Operator Fingerprinting

Recovering and Defending the Hidden Rebalancing Policy of Concentrated Liquidity Managers

K.R. Ryan

Independent Researcher | ORCID 0009-0004-6295-7040 | gancor@gancor.xyz

July 2026

Keywords operator fingerprinting · strategy identification · trigger geometry · rebalancing microstructure Aerodrome · Uniswap V3 · automated market maker

JEL C58, D82, G14, G23

∴

Abstract. An automated concentrated liquidity manager rebalances on a hidden control policy whose central parameter is a trigger fraction, fixing how close the price may drift to a range edge before the position moves. This paper recovers that fraction from public chain data alone, reading only standard market-maker events, without the operator's cooperation or any knowledge of its contract. It survives in the distribution of the price position at firing, where a proximity trigger leaves a step this paper names the *trigger tell*. Range width, placement convention and correction mode are read from the same events, giving a four-axis fingerprint. Two propositions characterise the recovery. Under stated conditions the trigger fraction is the unique discontinuity of the firing-position density, and jitter convolves that discontinuity into a ramp, bounding what any discontinuity-location estimator can recover at any sample size. A step-ratio estimator attains a measured convergence rate, fixing the sample a target precision requires. The recovery is validated three ways, against a control operator whose fraction of 0.25 is independently known and recovered from scratch, against synthetic operators of planted fraction, and against an independently run manager that disclosed its configuration only after the recovery had been recorded, where the blind value matched to grid resolution. At population scale the law recurs across operators the programme never configured, at per-pool fractions spanning 0.10 to 0.42 that vary as much within an operator as between them. Manager class predicts trigger convention, giving a two-level taxonomy. Out of sample, the recovered fraction anticipates the timing of the control operator's next rebalance. The same construction supplies the countermeasure, so the method that fingerprints an operator also shows how to stop being fingerprinted.

1. Introduction

A production concentrated liquidity manager holds liquidity in a narrow tick range on an automated market maker and moves that range as the price drifts. The decision of when to move is a control policy. On the manager studied by the companion papers,[1,2] the policy is a proximity trigger.[3] The position is rebalanced when the price comes within a configured fraction φ of the range width of either bound. The fraction is a small integer configuration value, hidden in the operator's off-chain code and never emitted on chain. The realised rebalances are public.

This paper asks whether the hidden policy can be recovered from the public record alone. The answer is yes. The trigger fraction of a proximity-triggered manager is a discontinuity in the distribution of the price position at firing, and that distribution is built entirely from the operator's own rebalance and swap events. The recovery uses no manager contract, no application binary interface, no privileged event, and no operator cooperation. It is unilateral in the plain sense that one measures another operator's policy without the operator's participation.

The paper extends a single-operator result into a population-level claim. The companion work recovered φ for one operator,[3] this programme's own manager, whose configuration was known to the programme from its own code and served only to validate the recovery. The open question was portability, whether the same measurement returns sensible values on operators whose configuration is not known from any source but their chain events. It does. Run at population scale through a signature-free index of the rebalancing layer, the proximity firing law recurs across independently discovered operators. The trigger fraction appears to be configured at the pool level rather than once per operator, and the recovered per-pool fractions span 0.10 to 0.42, with the discontinuity structure stable across the recovered proximity class.

The same construction also suggests a defence. What the method recovers is a discontinuity, a sharp step in the firing-position density, and a small random perturbation applied to the trigger before each rebalance smears that step into a ramp with no readable location. An operator that jitters its trigger therefore becomes unreadable by this method while changing rebalancing quality only slightly. The attack and the countermeasure belong together. The relevant audience is twofold, the searcher and security communities, for whom the behaviour of automated actors is an operational input, and the operators themselves, for whom the recoverability of policy is a risk to be managed.

One claim organises the paper. The recovery transfers from the single operator whose configuration was known to a population the author does not run, and the same construction that transfers it also bounds it. The sections develop that claim in turn. Section 4 states the conditions under which the recovery identifies a parameter and the rate at which it converges, Section 5 validates it against a known ground truth, Section 6 reports the population taxonomy it supports, Section 7 tests the out-of-sample anticipation the recovered fraction enables, and Section 9 derives the defence implied by the construction itself. Section 10 records what the method does not yet cover.

2. Related work

No published work known to the author, academic or practitioner, recovers the hidden trigger parameter of a production automated concentrated liquidity manager from public on-chain events, builds a taxonomy of concentrated liquidity vault microstructure, or analyses a jitter defence against such recovery. A dedicated search across the adjacent genres found no direct antecedent, though that cannot rule out unpublished or proprietary tooling. Private searcher-side profiling of automated liquidity providers, if it exists, would sharpen rather than weaken the threat model the defence answers. The paper situates itself by distinction, and four strands are most relevant.

The nearest antecedent identifies an automated strategy from its observed actions. Yang and coauthors classify individual traders and their algorithms from limit-order actions on regulated futures, using an inverse-reinforcement-learning feature space under a Gaussian-process prior.[4] That work requires a privileged exchange audit trail, returns a behavioural cluster label, and has no defence analogue. Wen develops the economics-side premise, that the recoverability of a hidden strategy rises with the frequency of disclosure, which sets up a game between operator and observer.[5] The same inverse problem, recovering a decision rule's hidden parameters from observed optimal actions, is the subject of inverse reinforcement learning in financial domains.[6] The recovery here differs throughout, returning one interpretable scalar in closed form, unilaterally, from public data, paired with its own countermeasure.

The methodological genre for reading a structural parameter off a distributional feature is bunching estimation, reviewed by Kleven, where agents facing a kink or a notch pile up in an observed density whose shape identifies a parameter.[7] Recovering the trigger fraction from a step in the firing-position distribution is the same inferential move, with one mechanical difference. The discontinuity here is imposed by the operator's trigger, not by incentive-induced bunching among optimising agents, so no counterfactual density need be estimated. The step is where trigger-driven firings can and cannot land, not where agents choose to cluster.

DeFi-native work characterises agents without recovering their parameters. Xiong and coauthors measure just-in-time liquidity attacks past the aggregate and down to individual bots.[8] The wider MEV supply chain attributes millions of arbitrages to a concentrated set of searcher entities,[9] groups frontrunning bots into clusters by shared transactions and shared bytecode,[10] and quantifies extractable value in aggregate.[11] None of these recovers an actor's decision parameters. The strongest demonstration that public traces leak enough to reconstruct strategy logic is the real-time imitation of pending victim transactions,[12] which copies execution reactively without recovering any internal parameter. This paper targets the trigger geometry, the parameter governing when a manager acts rather than how it reacts.

A fourth strand recovers identity, not policy. Behavioural regularities in public traces identify the actor behind an address, deanonymising accounts from time-of-day, gas-price and graph quasi-identifiers,[13] through account-clustering heuristics[14] in the lineage of the canonical actor-clustering result of Meiklejohn and co-authors.[15] Misra sets out the practitioner analogue, fingerprinting the software that produced a transaction from its on-chain traits, and observes that anti-fingerprinting measures protect only under broad adoption, a minority-adopted counter becoming a fingerprint itself; that source is not peer reviewed and is flagged as such.[16] The contribution here is to combine these elements into an interpretable recovery of a control-policy parameter from a distributional discontinuity in public data, replicated across operators and paired with a defence.

3. Method

The method has two parts, discovery and fingerprint. Discovery finds the managers in the public record without knowing their contracts. The fingerprint reads four axes of policy from each discovered manager. Both parts use only protocol-standard events, the events defined by the automated market maker and its position layer, never any event defined by a manager. This restriction is what makes the recovery unilateral, and it was learned the hard way. Appendix A collects the notation used below.

The signature route does not generalise, and this is a methodological result, not a gap. An earlier attempt discovered lineage relatives of the control operator by matching its distinctive event signatures, and it worked only because those relatives had kept the operator's exact events. An independent operator whose deployment could not be located by signature, a port of a shared codebase to a different venue with re-typed events, is invisible to exact-signature matching on every chain searched. It is not even findable through the event its own published work is about.[17]

Signature matching finds code relatives, and behavioural discovery on protocol-standard events is what generalises. The lesson set the method's central rule, that both discovery and fingerprint read only protocol-standard events. It is also why an intermediate range-only proxy for the firing price was retracted during the work, since a proxy built from the new range alone manufactures a false discontinuity for a re-centring manager, and the pool tick at firing is required.

3.1. Signature-free discovery

A concentrated liquidity manager on Base does not announce itself. On Aerodrome Slipstream every managed position is minted through a shared position manager, so at the pool mint and burn layer the operator, an off-the-shelf vault, and an ordinary provider all look alike, each carrying the shared manager as sender. The operator identity that survives at the standard-event layer is the staker of the position, and it is recoverable without transaction lookups because the concentrated liquidity gauge stake event carries the staker as an indexed field. One chain-wide scan of that event, grouped by indexed staker, ranks the population by the number of distinct pools each staker spans. A manager typically spans many pools; an ordinary provider spans one or two. No manager code is read.

