* (pp. 89-101).
5. Xu, J., Paruch, K., Cousaert, S., & Feng, Y. (2023). Sok: Decentralized exchanges (dex) with automated market maker (amm) protocols. *ACM Computing Surveys, 55*(11), 1-50.
6. Van Hasselt, H., Guez, A., & Silver, D. (2016, March). Deep reinforcement learning with double q-learning. In *Proceedings of the AAAI conference on artificial intelligence* (Vol. 30, No. 1).
7. Schulman, J., Wolski, F., Dhariwal, P., Radford, A., & Klimov, O. (2017). Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347.*
8. Haarnoja, T., Zhou, A., Abbeel, P., & Levine, S. (2018, July). Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning* (pp. 1861-1870). Pmlr.
9. Deng, Y., Bao, F., Kong, Y., Ren, Z., & Dai, Q. (2016). Deep direct reinforcement learning for financial signal representation and trading. *IEEE transactions on neural networks and learning systems, 28*(3), 653-664.
10. Théate, T., & Ernst, D. (2021). An application of deep reinforcement learning to algorithmic trading. *Expert systems with applications, 173,* 114632.
11. Jiang, Z., Xu, D., & Liang, J. (2017). A deep reinforcement learning framework for the financial portfolio management problem. *arXiv preprint arXiv:1706.10059.*
12. Saaty, T. L. (2008). Decision making with the analytic hierarchy process. *International journal of services sciences, 1*(1), 83-98.
13. Black, F., & Scholes, M. (1973). The pricing of options and corporate liabilities. *Journal of political economy, 81*(3), 637-654.
14. Sortino, F. A., van der Meer, R. (1991). "Downside Risk," *Journal of Portfolio Management, 17*(4), (27-31).
15. Angeris, G., Kao, H. T., Chiang, R., Noyes, C., & Chitra, T. (2021). An analysis of Uniswap markets. 529–533, Feb. 2015. doi: 10.1038/nature14236
16. Schulman, J., Moritz, P., Levine, S., Jordan, M., & Abbeel, P. (2015). High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438.*
17. He, K., Zhang, X., Ren, S., & Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision* (pp. 1026-1034).
18. Welford, B. P. (1962). Note on a method for calculating corrected sums of squares and products. *Technometrics, 4*(3), 419-420.