



Research project defense

Modeling and Evaluation of **Adaptive Liquidity** Strategies in Uniswap V4

Моделирование и оценка адаптивных стратегий управления ликвидностью в Uniswap V4

AUTHOR

Amira R. Khabibullina
group BPAD234, 3rd year

SUPERVISOR

Yury A. Yanovich
Associate Professor, FCS · PhD



Moscow · 2026



Decentralized exchanges and automated market makers

Trades settle on-chain through the AMM formula rather than an order book. Uniswap's evolution is a steady increase in capital efficiency and pool programmability.

V2

Constant product

Invariant $x \cdot y = k$; liquidity spread uniformly across the whole curve — simple, but capital is used inefficiently.

V3

Concentrated liquidity

Capital in a range $[Pa, Pb]$ with fee tiers. More efficient, but a position goes idle outside its range and needs manual management.

V4

Singleton + hooks

A single PoolManager, flash accounting (EIP-1153) and programmable hooks — adaptive logic embedded in the pool lifecycle.

Liquidity providers systematically lose to adverse selection

Arbitrageurs exploit stale pool prices, execute near risk-free trades and transfer value from LPs to themselves. This effect is formalized as **Loss-Versus-Rebalancing (LVR)**.

In Uniswap V3 a competitive position needs manual range updates or external bots: latency, gas overhead and extra trust assumptions.

>50%

of retail LPs in large V3 pools underperform passive holding once costs are accounted for

Research question

Can protocol-native adaptivity in V4 improve the LP outcome after gas?

GOAL Assess whether protocol-native adaptivity in Uniswap V4 — adaptive fees and ranges via hooks — improves the liquidity provider's net outcome after gas, compared with passive and externally managed strategies.

- 01** Study the V4 architecture (singleton, flash accounting, hooks) and the formalization of LVR / toxic flow.
- 02** Develop an adaptive fee-hook: a Solidity reference and an algebraically identical Python model.
- 03** Build an event-driven backtesting engine on Fractal and the data layer (data bridge).
- 04** Define six strategy classes and metrics: net PnL, RV, FARV, LVR reference.
- 05** Run rolling validation on ETH/USD history and test the statistical hypotheses.
- 06** Assess hook risks: threat matrix, dynamic Foundry tests, JIT side channel.

Adaptivity is already baked into protocols or offloaded to external bots

V4 offers a third path — programmable, pool-native logic without modifying the core.

SOLUTION	AUTHORS / COMPANY	APPROACH	LIMITATION
Curve · CryptoSwap	Curve Finance	Internal oracles, dynamic concentration	Logic hard-wired into the core, not extensible
Algebra Integral	Algebra Finance	Dynamic fees at the protocol level	Tied to a specific DEX implementation
V3 + external bots	Arrakis, Gamma, etc.	Off-chain range rebalancing	Latency, cross-call gas, trust
DRL agents	Zhang+ (2023), Xu & Brini (2025)	Reinforcement learning, LVR in the reward	Needs an interactive multi-agent environment
LVR · RV / FARV	Milionis+ (2022), Tangri+ (2023)	Adverse-selection metrics	Require a continuous price oracle

Requirements for the prototype

- **Fee adaptation**
adjust the swap fee to realized volatility inside the swap lifecycle
- **Deterministic backtest**
reproducible replay of price history and swap events with a fixed seed
- **Strategy comparison**
at least six classes on identical observations and capital
- **Two identical implementations**
an on-chain Solidity hook and an algebraically equivalent Python model
- **Metric suite**
net PnL after gas, 50/50, RV, FARV, range activity, rebalance count
- **Statistics and risks**
binomial tests, bootstrap CI; hook threat matrix and JIT model

Interpretable rules, anchored to the literature

Volatility estimate — EWMA ($\alpha = 0.20$)

Recursive and lightweight on-chain — no need to store an observation window.

Rule-based, not RL

The environment is single-agent and open-loop: without interactive feedback RL collapses into regression.

Framework — Fractal

Unified strategy interface, MLflow experiment tracking and a grid-runner for parameter sweeps.

Fee policy — saturating linear

Interpolation $f_{\min}=5 \rightarrow f_{\max}=100$ bps over σ thresholds plus a flow-surge premium.

Metrics from the literature — LVR, RV, FARV

LVR (Milionis+) as a conceptual reference; RV / FARV (Tangri+) as strict diagnostics.

Adaptive **fee-hook**: the fee follows volatility

PoolManager calls the hook twice per swap; flash accounting nets every balance change within a single transaction.

i · beforeSwap

The hook reads its EWMA state and returns the fee in basis points.

ii · execute

PoolManager applies the fee to the swap; balances net within one transaction.

iii · afterSwap

The hook updates the EWMA variance from the new mid price; state is persisted.