On Uniswap-v3-style venues,[18] there are no gauges, and the analogue handle is the pool mint event, whose owner and sending account are both public. A manager that holds positions in its own vault appears as the mint owner. A manager that routes through the venue's shared position manager does not, since the owner of every routed position is that position manager; the operator is then the mint transaction sender one level above it. Clustering the mint stream by owner for a selfcustody vault, and by sender otherwise, surfaces the operator population once the position-manager singletons are set aside. Together, the two handles, gauge stake on Aerodrome and pool mint on Uniswap-v3-style venues, cover the main ways a manager holds and drives its liquidity, and each reads only a standard event without reading any manager's contract.

The gauge handle can be scored rather than asserted, by running it as a classifier over its admission threshold against operators verified independently. It holds every verified operator to a ten-pool threshold and admits alongside them a wide margin of ordinary multi-gauge accounts that thins as the threshold rises, so raising the threshold trades recall for purity in the expected direction (Appendix B). A behavioural filter suggests itself and does not work, since deposits per staked position fail to separate verified operators from the rest; whether the staker carries code does separate them. So the handle is generous by construction, admitting a wide candidate set that the fingerprint then sorts, which is the appropriate division of labour when the fingerprint returns diffuse on anything that is not a proximity-triggered manager.

3.2. The fingerprint

For a discovered manager, four axes are read from its rebalances. The trigger axis is the object of interest, the other three are cheaper to measure and often more discriminating.

Trigger geometry. For a range $[\ell, u]$ of tick width $W = u - \ell$, a proximity trigger at fraction φ does not fire when the price exits the range, it fires when the price reaches the trigger line $\ell + \varphi W$ or $u - \varphi W$, a fraction φ of the width inside a bound.[3] For each rebalance let u^* be the position of the price within the *previous* range at the moment of firing, normalised so the bounds sit at 0 and 1 and measured from the pool tick immediately before the rebalance block. A firing with the price already past a bound falls outside $[0, 1]$; those firings are counted separately, as the beyond-range share that identifies an exit convention in Section 6.2, and the folded distribution below is over the in-range firings, for which $u^* \in [0, 1]$. Folding to the distance from the nearer edge, $d = \min(u^*, 1 - u^*)$, a proximity trigger piles firings just inside the trigger line and starves the band just outside it, since the region between the trigger line and the middle is reached only by a firing that the trigger let through. This step in the firing-position density is the operator's trigger tell, the signature its otherwise concealed policy leaves in public data. The recovered φ is the grid point that maximises the ratio of firing mass just inside the line to firing mass just outside, the step-ratio estimator, and a resampling of the firing set gives the sampling band on that estimate. The pool tick at firing is read either from pool state one block before the rebalance, or from the manager's own rebalance event on venues whose managers emit the current tick there.

Range width. The median width of the concentrated ranges the manager mints, in units of the pool tick spacing, separates operators by one to two orders of magnitude on its own and needs no trigger recovery. Near-full-range legs, minted by managers that pair a wide base with a narrow working range, are excluded before the width is taken, since a near-full-range leg otherwise places the firing tick at its edge and manufactures a spurious trigger fraction near zero. Position structure is therefore classified before any trigger recovery, and managers that mint only full-range legs have no trigger geometry to recover.

Placement. With firing tick ft, new range $[\ell', u']$, and tick spacing τ, the placement offset is $z = (ft - (\ell' + u')/2)/\tau$, the position of the firing price relative to the centre of the new range in spacings. A manager that floor-quantises its centre drops the placed centre below the price by a sub-spacing amount, so z sits just above zero, near $+0.5$ in the ideal and attenuated in practice. A manager that centres exactly on the reference price leaves z near zero, and a manager that places its range deliberately above or below the price leaves z a whole number of spacings away. The reference price for a manager that runs a corrective swap in the rebalance is the price after that swap, not the price that triggered the rebalance. Using the pre-swap tick as the reference is a displacement confound, corrected by reading the in-transaction swap tick where one is present.

Correction mode. A rebalance that moves a range changes the token ratio the position requires, and a manager may correct that ratio inside the rebalance transaction or outside it. The axis is read from the transaction, not the log stream. A rebalance counts as corrected in transaction when its transaction contains a swap on the pool being rebalanced and the transaction is attributable to the manager, either sent to it or sent by the address that sends most of its rebalances. Attribution is required, not optional. A router that swaps and then deposits into the same pool places a swap in a transaction the manager never sent, so counting swaps without attribution assigns another party's execution to the operator. The axis also reports what fraction of a manager's rebalances are its own transactions at all, which distinguishes a manager whose rebalances are sent only by its own keeper from one that outside callers may trigger.

3.3. Population-scale recovery

Applied one manager at a time, the fingerprint needs the pool tick at each rebalance, and reading it one rebalance at a time from archive state bounds the recovery to a small sample and makes it sensitive to which window is read. The recovery is instead run at population scale over an index built from bulk event streams alone. For each operator its own mints, filtered by the indexed owner or staker, give the new range at every rebalance across its whole window, and the pool swap stream gives the tick path, from which the firing tick is the last swap strictly before each rebalance block. Both are single indexed-field scans over a public archive, with no per-rebalance state read and no manager application binary interface. The firing-position distribution for a pool is assembled once, and the trigger recovery of Section 3.2 runs offline over the whole population. The index reads only the standard mint and swap events, decoding the two swap-event layouts the Uniswap-v3-style venues on Base use, and it remains signature-free in the same sense as the discovery stage.

4. Identification and its limits

The recovery is a grid search for a discontinuity, so its guarantees are those of a discontinuity-location estimator. This section states the model under which the trigger fraction is identified, shows that it is the unique discontinuity of the firing-position density, the trigger tell of Section 3.2, and reports the step-ratio estimator's convergence rate and the sample size it implies.

Definition 1 (Firing model)**.** A position holds a range $[\ell, u]$ of tick width $W = u - \ell$. For a price at pool tick p, its normalised position within the range and its folded distance to the nearer edge are

$$u^* = \frac{p - \ell}{W} \in [0, 1], \qquad d = \min(u^*,\ 1 - u^*) \in [0, \tfrac{1}{2}]. \tag{1}$$

The operator runs a proximity trigger at fraction $\varphi \in (0, \frac{1}{2})$. Firing is modelled as two competing per-block hazards, each the probability that the position is rebalanced in a given block. A proximity hazard $\eta > 0$ acts only while $d \leq \varphi$, and a scheduled hazard $\lambda \geq 0$ acts at every d. On firing the range re-centres on the price. Write ψ for the occupation density of d over the observation window, the time the price spends per unit of d before a firing.

The scheduled hazard collects everything that fires a rebalance independently of position, width changes, calendar recalibration and manual moves, and it is what a diffuse operator carries in place of a trigger.

Proposition 1 (Identification)**.** *Suppose (A1) ψ is continuous on $(0, \frac{1}{2})$ and strictly positive on a neighbourhood of φ, (A2) φ is constant over the estimation window, and (A3) η and λ are constant over it. Then the firing-position density f on $(0, \frac{1}{2})$ satisfies*

$$f(d) \ \propto\ \psi(d)\,\big(\lambda + \eta\,\mathbf{1}\{d \leq \varphi\}\big), \tag{2}$$

up to the constant that normalises f over $(0, \frac{1}{2})$, so f is continuous at every $d \neq \varphi$ and discontinuous at φ. The trigger fraction is identified as the unique discontinuity location of the firing-position density.

Proof. A firing at position d occurs when the price occupies d and either hazard fires, so the intensity of firings at d is the occupation density times the total hazard active there, which is λ for $d > \varphi$ and $\lambda + \eta$ for $d \leq \varphi$. This gives the displayed form. Away from φ the indicator is locally constant, so f inherits the continuity of ψ that (A1) supplies across $(0, \frac{1}{2})$. At φ the indicator jumps while ψ is continuous and non-zero there, so f is discontinuous at φ and nowhere else. Assumptions (A2) and (A3) are what make a single φ and a single pair of hazards well defined over the window. Hence φ is the unique location at which f is discontinuous. ■

In words, firings land where the price spends its time, weighted by how likely a firing is once the price is there. Both hazards act inside the trigger zone and only the scheduled one acts outside it, so the firing density inherits a step at exactly the trigger fraction, whatever shape the price path gave the occupation density.

Identification is of the discontinuity's location, not its size. The one-sided heights of f at φ would encode the hazard ratio η/λ in the idealised model. Firing latency defeats that, since it lets the price penetrate the trigger zone before the position moves and so shifts firing mass to a depth below φ. The density just inside the line is therefore not raised by the full proximity hazard, and the observed step understates any closed-form ratio. Latency also lets a price that crossed the line recover past it before the tick is read, so the smearing runs both ways, though inward at the scale that matters here.

The estimator is built around this. Rather than estimating the one-sided limits of f at φ, it compares firing mass in a band just inside the trigger line with the mass in the band just outside, taking the band wide enough to contain the penetration mass. The recovered location is then invariant to the scheduled mixture across two orders of magnitude in η/λ, while the step height is not. The step is therefore reported as a per-operator sharpness statistic, monotone in the proximity share of an operator's firing, and never as an estimate of the hazard ratio.