$\alpha = 0.20$

$f_{\min} = 5 \text{ bps}$

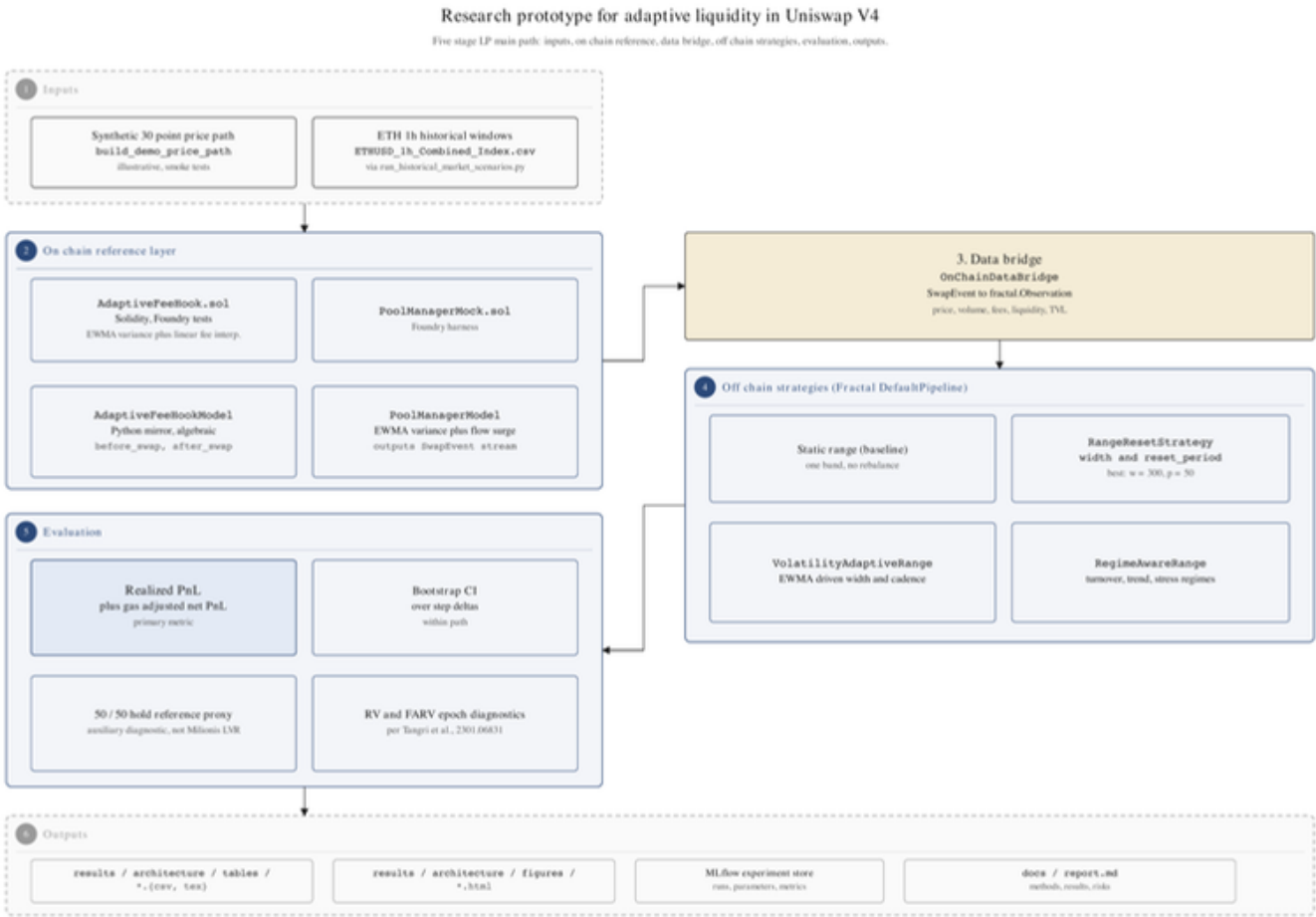
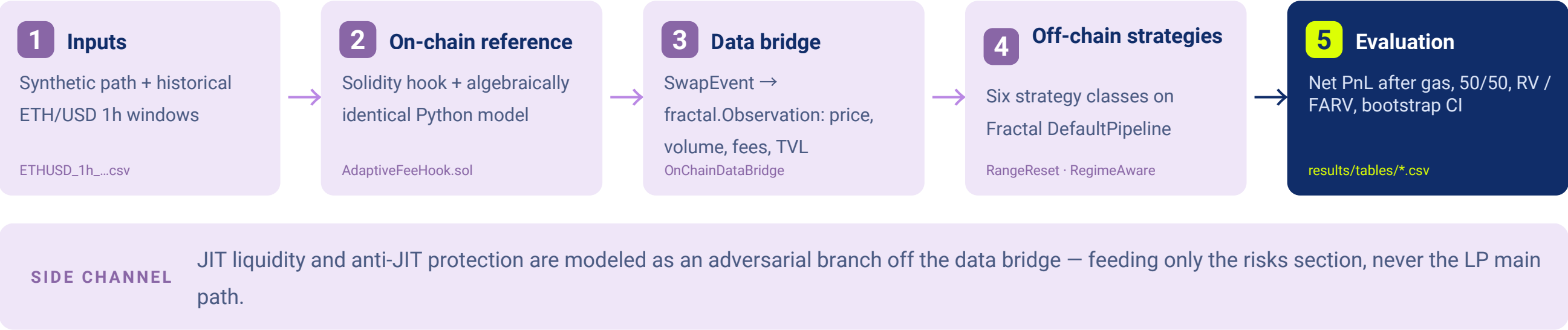
$f_{\max} = 100 \text{ bps}$