The step-ratio estimator is the grid argmax of that band ratio, a discontinuity-location statistic, and its rate is measured on the synthetic construction of Section 5.1 rather than assumed. Driven across sample sizes from sixty to two thousand firings, thirty replicates at each, the root-mean-square error of the recovered fraction falls as $n^{-0.64}$ on a fine grid, near the root-n rate of a well-behaved location estimator and away from both the cube-root rate of mode estimation and the n^{-1} of likelihood-based change-point estimation on a regular jump. Inverting the fit gives the sample a target precision requires (Table 1), fewer than twenty firings for ± 0.05 and under two hundred for ± 0.01. The recovery grid then binds. On the 0.01 grid the error floors at one grid step by about a thousand firings, past which a finer grid, not more data, sharpens the estimate, and the population inclusion threshold is set by this rule.

Table 1. Firings required for a target precision in the trigger fraction.

Target precision	± 0.05	± 0.02	± 0.01	± 0.005
Firings required	14	60	177	519

The binding assumption is (A2), and its violation is measured rather than assumed away. Configuration churn over the studied venues holds a pool at one fraction for only about half its segments, so the valid window is bounded by the retuning rate and each reported recovery is the pool's longest run of agreeing segments rather than a lifetime constant. Assumption (A1) fails when a single block gaps the price across the whole trigger zone, which smears the jump on very narrow ranges, the same convolution the jitter defence of Section 9 induces deliberately, so the narrow-range limit and the defence are one phenomenon at two scales. Within these conditions the recovery estimates a well-defined parameter, which is what lets a value recovered on one operator be set beside a value recovered on another.

The same model run backwards bounds what the recovery can reach, and the bound is what the defence of Section 9 exploits.

Proposition 2 (Recoverability bound)**.** *Take Definition 1 with the trigger jittered, so that rebalance i fires at $\varphi_i = \varphi + \xi_i$ with the ξ_i independent of the price path, mean zero, of standard deviation σ_J, and drawn from a continuous law with distribution function K_{σ_J}. The firing-position density becomes*

$$f_{\sigma_J}(d) \;\propto\; \psi(d)\,(\lambda + \eta\,(1 - K_{\sigma_J}(d - \varphi))), \tag{3}$$

so the gate $\mathbf{1}\{d \le \varphi\}$ of (2) is replaced by the survival function $1 - K_{\sigma_J}(d - \varphi)$, a monotone transition of width of order σ_J about φ. Hence for every $\sigma_J > 0$ the density f_{σ_J} is continuous and the trigger tell is no longer a discontinuity, so recovering φ degrades from locating a jump to locating a ramp of width σ_J.

Proof. A firing at d under a realised threshold φ_i has intensity $\psi(d)(\lambda + \eta\,\mathbf{1}\{d \le \varphi_i\})$. Averaging over the jitter, $\mathbb{E}\,\mathbf{1}\{d \le \varphi_i\} = \Pr(\varphi_i \ge d) = 1 - K_{\sigma_J}(d - \varphi)$, which gives the displayed density. The averaging factorises because the jitter is drawn independently of the path. A residual dependence of occupation on the realised threshold, since a wider realised trigger truncates deep occupation before it accrues, contributes a term continuous in d, so it changes none of what follows. The survival function of a continuous law of standard deviation σ_J falls from near one to near zero over an interval of width of order σ_J, so the gate's jump is spread over that width and f_{σ_J} is continuous. No discontinuity remains for Proposition 1 to identify. ∎

In words, jitter turns a hard gate into a soft one. The step the recovery searches for is smeared across the width of the jitter, and once it is smeared there is no jump left anywhere to find.

Continuity of the jitter law is not a technicality the operator may ignore. A trigger jittered over a handful of fixed offsets leaves a step at each of them, so the density is again discontinuous and the recovery finds the offsets rather than failing, which makes a smooth perturbation the right way to draw the jitter and a coin toss over two values the wrong one. Nor is the estimator's behaviour part of the proposition. What the proposition removes is the discontinuity; that the step-ratio estimator cannot resolve below the ramp width is a property of that estimator, stated next.

The estimator meets this bound through its comparison band, in two regimes. While the jitter is small against the band the ramp still fits inside it, the step ratio is reduced but keeps an argmax near φ, and a recovery is returned whose value is biased by the smearing. Once the jitter approaches the band the ramp fills it, the mass just inside the trigger line equals the mass just outside, the ratio falls to one, and the read returns the diffuse signature. The two-stage degradation the sweep of Section 9 records, a corrupted value before a collapsed read, is this single transition seen through a finite band rather than a separate fact.

Corollary 1 (Configurable defence)**.** *A jitter of standard deviation of order the comparison band drives the ramp to fill the band and the step ratio to one, so the step-ratio recovery fails irrespective of the adversary's sample size. Since the rate measured above lets a fixed trigger be resolved to order $n^{-0.64}$ in n firings, no quantity of data restores to a discontinuity-location estimator the discontinuity that jitter removes. The operator therefore has a threshold it can set, a jitter a few hundredths of the range width wide, at the cost of rebalancing a like fraction early or late.*

The bound is against this class of estimator, and the honest statement of its scope is the stronger one. The occupation density is itself public, so an adversary who forms the ratio of firings to occupation faces a monotone ramp whose location is estimable in principle from enough data, and the defence is better read as destroying the fast rate than as an impossibility result. What survives that adversary is the part the operator most wants. Each firing draws its own threshold, so even a jitter magnitude and a centre known exactly leave the next firing's trigger uncertain by the jitter, and the anticipation of Section 7 degrades with it. Jitter defends the timing whether or not it defends the parameter.

5. Validation

The method is validated by three tests of ascending independence. The first recovers a fraction this programme itself configured, the second recovers fractions planted by the analyst in synthetic operators, and the third recovers the fraction of an operator run by someone else, before that operator disclosed it. Each answers an objection the one before it leaves open.

The first test uses the one operator whose trigger fraction is known independently, this programme's own manager, referred to throughout as the control operator. Its era value is $\varphi = 0.25$, recovered from on-chain data by the companion work[3] and confirmed against the deployed configuration, which the programme knows first hand. A from-scratch recovery that returns 0.25 here therefore re-derives a settled answer on a settled operator, which tests that the pipeline reproduces a known result end to end. It does not establish independence, both because the operator is a single deployment run by this programme and because the reference value was itself obtained from the firing-position distribution, so the test shares an estimator family with the recovery under test. The independence the population claim needs is supplied by the two tests that follow.

Discovery recovers the control operator signature-free. In a recent window the gauge stake scan grouped by indexed staker surfaces the control operator's manager contract as a multi-pool staker with no manager code read, which is the acceptance test the discovery step is designed to pass. The fingerprint then recovers the trigger fraction. On the era-pinned sample, the 126,227 of the window's 126,339 rebalances that carry a recorded new range, across 48 pools, of which 88,325 carried a measured firing position, the recovered φ is 0.25. The ratio of just-inside to just-outside firing mass peaks sharply at 0.25 and falls away on both sides (Table 2). A resampling of two hundred draws returns the mode 0.25 in every draw. A separate slot-based read on a smaller contiguous sample returns the same 0.25 with the inside-to-outside ratio at 5.7 and the resampling mass on the modal value at 0.96. A mis-specification cross-check confirms the value is not an artefact of the grid. Refitting the joint frequency regression of the companion work at the recovered 0.25 returns its published slopes, a drift slope of 1.27 and a placement slope of 1.06 on volatile–stable pairs and a placement slope of 1.02 on volatile–volatile pairs, where that work records the volatile–volatile drift term as null rather than attenuated. Refitting at the wrong neighbour 0.20 moves both classes off theory, collapsing the drift slope on volatile–stable pairs and pushing placement away from one, so the data rejects the mis-specified trigger.

Table 2. Control operator, inside-to-outside firing mass by candidate trigger fraction, era-pinned sample.

Candidate φ	0.20	0.22	0.24	**0.25**	0.26	0.28	0.30	0.35
Inside/outside ratio	0.85	1.14	1.51	**2.11**	1.83	1.43	1.03	0.95

The discontinuity is visible directly in the firing-position distribution. Folding each in-range firing of the slot-read validation sample to its distance from the nearer range edge and binning gives the density of Figure 1. Firing mass rises across the trigger zone and drops sharply at the trigger line, then recovers into a lower shelf of mid-range recalibration firings that obey no proximity rule. The step sits at $\varphi = 0.25$, the recovered value, and its sharpness is what the inside-to-outside ratio measures. On the era-pinned sample the same step is present at the same location, but with the lower ratio reported above, since the longer window admits more mid-range recalibration mass.