$\sigma_{\text{low}} = 2 \cdot 10^{-3}$

$\sigma_{\text{high}} = 3 \cdot 10^{-2}$

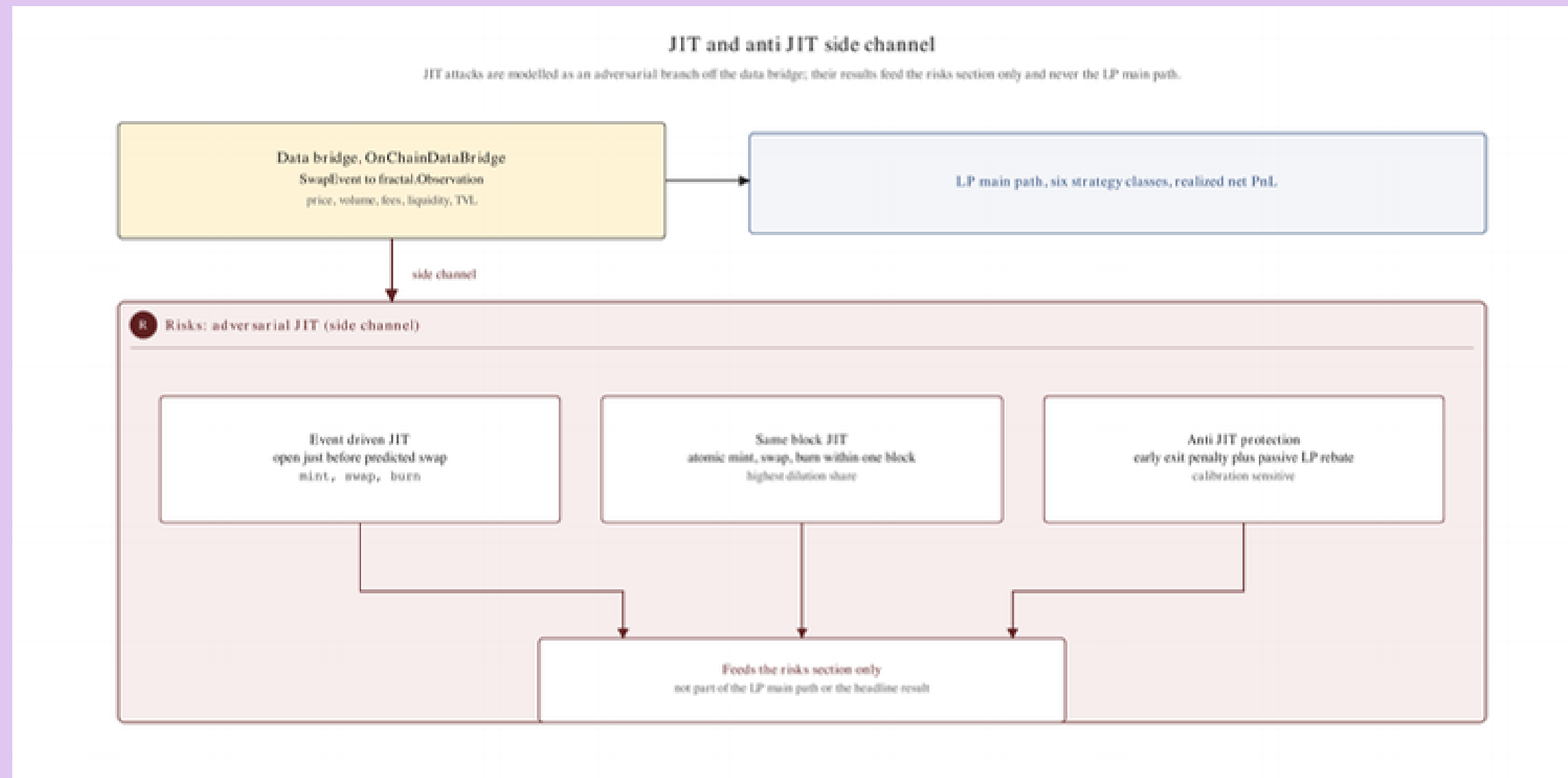
+ flow-surge premium

A five-layer research prototype



Security, protocol risk, and stress limits

the project evaluates not only profit, but also where the architecture remains vulnerable or incomplete.



Jit side channel

modelled separately from the core LP
comparison

The main open problem is still stress behaviour: very adverse windows remain difficult for the current rule-based policy.



Tools and validation design

IMPLEMENTATION TOOLS

Solidity · Foundry	<i>on-chain reference, forge tests</i>
Python	<i>mirror model, backtest</i>
Fractal · MLflow	<i>pipeline, grid-runner, tracking</i>
pytest · bootstrap	<i>20 unit tests, statistics</i>

EXPERIMENT PLAN

9 yrs
ETH/USD, 1-hour frequency

110
non-overlapping 7-day windows

6
strategy classes

PnL
net after gas — primary metric

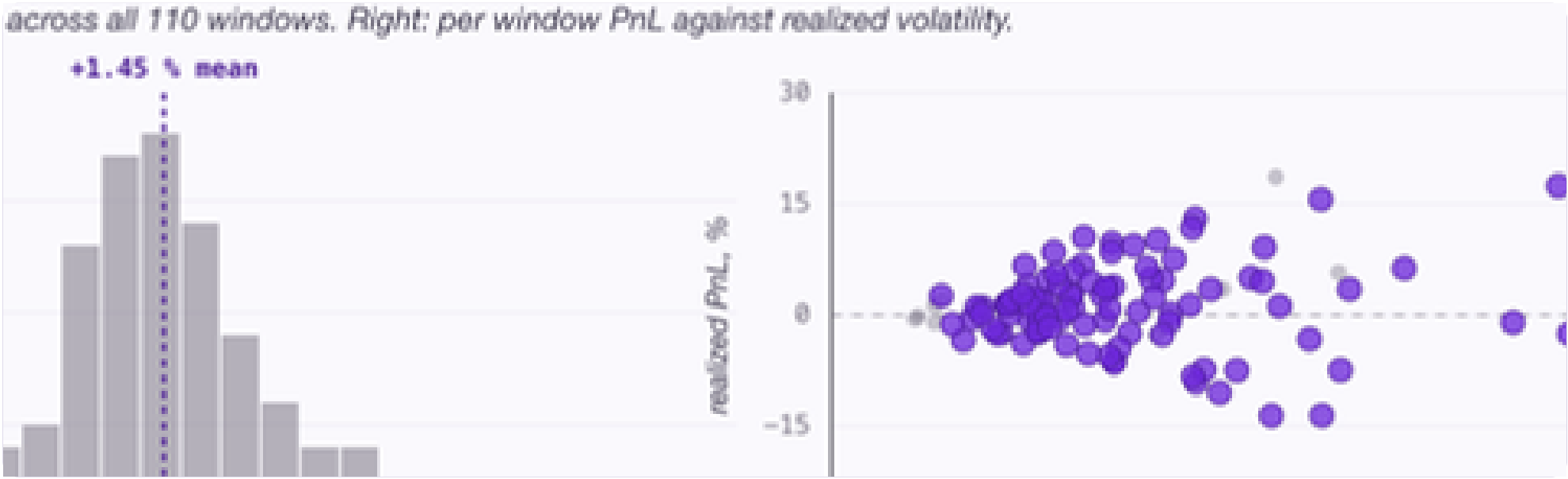
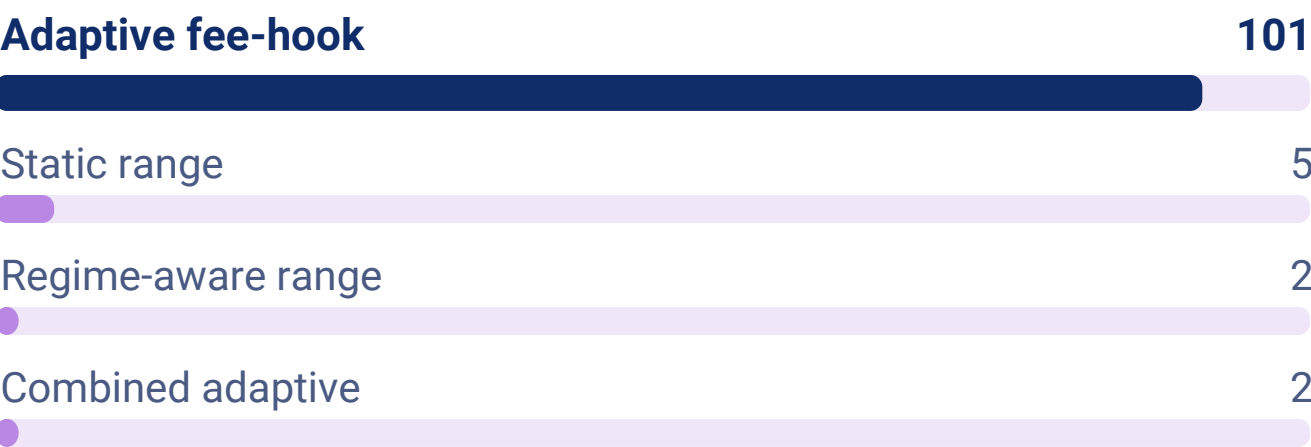
Capital 1000 USD, gas 2 USD per on-chain action, 30-day window stride — no manual interval selection.

The adaptive fee-hook wins 101 of 110 windows

101 / 110

Windows won · mean **+1.32 %** · median **+1.18 %** · positive in **60.4 %**

WINS BY STRATEGY CLASS



The edge comes at the **fee** level, not range geometry — at 1000 USD capital the gas of frequent rebalancing eats the benefit.