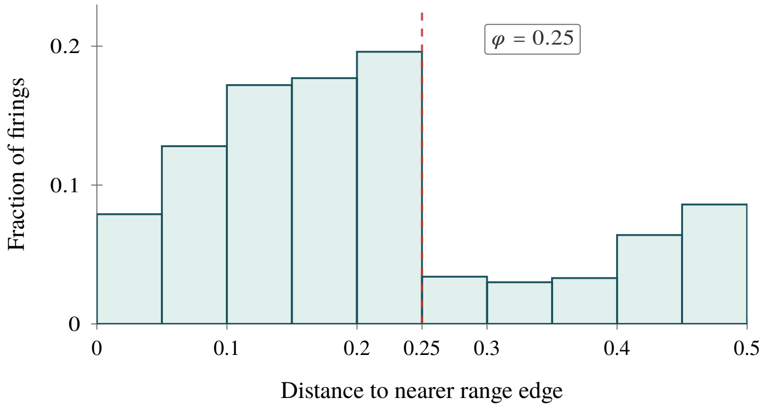

Fig. 1. Firing-position density, control operator, slot-read validation sample, folded to the distance from the nearer range edge. The step at the dashed line is the recovered fraction.

The placement axis validates on the same operator. On a thousand-rebalance sample the mean placement offset is +0.31 with two-thirds of firings above the placed centre, a floor-quantised placement consistent with the operator's known arithmetic. The estimate is attenuated from the ideal +0.5 because the pre-rebalance tick is not exactly the operator's centring reference. The sign and the majority above centre are the features that survive the attenuation. With the trigger and placement recoveries both returning the known answer on the control operator, the method is carried to operators whose answer is not known.

5.1. Synthetic-replay validation

Validating against the control operator shows that the recovery returns the right value on one operator whose configuration is known. It does not, on its own, rule out that the recovery was tuned to that value, especially because the ground truth here is the programme's own manager. The natural question is therefore whether 0.25 was genuinely recovered or effectively assumed. A synthetic replay settles the question without a second real operator. Rebalance sequences are generated from a planted trigger fraction and pushed through the identical recovery pipeline, so the estimator meets data whose answer is fixed in advance and known to the analyst alone. The synthetic operator is not a clean textbook density. It carries a cooldown between rebalances, a firing latency that lets the price penetrate the trigger zone before the position moves, a floor-quantised placement of each new range, and a scheduled-recalibration population that fires mid-range regardless of price. A pass is therefore evidence the recovery survives the confounds that attenuate a real read, not evidence it works only in the ideal.

Across a grid of planted fractions from 0.10 to 0.35 the recovery returns the planted value to within the 0.01 grid step, with a uniform one-step positive offset that reflects the discretisation of the recovery grid rather than a bias of the method (Table 3). Every planted operator classifies as proximity with a sharp step and a concentrated resampling band. Repeating the recovery on eighty independent synthetic operators at a fixed fraction, the point estimate lands within one grid step of the planted value in 88% of runs and within two in every run. An interval formed by widening the resampling band by the grid step contains the planted value in every run; this widening reflects the resolution floor a grid estimator cannot see below. The resampling band on its own concentrates so tightly at production sample size that it understates this resolution floor, which is why the recovered fractions are stated with a grid-resolution caveat rather than as exact constants. Two negative controls confirm the recovery fails safely. A synthetic operator that fires only once the price has left its range returns diffuse with no recovered fraction, and a synthetic operator whose trigger is jittered by a standard deviation above the comparison band returns diffuse rather than a false fraction, the same signature a genuinely scheduled or already-defended operator leaves. The full transition from clean recovery to this diffuse signature is quantified in Section 9.

Table 3. Synthetic-replay recovery across planted trigger fractions.

Planted φ	0.10	0.15	0.20	0.25	0.30	0.35
Recovered φ	0.11	0.16	0.21	0.26	0.31	0.36
Inside/outside ratio	43.9	40.7	37.0	28.3	36.0	47.5
Resampling concentration	1.00	1.00	1.00	0.99	0.99	0.99

5.2. *External validation*

The synthetic replay answers the circularity of a self-check but cannot show the method reads a real operator configured by someone else. One operator discovered independently, and not run by this programme, agreed to disclose its configured trigger fraction after the fact. The recovery was performed first and written to an artefact anchored at Base block 49,241,569, and the disclosure was requested only once that value was recorded, so the comparison is blind. From public chain data alone, over 225 firing positions across the operator's three pools, the recovery returned a trigger fraction of 0.27, or 0.265 on the finer grid, and the operator then disclosed a configured band of a quarter of the range, $\varphi = 0.25$. The recovered and configured values are one to two steps of the recovery grid apart, inside the envelope the synthetic replay characterises, on an operator whose configuration was unknown until after the recovery was fixed. The block anchor is what makes the claim checkable, since the recovery reads only chain data to that height and an independent party can repeat it there. That the recording preceded the disclosure rests on the author's own frozen record rather than on a public commitment, and is stated as such.

The disclosure also exposed a structure the control operator does not have. This operator interleaves fee-compounding re-mints, which re-open the position at the same centre while the price sits mid-range, with the genuine band-triggered relocations, so a recovery that counts every rebalance mixes non-trigger events into the firing distribution. Restricting to the transactions that relocate the range, the firings concentrate at a median distance of 0.248 from the range edge, reproducing the disclosed quarter-range band, with none in the middle half of the range, which is where the operator states its trigger never fires. The recovery is therefore validated against an operator this programme does not run. That is a single operator, and one blind agreement is weak evidence on its own; what it establishes is that the method can return a correct value on a configuration nobody in this programme chose, which is the step the control and the synthetic replay cannot supply. The fee compounding this operator interleaves with its triggers is also a confound a population recovery must separate from genuine relocations.

6. Results

Validated on the control, against planted fractions and against one operator it does not run, the method is now carried where no ground truth exists. The population studied is the set of concentrated liquidity managers discovered signature-free on Base across two venues, Aerodrome Slipstream through the gauge handle and Uniswap-v3-style venues through the mint handle. On the Uniswap-v3-style venues the operator is the account that drives the rebalance, the mint transaction sender, since the position itself is held by the venue's shared position manager. The recovery keys on that sender once the position-manager singletons are set aside. The four fingerprint axes of Section 3.2 are reported in turn. The trigger axis carries the central result, that the proximity firing law recurs across independently discovered operators at fractions configured per pool. The width axis is reported with the trigger convention it tracks, since the two are views of one choice and together give a two-level taxonomy in which manager class predicts convention. Placement and correction mode follow, each separating operators that share a trigger convention. Every operator other than the control is anonymised, which is this work's settled policy on naming third parties, and the recovered values are keyed to a private table held outside this paper.

6.1. The trigger axis, replicated across operators

The central result is that the proximity trigger recurs across operators the programme has never seen, and that the trigger fraction is a per-pool configured quantity rather than one value per operator. A signature-free index over the discovered population (Section 3), keyed on the mint transaction sender above the venue position managers, recovers the firing-position discontinuity for every pool an operator rebalances. Across a recent window on the studied pools, eighty-one distinct operators imprint a clean proximity discontinuity, and the recovered per-pool trigger fractions form a distribution spanning 0.10 to 0.42, binned to the nearest 0.05 in Table 4 so that the highest recoveries sit in its 0.40 column. One hundred and two operator-pool cells carry a clean step, a cell being one operator paired with one pool it rebalances, and the same law holds in each, a proximity trigger a fraction inside the range, with only the fraction moving. Nine of the operators are multi-pool.

The spread is not an artefact of pooled data. Those hundred and two cells fall across nineteen distinct pools, so a pool is often rebalanced by several of the discovered operators at once and their recoveries are read from a shared tick stream. Keeping only the best-sampled cell in each pool leaves nineteen cells on nineteen operators with no tick stream in common, and the recovered fractions still run from 0.13 to 0.42 with a median of 0.27, so the population is broad whether or not the dependent cells are counted.[1]

Two recoveries check the key rather than the fraction. The control operator's keeper is recovered at its own address, returning fractions of 0.26 and 0.27 on its two pools against a configuration known to be 0.25, which is the cross-check that the recovered entity is the operator and not the shared position manager it mints through. A second operator whose custody architecture is known independently is recovered at the right address and the right custody class, but imprints no clean discontinuity on any pool, so it contributes no fraction.

The fraction also varies over time. On an earlier pass over the same venues, cutting each pool's history into equal firing segments and recovering the fraction segment by segment showed the median pool holding one value for only half its segments, so a per-pool fraction is a summary of a configuration that is retuned. Each reported recovery is therefore the pool's longest run of agreeing segments, a stable read rather than a mixture, and the retuning itself is part of the fingerprint.

Two features of the distribution's edges matter for how it is read. Some recoveries are exceptionally sharp, one pool piling over a hundred times as much firing mass just inside its trigger line as just outside, so the recovered fraction there is a genuinely configured value and not a statistical accident. The few recoveries at the top of the candidate grid carry strongly one-sided firing counts and read as convention results, not trigger fractions. This is the population form of the cross-operator claim first stated on one operator, and while the distribution is broad it carries a soft concentration near a quarter that sharpens once the coarsely resolved narrow-range cells are set aside, rather than a single dominant value.[3]

The trigger fraction varies as much within an operator as between operators. The most active multi-pool operator's pools span 0.14 to 0.37 on their own, so the width-trigger interaction seen across the population is visible inside a single operator as well. A single fraction per operator is therefore a summary of a per-pool distribution, not a policy constant. The pool remains the unit the policy is set on, and the distribution of Table 4 is the object the paper reports.

[1]The independence check and the data-quality sweeps of Appendix D are computed on the population gate that admits one further cell, giving 103 against the 102 reported here; the difference is one thinly sampled cell and no claim turns on it.

Table 4. Recovered per-pool trigger fractions, discovered operator population.

Trigger fraction φ	0.10	0.15	0.20	0.25	0.30	0.35	0.40
Pools recovered	11	16	19	17	16	15	8

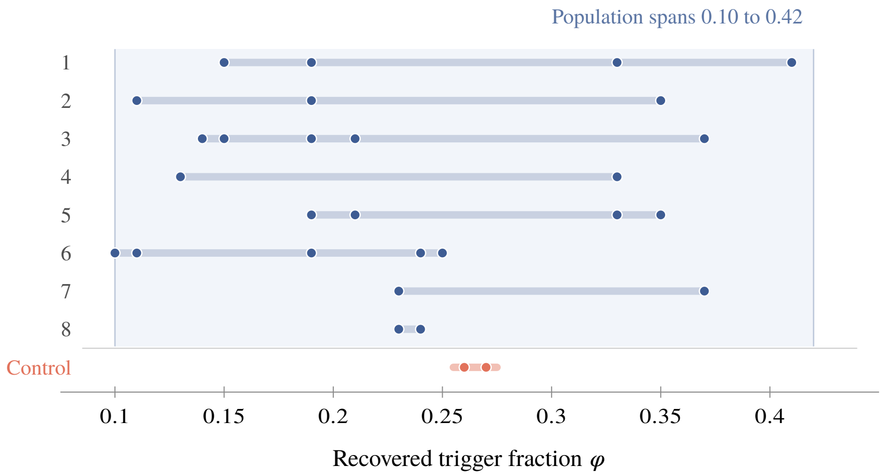

Fig. 2. Per-pool trigger fractions within each multi-pool operator, one row per operator, ordered by spread. Dots are that operator's pools and the bar spans its range. The control operator's keeper is the bottom row.

Figure 3 shows the distribution against the control, whose own pools, read with its execution record over the full era, concentrate at the configured default the validation recovered. The control panel is the thirty-five of its pools carrying a per-pool slope fit, a narrower filter than the forty-eight that read a fraction. The spread is the finding, and the control's spike is what one operator's policy looks like inside it.

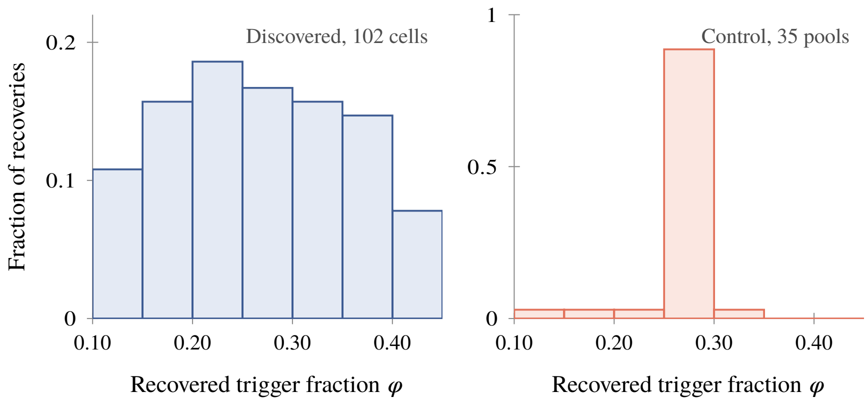

Fig. 3. Recovered trigger fractions. Left: the discovered population. Right: the control operator. Note the differing vertical scales.

The replication is visible when the firing-position densities are overlaid. Figure 4 places the control operator against a discovered operator, fingerprinted with no cooperation and no manager code, whose recovered fraction is 0.13 on the pool shown, read over eleven thousand firings. Each density has the same shape, mass through the trigger zone and a step down at the trigger line, and the step sits at a different place for each, at 0.13 for the discovered operator and 0.25 for the control. The step location is the per-pool trigger fraction, and its movement between operators is the recovery made visible.

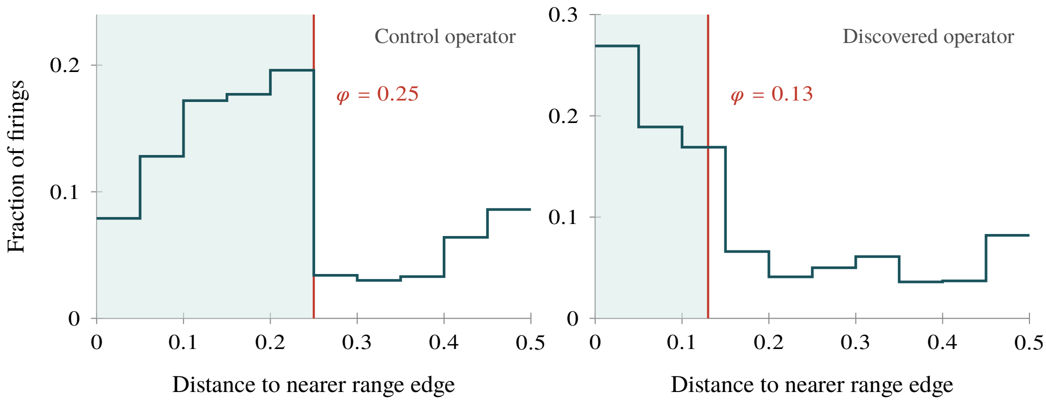

Fig. 4. Firing-position density folded to the distance from the nearer range edge. Left: the control operator. Right: a discovered operator. Each operator's mass runs through its own trigger zone (shaded) and steps down at its trigger line.

Each per-pool fraction is a point estimate that moves with the window and the sample size. The stable finding for a well-sampled pool is the convention, proximity against exit or schedule, while the fraction itself settles only at a large firing count, so Table 4 is over pools sampled well enough to resolve a clean step. Section 10 bounds the digits, the coverage and the resolution.

The firing law's deeper form, not only its trigger geometry, is reproduced on the control against a calibrated reference. On the control operator, with the tick stream taken from the operator's own execution record over the full era, the recovery reproduces the slope-one placement law of the companion work, the log ratio of up-move to down-move firings fitted against the drift and placement regressors at the recovered trigger lines.[3] The control's pools enter these fits under different filters, forty-eight carrying enough firings to read a fraction, thirty-nine surviving into the joint regression and thirty identified well enough for an individual slope. The placement slope, fitted pool by pool at each pool's own trigger, forms a distribution over those thirty well-identified pools with precision-weighted mean 1.00 and median 1.03, and the joint regression across thirty-nine pools returns a placement slope of 0.92 with an intercept indistinguishable from zero. Where the tick data is the operator's own, the law is quantitative, and the wide but centred pool-level scatter is overdispersion around a reproduced slope of one rather than a drifting estimate.

The trigger geometry transfers broadly, recovered across the eighty-one operators above, but the deeper first-passage slope law is a stronger claim that this paper pins quantitatively only on the control. A per-operator fit on public data would have to be keyed on the mint sender, since a fit on the aggregate flow through a shared position manager mixes operators rather than measuring one, and that regression is left for later work. What the public pipeline does to a slope is nonetheless measurable on the control. It inflates the precision-weighted slope to 1.9 while leaving the median pool at 0.95, so the bias concentrates in that operator's most frequently rebalanced pools, and across the control's own pools neither a pool's firing rate nor its trading intensity accounts for the spread. The slope also moves with the window even in privileged data, from 0.92 over the full era to about 1.3 over a short one, where the drift term cannot be estimated pool by pool and market drift confounds with placement. What transfers from public data alone is therefore the law's form, an asymmetric first-passage structure with a placement slope of order one and a zero intercept on the control, while the slope itself is pinned only by the operator's own execution data.

6.2. The width axis and a two-level taxonomy

Trigger convention is not uniform across the population, and its variation is not noise, it tracks how the manager holds and moves its liquidity. The managers discovered through the mint handle on Uniswap-v3-style venues are the ones that trigger on proximity at a clean fraction (Section 6.1). The gauge-staking managers on Aerodrome are, in the characterised sample, mostly something else.

Some fire only after the price has left the range, an exit convention, and some fire mostly mid-range on a schedule or another signal with no readable proximity step (Table 5, which reports for each the share of its firings landing beyond the range). None of the characterised gauge-staking managers triggers on a clean proximity fraction the way the control operator does. The control operator is one point in the population, not its representative.

Table 5. Trigger convention against range width, characterised gauge-staking population, Aerodrome Slipstream.

Operator	Width (sp)	Beyond-range (%)	Convention
A	3	81	Exit
B	3	71	Exit
C	1	83	Exit
D	1	16	Diffuse
E	2	32	Diffuse
F	7	0	Diffuse
G	1	43	Diffuse

Width policy correlates with convention across the population. The exit-triggered managers run tight ranges of one to a few spacings, and a tight range is exited often, so an exit rule is the natural convention for it. Across the wider census the diffuse managers include the widest books, in which the price stays inside the range far longer, so they rebalance on schedule or on another signal rather than on the price reaching an edge. Among the well-sampled managers reported here the two width distributions overlap, so the association is a tendency across the population rather than a separation visible in every row of Table 5. Two of the fingerprint axes, width and convention, are two views of the same underlying choice. The two-level structure is therefore class first, then parameter, the manager class predicts the convention and, within the proximity class, the operator-specific fraction distinguishes operators.