Three hypotheses on the same 110 windows

H1a

Strategy dominance

$H_0: p = 0.5 \rightarrow H_1: p > 0.5$ $k = 101 / 110$, binomial

$p \approx 3.92 \cdot 10^{-21}$

reject H_0

H1b

Positive profitability

$H_0: p = 0.5 \rightarrow H_1: p > 0.5$ $k = 66 / 110$, binomial

$p = 0.022$

reject H_0

H1c

Positive effect size

$H_0: \mu \leq 0 \rightarrow H_1: \mu > 0$ bootstrap $B = 10\,000$

$CI_{95} = [+0.17; +2.75] \%$

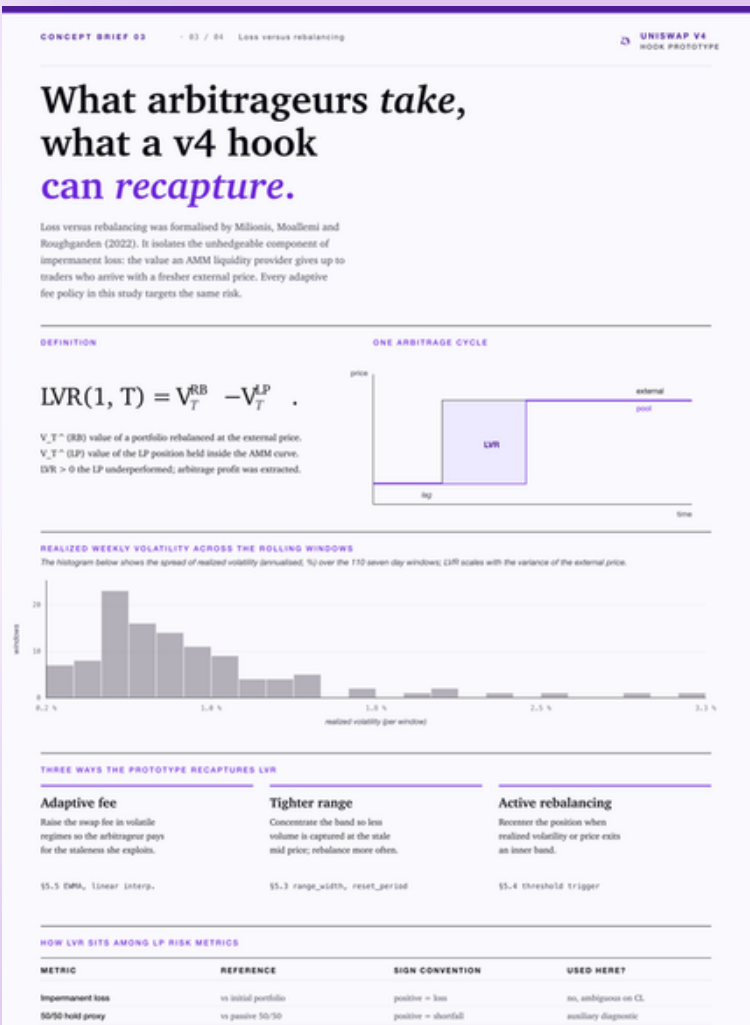
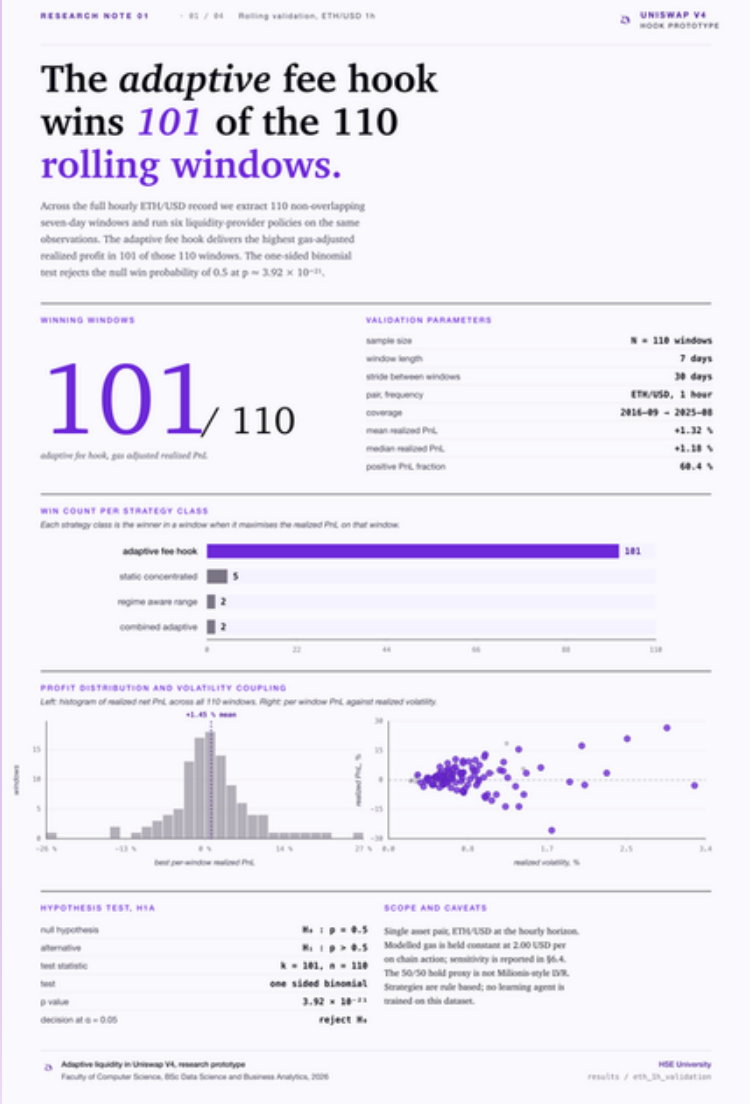
CI above zero

All three null hypotheses are rejected at the 5 % level. Mean across wins **+1.45 %** ; seed = 42, results reproducible from the project scripts and notebook.

A research paper and infographics

The project also produced an IEEE-format research paper: «Modeling and Evaluation of Adaptive Liquidity Strategies in Uniswap V4» .

- Four-layer prototype: **Solidity** + Python mirror
- Unified evaluation layer: realized PnL, 50/50, RV / FARV
- Rolling validation, binomial tests, bootstrap
- HookScope-style security assessment of the hook



Modeling and Evaluation of Adaptive Liquidity Strategies in Uniswap V4

Amira Khabibullina

Abstract—Liquidity providers in Uniswap V3 frequently underperform passive holding when prices move outside their chosen range or when arbitrage imposes loss versus rebalancing. Uniswap V4 introduces a singleton pool architecture and programmable hooks, which make it possible to adjust the swap fee and the active range from within the pool lifecycle itself. We build a four layer research prototype that pairs a Solidity reference hook with an algebraically equivalent Python mirror, a Fractal based off chain back testing layer, and an evaluation layer that reports realized gas adjusted PnL, together with a 50/50 hold reference proxy and the article grounded relative value and fee adjusted relative value of Tangri et al. The prototype is validated on 110 non overlapping 7 day windows of hourly ETH/USD data with no hand picking. The *Adaptive fee* hook is the realized net PnL winner in 101 of 110 windows; a one sided binomial test against the null $H_0: p = 0.5$ yields $p = 3.92 \times 10^{-11}$, and a 95% bootstrap confidence interval on the mean best realized net PnL is [+0.174, +2.750]%, strictly above zero. Just in time liquidity attacks and anti JIT protection are modelled as an adversarial side channel and do not enter the LP main path.