6.3. The placement axis

Placement separates operators that share a trigger convention. After correcting for the displacement confound by referencing the post-swap price, the placement offset splits three ways across the characterised population. Most discovered managers centre their new range on the reference price, with an offset near zero and no visible quantisation bias. The control operator floor-quantises, sitting at an offset near +0.31, a fraction of a spacing above centre and distinct from the exact-centring majority. A small number place their range about two spacings below the price, with more of the range's coverage sitting under the price than over it. That is a directional placement, not a rounding artefact. Floor arithmetic in *placement* is therefore an operator-specific choice rather than a universal one, which confirms across operators what the companion work argued from one.[19] The distinct question of whether an operator also floors the price at which it *sizes* its corrective swap, which is the arithmetic that work's own law turns on, is not read by these axes and remains open. The full fingerprint per operator is width, trigger convention with its fraction where proximity applies, placement offset, and the correction mode of Section 6.4. The control operator is distinct on all four axes where each is measurable.

6.4. The correction-mode axis

A manager that moves a range must also fix the token ratio the move implies, and where it does so is visible in the same events. The measurement samples rebalance transactions per operator and reads each transaction's receipt. A rebalance counts as corrected in transaction only if it carries a swap on the pool being rebalanced and that transaction is attributable to the operator, either because it is sent to the manager or because it is sent by the operator's own keeper. The control operator and the discovered operators go through the same procedure, with the control's transactions taken from its own rebalance record and classified from public receipts like the rest.

The population divides. The control corrects in transaction on about two thirds of its sampled rebalances, and three of the discovered operators correct at comparable rates, from 0.60 to 0.75, with one further operator correcting on a tenth of its sampled rebalances and the remainder on none. Roughly two in five sampled operators correct in transaction. Attribution is not optional, since a router that swaps and then deposits into the same pool leaves a swap in a transaction the manager never sent, and the apparent same-pool swaps that fail attribution testing all belong to third-party routers.

The same receipt read also shows who may call a rebalance. Two operators take a material share of their rebalances from external callers rather than from their own keeper, and the receipt shows those callers paid in the same transaction, an open and incentivised rebalance surface recovered without reading the contract. Others send only their own rebalances. The permission surface is a fingerprint axis in its own right.

The consequence reaches past the taxonomy. The dust mechanisms the companion work identifies, a target ratio computed at a quantised price racing the price impact of the corrective swap, presuppose that the correction happens in the same transaction.[19] An operator that does not correct in transaction has no such race, so those laws do not describe it. How the deployed population divides on this point is the scope question the companion work leaves open, and the sample here answers it, with roughly two in five sampled operators correcting in transaction and the remainder not.

For the operators that do correct, one further quantity of that framework becomes readable from outside. The companion work gives the corrective swap's tick displacement as a factor of trigger geometry times the operator's footprint ratio, and names that ratio the quantity to measure per manager.[19] Both the displacement and the geometry factor are public, so the relation inverts, and the ratio follows from the operator's own corrective swaps rather than from position values only the operator holds. Applied to one discovered operator it returns a ratio of order 10^{-4}, which is the first such measurement of a manager this programme did not configure. The sample is thin rather than the method unsound, since these operators run narrow ranges, and a narrow range leaves a geometry factor small enough that most events divide by a quantity consistent with zero. The inversion is therefore reported as a demonstration that the footprint is recoverable without cooperation, not as a distribution over the population.

6.5. Coverage and scope

Coverage is reported as a result, not hidden as a shortfall. Among the managers found through the mint handle, every one sampled well enough to trust the recovery triggers on proximity (Section 6.1), and among the gauge-staking managers on Aerodrome the characterised set splits into exit and diffuse conventions with none proximity (Section 6.2). Operators too thinly sampled in the window are not classified, a limit Section 10 states precisely, and a handful of gauge stakers staked too little to fingerprint at all. An operator with no readable discontinuity is a taxonomy result, exit-triggered, scheduled or already jittered, not a recovery failure quietly dropped. The method reads convention reliably wherever an operator is well sampled, and recovers a clean trigger fraction for the proximity class.

The study is on Base, but the recovery is not tied to it. A probe of the largest Uniswap V3 pools on Ethereum mainnet, keyed the same signature-free way on the mint sender, finds the same split, the most active direct-minting manager imprinting a clean proximity discontinuity while the wide-range managers read diffuse and fire mid-range, so the tight-proximity against wide-diffuse taxonomy is not an artefact of one chain. That transfer also bounds the exposure honestly, since proximity triggering is a tight-range behaviour and the wide-range managers that carry the larger flow are the diffuse ones, so a larger chain does not imply a larger recoverable stake. A full cross-chain treatment, and a large-notional proximity operator should one exist, are left for later work.

7. Anticipating the next rebalance

A recovered trigger fraction is a threat only if it predicts behaviour the operator has not yet exhibited, so the test is out of sample. The fraction is recovered on the first half of the control operator's history, the recovered boundary is carried forward onto the held-out half, and the operator's own tick path is then walked block by block, each moment labelled by whether the price sits inside the recovered trigger band or away from it. The question is whether a rebalance follows sooner from inside the band than from outside.

It does, by a wide margin at short horizons. Across the control operator's pools, a rebalance within the next ten blocks, about twenty seconds on this chain, is roughly sixteen times more likely when the price is inside the recovered boundary than when it is outside, against a position-blind baseline of one (Table 6). The ratio falls smoothly as the horizon lengthens, towards two over a thousand blocks, because a position far from its trigger eventually reaches it, so the discrimination is a short-horizon one and the table records its decay rather than a single figure.

The effect is a position signal and not a restatement of elapsed time. On the four pools carrying enough held-out history for the comparison, a model that ranks moments only by how long the position has held its range is worse than a coin toss at this task, since a long-held range is a quiet one, whereas position sorts the moments at an area under the curve of 0.76. Conditioning on holding time does not remove the effect, the inside-against-outside ratio staying above one within every decile of it. A calibration of the realised hazard against position locates the rise at the recovered fraction, the hazard climbing to a peak just inside the trigger line and collapsing across it, the same discontinuity the recovery reads, now expressed as a rate.

Table 6. Out-of-sample hazard ratio by horizon, control operator held-out history, nine pools.

Horizon N (blocks)	10	40	160	640	1,280
Hazard ratio	16.3	12.2	6.4	2.7	1.85
95% interval	14.8–18.0	11.0–13.6	5.8–7.1	2.4–2.9	1.7–2.1

Two limits are stated plainly. The demonstration is on the control operator, whose configuration the programme knows, so the recovery is genuinely out of sample while the subject is not independent, the same single-ground-truth boundary recorded for the population recovery. And the capability is short-horizon by construction, sharp over seconds to minutes and modest over hours. Within those limits the paper exhibits the capability its defence answers rather than asserting it.

The demonstration is one operator, but the exposure is the population. Every proximity manager the recovery reaches, the eighty-one operators across one hundred and two operator-pool cells of Section 6.1, rebalances on a schedule its trigger tell makes anticipatable, some fourteen hundred rebalances a day averaged across the population over the study window. The stake is not one manager's next move but a standing, population-wide stream of position changes whose timing is public, and it is the scale of that exposure, not any valuation of what an adversary would extract from it, that the defence answers.

8. Ethics and disclosure

This work reads, and now anticipates, the behaviour of production operators who did not consent to being studied and cannot opt out, and who are pseudonymous, so the private notice and embargo customary for a vulnerability have no party to address. The posture adopted in place of coordinated disclosure is to publish the countermeasure with the method. The trigger jitter of Section 9 defeats the recovery at a cost to the operator that is negligible against becoming unreadable, so an operator that reads this paper can defend against it the same day, and where the affected parties cannot be reached an attack published with a sufficient and inexpensive remedy is the responsible substitute.

The anticipation of Section 7 is a stronger disclosure than reading configuration alone, and the paper bounds it deliberately, demonstrating it only on an operator this programme configures. It reports the exposure as the scale of anticipatable activity, but publishes neither a strategy conditioned on the prediction nor a valuation of the profit an adversary would extract, either of which would operationalise the capability against operators who have not consented. Every other operator is anonymised against a private table held outside the paper, and the external validation of Section 5.2 used an operator that was informed and disclosed its configuration voluntarily after the recovery was recorded.

9. Defence

The recovery has a single point of failure that the operator controls, and Proposition 2 locates it. What the method reads is a discontinuity, and jitter on the trigger removes the discontinuity, so the resampling concentration that certifies a clean recovery collapses and the fingerprint returns diffuse, indistinguishable from a genuinely scheduled manager. That much is settled by the bound. What remains is operational, how much jitter is needed, what it costs the operator to run, and what an analyst sees when it has been applied. The answers are that the countermeasure is cheap and that its failure mode is silent. A rebalance a fraction of a range width early or late is a negligible change in position quality against the value of becoming unreadable.

The countermeasure can be sized without reference to any deployment, because every quantity involved is a fraction of the range width. A sweep of the jitter standard deviation against the exact recovery pipeline, at the sample sizes of the synthetic replay, measures how the recovery degrades (Figure 5). The degradation has two stages, and the value fails before the read does. At a standard deviation of one to four grid steps the recovery still returns proximity classifications whose certificates look clean, but the recovered fraction rides the upper edge of the jittered band, biased upward by roughly twice the standard deviation. At two grid steps nine in ten reads land more than two grid steps from the planted value. Once the standard deviation reaches the width of the comparison band the resampling concentration collapses and the read rate falls towards the noise floor, the diffuse signature of Section 5.1.

A jitter of a few hundredths of the range width is operationally negligible, so an operator can price the defence at either stage, an untrustworthy value at two grid steps and a read that fails on most attempts at five.

The failure is silent in the middle regime, and this cuts both ways. An analyst running the recovery is not warned that the number is wrong, and a defended operator should not expect the attack to report failure. The upward bias is systematic but cannot be inverted without knowing the standard deviation, which is the one parameter a defended operator never publishes. The curve keeps its shape at a different planted fraction, so it prices the mechanism rather than one configuration.

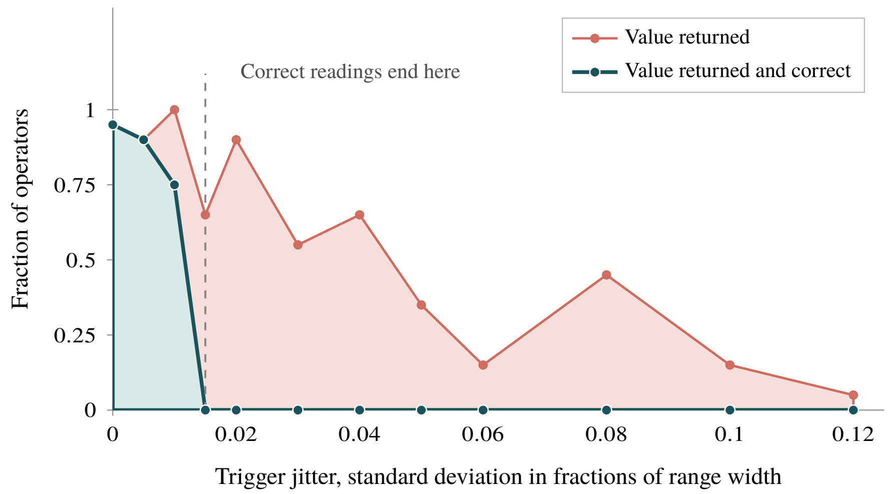

Fig. 5. Jitter against recoverability, twenty synthetic operators per magnitude at planted $\varphi = 0.25$. The gap between the two curves is the silent-failure region, where a value is still returned and is wrong.

The defence is presented generically, and the paper makes no claim that rests on a particular deployment. What the paper claims is the mechanism, that jitter smears the recoverable discontinuity, and the mechanism follows from the construction of the recovery itself. It is demonstrated in the controlled setting of the synthetic replay, where a synthetic operator whose trigger is jittered by a standard deviation above the comparison band returns diffuse rather than a false fraction (Section 5.1). The fixed-trigger step it would smear is exactly the control operator's validated recovery of Section 5. The defence is also the responsible-disclosure anchor of the work. A method that reads an operator's policy from public data is disclosed together with the countermeasure that neutralises it, so the same paper that shows an operator is fingerprintable shows the operator how to stop being one.

10. Limitations

The trigger fraction is per-pool and sample-sensitive, and its digits should not be over-read. The convention is stable for a well-sampled pool, but the fraction stabilises only with a large firing count, and a short window or a thinly used pool can move it by several hundredths, and at small firing counts can even misread proximity as diffuse. The resolution is bounded by range width as well as by sample. About half the clean cells run ranges of one or two tick spacings, on which the folded firing position takes only a few values, so their fractions are coarse and the low bins of the distribution should be read at that resolution rather than as a dense population of low-fraction triggers. The distribution reported here is over pools sampled well enough to resolve a clean discontinuity. Thinly sampled pools are excluded rather than reported at a soft fraction.

The placement axis is confounded where a manager fires far from its new range, an exit-triggered manager most of all, because the offset then measures the displacement between the trigger and the eventual placement rather than the quantisation. The confound is corrected by referencing the post-swap price where an in-transaction swap is present, and the corrected offsets are the ones reported. A manager with no readable reference price is left with a placement that mixes quantisation and displacement, and is not reported.

Coverage is partial in three specific ways. First, the recovery reaches gauge-staking managers on Aerodrome and, through the mint handle, managers on Uniswap-v3-style venues, and a manager that neither stakes in a gauge nor appears in the mint stream, or that operates on a venue outside these two classes, is not discovered by either handle.

Second, on the Uniswap-v3-style venues, the per-rebalance state read that once bounded the recovery is replaced by the population index of Section 3.3, which reads the firing tick and the new range from bulk event streams with no per-rebalance archive call. The swap-event layout, which once excluded two venues outright, turned out to be a shallow gap. The venues studied here diverge in the identifier and the trailing fields of their swap event but agree on where the tick sits in the payload, so admitting a venue is a table entry, not a decoder. The index now ticks all but about two rebalances in a thousand. A venue that moved the tick, rather than renaming the event around it, would still need work.

The population is also conditioned on the pools it was searched over, which were selected by an earlier pass keyed on the position owner rather than on the sender, so the operators reported here are those active on that pool set rather than a census of the venues.

Third, single-pool vault families, which appear as separate owners under the mint handle, are assembled by grouping on the shared mint sender, which lifts the handle's recall from a small fraction of the verified operators to nearly all of them. A shared sender alone cannot tell a platform keeper running many vaults from an automation service acting for unrelated customers, so every assembled family is put to a second and independent test on its members' deployed code, which resolves the large components and leaves a residue of small ones unattributed (Appendix C).

The recovery is insensitive to its data-quality guards, on the control and across the discovered population alike, and the sweeps that establish this are reported in Appendix D.

The firing law's slope is not pinned quantitatively from public data, and Section 6.1 measures the degradation. Its cause is that the signature-free firing tick, taken from the last public swap, need not be the price the operator acted on, so the slope reads high. The bias is established but not modelled, and cannot be corrected by extrapolation. The law's structure transfers but its exact slope does not, which is a boundary of unilateral recovery rather than a data-quality defect.

The recovery identifies a policy parameter, and using it as a point forecast of where the next rebalance will land fails, since the trigger fraction locates a boundary, not a typical firing position and firings spread inside the trigger zone rather than on its edge. What the fraction does support is a timing prediction, demonstrated in Section 7. That demonstration is on the control operator alone, so whether the anticipation transfers to operators the programme does not run is not established here, and the effect is short-horizon, decaying as every position eventually reaches its trigger.

Finally, the control operator is a single deployment run by this programme, so the validation against it is a self-check against a known configuration rather than an independent audit. The synthetic replay of Section 5.1 answers the circularity this raises, and the external validation of Section 5.2 adds the second real operator the self-check lacked, recovering that operator's configured fraction from public data before the operator disclosed it. That external check is a single operator, so the cross-operator replication still rests on a wider population whose ground truth is not available, interpreted as the control operator was after the control, synthetic and external checks showed the method behaving correctly. The claim is falsifiable in an obvious way. A second operator that disclosed a configured fraction materially different from the value recovered for it, or a population recovery that returned sharp fractions from operators later shown to fire on a schedule, would refute it, and the recovery is published in enough detail for either test to be run by someone else.

11. Data and code availability

Every quantity reported here is derived from public chain data. The rebalance and swap streams are standard events on Aerodrome Slipstream and the Uniswap-v3-style venues studied, readable from any Base archive node or indexer, and the recovery reads nothing else. The control operator's privileged execution record, used only where the text says so, is the one input a third party cannot reconstruct; every population result, the taxonomy, the anticipation test and the jitter sweep depend on public data alone.

The recovery is specified in enough detail to reimplement, and the synthetic construction of Section 5.1 is the natural first test of any reimplementation, since it plants a fraction the implementer chooses and the pipeline should return it to within the grid step. A reproduction harness and the sanitised artefacts behind the tables and figures are at https://github.com/g4nc0r/operator-microstructure, anonymised as Section 8 records. The per-operator identifications behind those pseudonyms are held with the private operator key and are not distributed.

12. Conclusion

The hidden rebalancing policy of a production concentrated liquidity manager is recoverable from public chain data alone, without cooperation or contract knowledge. The trigger fraction is a discontinuity in the firing-position distribution, the trigger tell. Range width and placement convention are read from the same events, and the whole fingerprint uses only the events the automated market maker and its position layer define.