Index Terms—Uniswap V4, programmable hooks, adaptive liquidity, rebalancing, dynamic fees, security analysis, loss versus rebalancing, hypothesis testing.

I. INTRODUCTION

Decentralised exchange design in DeFi has evolved from the constant product invariant of Uniswap V2 [1], [2], where liquidity is spread uniformly along the price curve, to the concentrated liquidity design of Uniswap V3 [1], in which a liquidity provider (LP) allocates capital inside a chosen price interval $[P_s, P_t]$. Concentration improves capital efficiency but makes the LP inactive whenever the spot price exits its range. Uniswap V4 extends this evolution with two new architectural ingredients [2]: a singleton *PoolManager* contract that hosts every pool, and *hooks*, external contracts attached to a pool that execute user defined logic at well defined lifecycle points such as *beforeSwap*, *afterSwap*, and liquidity modification callbacks. A third ingredient, flash accounting based on EIP 1153 transient storage [3], [4], nets intermediate balance changes inside one transaction and settles only the final delta. Together these mechanisms make it cheap to embed adaptive logic inside the pool itself.

The phenomenon of *static flow* and its formalisation as Loss Versus Rebalancing (LVR) [5] captures one of the central frictions faced by LPs in modern AMMs. Arbitrageurs exploit temporary price staleness on chain, execute effectively riskless trades, and transfer value from passive LPs to themselves. Empirical studies of Uniswap V3 LP returns [6]–[8] report that providing liquidity in unfavourable regimes can underperform a passive 50/50 hold of the same notional after gas costs. The

expected practical significance of V4 hooks is therefore to reduce stale liquidity exposure and improve fee capture by adjusting fees and range geometry inside the pool lifecycle [9], rather than relying on third party bots whose latency and gas overhead are well documented [10].

Scope. This paper does not propose a deep reinforcement learning agent. The scope is the comparison of rule based hook policies against a static V3 like baseline and an externally rebalanced alternative. Reinforcement learning based extensions [11], [12] are listed only as future work.

Research question. Do adaptive fee and adaptive range policies improve LP realized net PnL relative to a static range, fixed fee baseline on a rolling ETH 1h validation, after subtracting modelled gas?

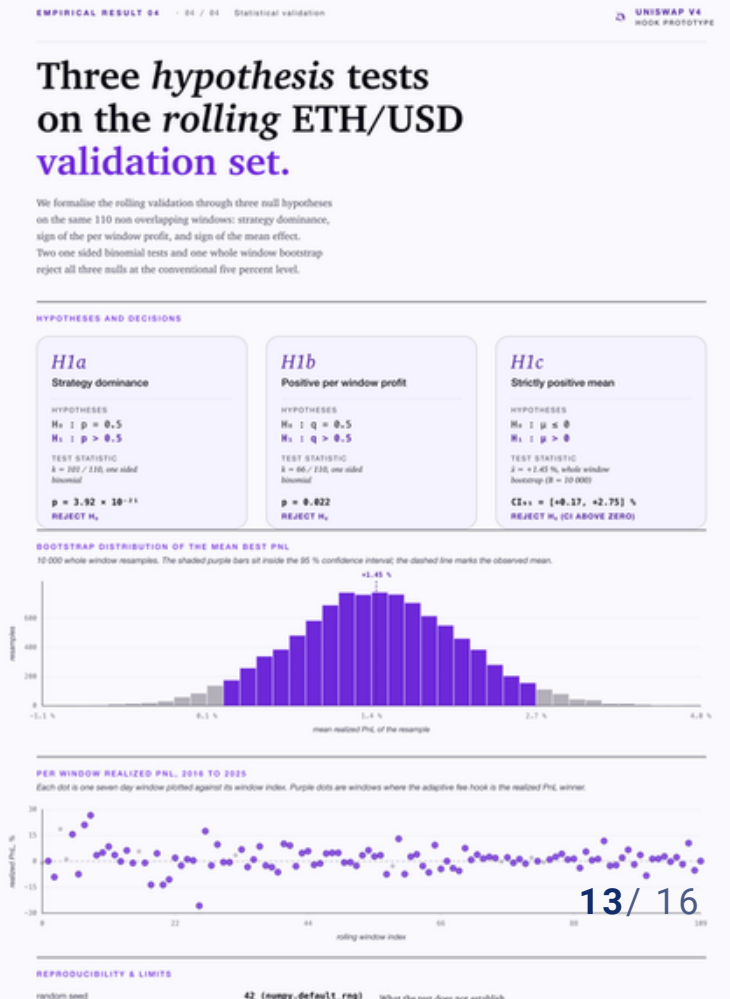
Contributions. The paper makes four contributions:

- A four layer research prototype that pairs a Solidity reference hook (Foundry tested) with an algebraically identical Python mirror suitable for fast back testing (Figure 1).
- A unified evaluation layer that reports realized PnL, a 50/50 hold reference proxy, and the article grounded RV and FARV diagnostics of Tangri et al. [13] on the same trajectories.
- A rolling ETH 1h validation over 110 non overlapping 7 day windows that removes hand picking bias and supports a binomial test of wins together with a whole window bootstrap confidence interval on the headline effect size.
- A HookScope inspired security evaluation of the reference hook's lifecycle entry points [14]–[16].

II. BACKGROUND AND RELATED WORK

The Uniswap whitepapers [1], [2] define the constant product invariant and the tick representation of price, $p(i) = 1.0001^i$. Concentrated liquidity changes the operational profile of an LP position: the position is active only when the spot price stays inside the chosen tick interval. Heimbach and colleagues [6] and Goyal and colleagues [7], [8] document empirically that a large share of V3 LPs underperform a passive hold once fees are weighed against impermanent loss.

Milosin, Moullemi and Roughgarden [5] formalise LVR as the systematic loss of an AMM LP against a rebalancing strategy with a continuous external reference price. LVR is fundamentally distinct from a 50/50 hold shortfall: it measures the unhedgeable component of divergence loss driven by adverse selection and depends only on the path of the reference price and the liquidity density at that price, not on the LP's initial portfolio mix. The prototype reports a 50/50 hold



14 · MAIN RESULTS AND CONCLUSIONS

V4 as programmable infrastructure for autonomous liquidity

101/110

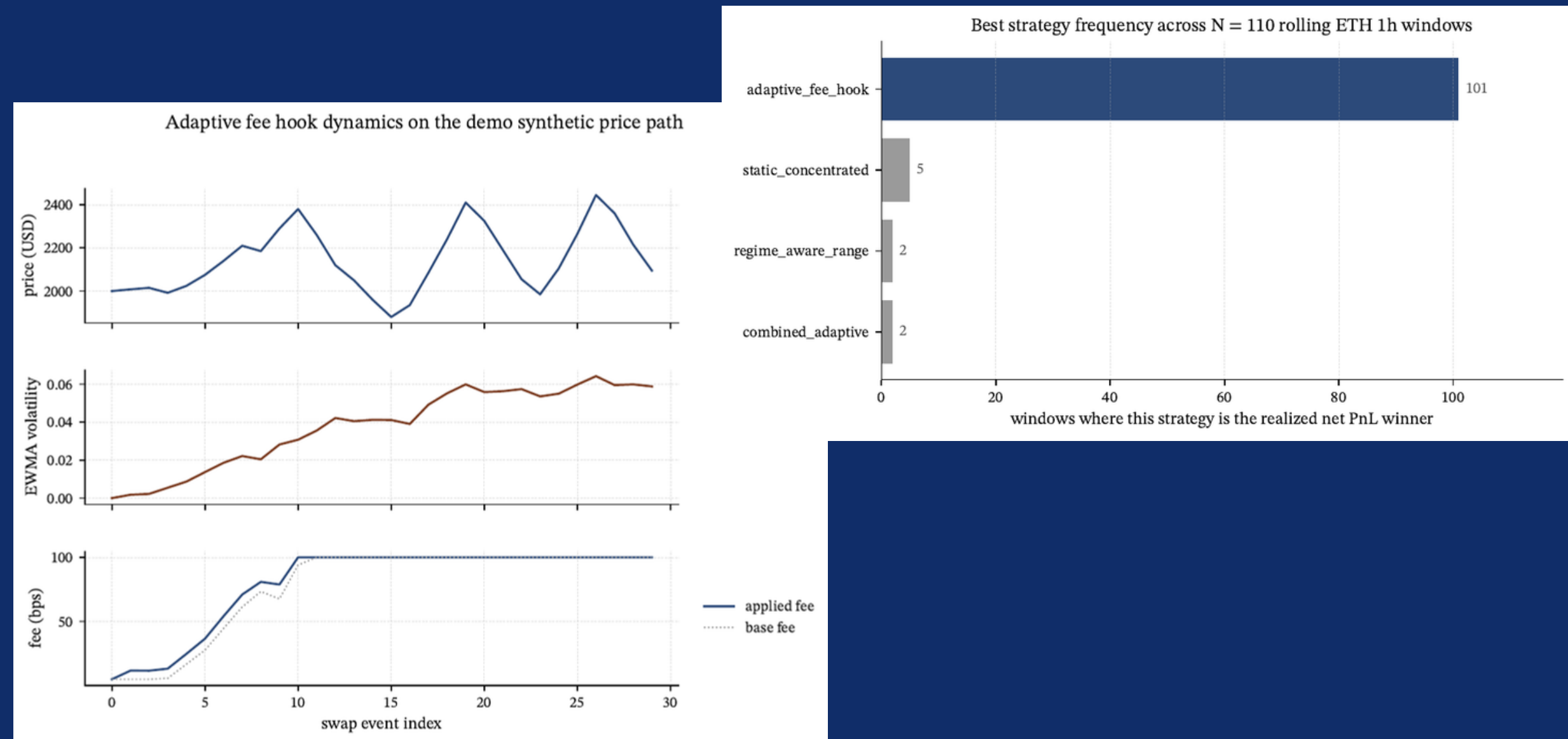
The adaptive fee-hook is the rolling-validation winner; dominance is statistically confirmed.

fee > range

At retail capital the edge comes from fee adaptivity, not range geometry — gas decides.

method

A reproducible methodology for comparing native adaptivity with off-chain management on economic metrics.



The conclusion holds within the experiment's scope: a single ETH/USD pool, a constant gas model and rule-based policies. The results establish feasibility and comparative behavior, not production readiness.

Statistical validation and market regimes

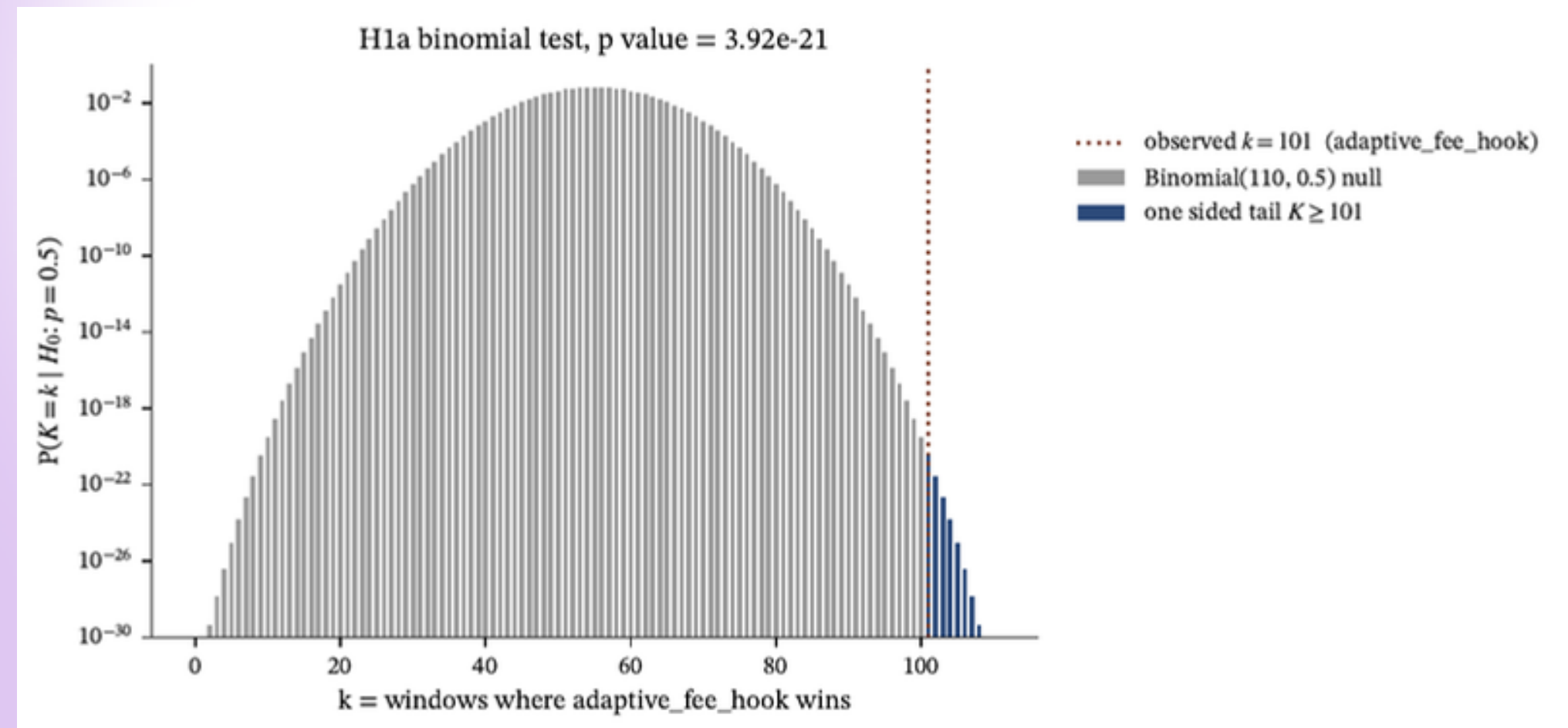


Statistical layer

- H1a, H1b, and H1c are all rejected on the rolling validation set
- the binomial tests are used for window counts, while bootstrap is used for the mean effect size

Market interpretation

- mixed and trend-up windows are the most favourable
- calm windows are close to neutral
- stress windows remain the main weakness
- strict LP profitability stays harder than raw realized pnl



Where to take the prototype

On-chain integration

a unified fee + range hook, EVM fork tests, guards against parameter oscillation

Adversarial testing

fuzzing, invariant testing, trigger manipulation and toxic flow

Market microstructure

endogenous order-flow response to fee and range changes

Cross-pool and regimes

tests on BTC/USD, stable-to-stable and pools of varying depth

Multi-agent environment + RL

closed loop, PPO / SAC against rule-based baselines

Deployment readiness

gas budgets, trigger monitoring, fail-safe and emergency shutdown

Key references

[1]

Adams H. et al. **Uniswap v4 Core**. Uniswap Labs, 2024.

[3]

Milionis J., Moallemi C., Roughgarden T. **Automated Market Making & Arbitrage Profits...** arXiv:2305.14604, 2025.

[5]

Fan Z. et al. **Strategic Liquidity Provision in Uniswap V3**. AFT, 2023.

[7]

Oh H., Lee S. **Lifecycle-aware security evaluation of DeFi hooks...** Computer Networks, 2026.

[9]

Xu H., Brini A. **Improving DeFi Accessibility... with DRL**. arXiv:2501.07508, 2025.

[2]

Adams H. et al. **Uniswap v3 Core**. 2021.

[4]

Tangri R. et al. **Generalizing Impermanent Loss on DEX with CFMMs**. arXiv:2301.06831, 2023.

[6]

Akhunov A., Salem M. **EIP-1153: Transient storage opcodes**. 2018.

[8]

Zhang H., Chen X., Yang L. **Adaptive Liquidity Provision in Uniswap V3 with DRL**. arXiv:2309.10129, 2023.

[10]

Daian P. et al. **Flash Boys 2.0**. arXiv:1904.05234, 2019.

The full list of references is provided in the coursework report.