The recovery is validated three ways, against a control operator whose fraction of 0.25 was settled before this paper began, against synthetic operators of planted fraction, and against one independently run manager that disclosed its configuration only after the blind recovery was recorded. Run at population scale over the discovered operators, it confirms the programme's proximity firing law across operators it did not configure. The trigger fraction is a per-pool configured quantity, and the recovered fractions form a distribution from 0.10 to 0.42 that varies as much within an operator as between them, while the law's form is invariant. Its deeper structure, an asymmetric first-passage relation with a placement slope of order one and a zero intercept, is reproduced on the control from that operator's own execution record. What transfers across operators from public data is the trigger geometry, not the exact slope.

Three results follow from the population. It organises into a two-level taxonomy in which manager class predicts trigger convention. The recovered fraction anticipates the timing of the control operator's next rebalance out of sample. The same construction also supplies its own defence, since jitter on the trigger smears the recovered discontinuity, corrupting the recovered value at two grid steps and driving the read itself towards the noise floor as the jitter reaches and passes the comparison band. The recoverability bound derives that degradation as the trigger tell convolved with the jitter, which turns it into a threshold the operator can set and leaves the next firing's trigger uncertain even where the jitter is known. That transfer, from one validated operator to a population the programme does not run, is the paper's main result, and the bound that limits it is the same construction read backwards.

References

[1] Ryan, K.R. "The Geometric Siphon: Existence, Equilibrium, and Directional Properties of the Residual in Concentrated Liquidity Portfolios." *SSRN preprint 6686798* (2026) https://ssrn.com/abstract=6686798.

[2] Ryan, K.R. "The Hidden Microstructure of Shared Balance Concentrated Liquidity: A Master Equation for the Dust Ledger and Propagation of Chaos." *SSRN preprint 6745218* (May 2026, revised July 2026) https://ssrn.com/abstract=6745218.

[3] Ryan, K.R. "Operator Microstructure: Decomposing the Directional Asymmetry of Concentrated Liquidity Dust Flows." *SSRN preprint 7164459* (2026) https://ssrn.com/abstract=7164459.

[4] Yang, S.Y., Qiao, Q., Beling, P.A., Scherer, W.T., Kirilenko, A.A. "Gaussian Process-Based Algorithmic Trading Strategy Identification." *Quantitative Finance* **15.10** 1683–1703 (2015) https://doi.org/10.1080/14697688.2015.1011684.

[5] Wen, X. "The New Transparency Trap: Machine Learning and the Reverse-Engineering of Strategies." *Working paper, Hong Kong University of Science and Technology* (January 2026) https://www.xudongwen.com/assets/files/JMP.pdf.

[6] Roa-Vicens, J., Chtourou, C., Filos, A., Rul·lan, F., Gal, Y., Silva, R. "Towards Inverse Reinforcement Learning for Limit Order Book Dynamics." *arXiv:1906.04813* (2019) https://arxiv.org/abs/1906.04813.

[7] Kleven, H.J. "Bunching." *Annual Review of Economics* **8** 435–464 (2016) https://doi.org/10.1146/annurev-economics-080315-015234.

[8] Xiong, X., Wang, Z., Knottenbelt, W., Huth, M. "Demystifying Just-in-Time (JIT) Liquidity Attacks on Uniswap V3." *Cryptology ePrint 2023/973; IEEE BRAINS 2023* (2023) https://eprint.iacr.org/2023/973.

[9] Wu, F., Sui, D., Thiery, T., Pai, M. "Measuring CEX-DEX Extracted Value and Searcher Profitability: The Darkest of the MEV Dark Forest." In *7th Conference on Advances in Financial Technologies (AFT 2025)*, Leibniz International Proceedings in Informatics **354**, pp. 26:1–26:23 (2025) https://doi.org/10.4230/LIPIcs.AFT.2025.26.

[10] Torres, C.F., Camino, R., State, R. "Frontrunner Jones and the Raiders of the Dark Forest: An Empirical Study of Frontrunning on the Ethereum Blockchain." In *Proceedings of the 30th USENIX Security Symposium*, pp. 1343–1359 (2021) https://arxiv.org/abs/2102.03347.

[11] Qin, K., Zhou, L., Gervais, A. "Quantifying Blockchain Extractable Value: How Dark is the Forest?" In *2022 IEEE Symposium on Security and Privacy*, pp. 198–214 (2022) https://arxiv.org/abs/2101.05511.

[12] Qin, K., Chaliasos, S., Zhou, L., Livshits, B., Song, D., Gervais, A. "The Blockchain Imitation Game." In *Proceedings of the 32nd USENIX Security Symposium*, pp. 3961–3978 (2023) https://arxiv.org/abs/2303.17877.

[13] Béres, F., Seres, I.A., Benczúr, A.A., Quintyne-Collins, M. "Blockchain is Watching You: Profiling and Deanonymizing Ethereum Users." In *2021 IEEE International Conference on Decentralized Applications and Infrastructures (DAPPS)*, pp. 69–78 (2021) https://arxiv.org/abs/2005.14051.

[14] Victor, F. "Address Clustering Heuristics for Ethereum." In *Financial Cryptography and Data Security 2020*, Lecture Notes in Computer Science **12059**, pp. 617–633 (2020) https://doi.org/10.1007/978-3-030-51280-4_33.

[15] Meiklejohn, S., Pomarole, M., Jordan, G., Levchenko, K., McCoy, D., Voelker, G.M., Savage, S. "A Fistful of Bitcoins: Characterizing Payments Among Men with No Names." In *Proceedings of the 2013 ACM Internet Measurement Conference*, pp. 127–140 (2013) https://doi.org/10.1145/2504730.2504747.

[16] Misra, I. "Wallet Fingerprints: Detection & Analysis." Practitioner article (2023) https://ishaana.com/blog/wallet_fingerprinting/.

[17] P., I. "Convexity and Topology in Cross-Pool Dust Redistribution: Structural Edge in Active CL Management." *SSRN preprint 6509938* (2026) https://ssrn.com/abstract=6509938.

[18] Adams, H., Zinsmeister, N., Salem, M., Keefer, R., Robinson, D. "Uniswap v3 Core." *Uniswap Labs Technical Whitepaper* (2021) https://uniswap.org/whitepaper-v3.pdf.

[19] Ryan, K.R. "Quantisation Microstructure: The Tick-Race Law of Rebalancing Dust." *SSRN preprint 7166739* (2026) https://ssrn.com/abstract=7166739.

Appendix A: Notation

The trigger fraction φ is the programme's proximity parameter and is distinct from the value function denoted ϕ in the foundational paper; the two symbols are never used in the same role. Three further symbols are chosen to avoid collisions with the companion papers. The placement offset is written z because δ denotes displacement from the range midpoint throughout the programme, the proximity hazard is written η because h denotes range half-width, and the occupation density is written ψ because g denotes the local tick sensitivity of the value share. The jitter scale carries a subscript in σ_J to separate it from realised volatility, and the folded distance d of this paper is a distance within a range, unrelated to the per-event dust credits written $d^{\pm}$ in the companion work. The table records every symbol used in the method and in the propositions.

Table 7. Notation used throughout the paper.

Symbol	Meaning
Trigger geometry and the fingerprint.	
φ	trigger fraction, the proximity policy parameter recovered
$[\ell, u]$	a concentrated liquidity range in ticks, lower and upper bound
W	range tick width, $u - \ell$
u^*	price position within the previous range at firing; in $[0, 1]$ for an in-range firing
d	distance from the nearer edge, $\min(u^*, 1 - u^*)$
ft	pool tick at firing, read one block before the rebalance
τ	pool tick spacing
z	placement offset, $(ft - (\ell' + u')/2)/\tau$ for new range $[\ell', u']$
Firing model, estimator, and defence.	
p	pool tick at an arbitrary moment
η	proximity hazard, active only inside the trigger zone
λ	scheduled hazard, active at every position
ψ	occupation density of d over the observation window
f	firing-position density, whose discontinuity is recovered
n	number of firings in an estimation sample
N	horizon in blocks, in the anticipation test
σ_J	standard deviation of the trigger jitter
ξ_i	the jitter draw applied to rebalance i
K_{σ_J}	distribution function of the jitter draws

Appendix B: Discovery handle precision

The gauge handle of Section 3.1 is a threshold rule, admitting a staker that spans at least a given number of distinct pools, and it can be scored against the operators verified independently. Over a window carrying two thousand six hundred candidate stakers, every verified operator survives to a ten-pool threshold and the handle loses its first at fifteen pools. What it admits wrongly is a real population, not an artefact, ordinary accounts staking across many gauges, and their share falls from about one admitted candidate in four at a two-pool threshold to one in fifteen at twenty. Raising the threshold therefore trades recall for purity in the expected direction, and the paper runs the handle generously and lets the fingerprint sort what it admits.

The behavioural filter that suggests itself does not work. Deposits per staked position fail to separate verified operators from the rest, both sitting at a median near one and a fifth, so the quantity that looks like it should distinguish an automated manager from a patient provider does not. Whether the staker carries code does separate them, and that is the only cheap filter applied before the fingerprint.

The mint-handle half of discovery is not scored the same way. Owner-keyed precision is inflated by counting venue custody singletons as operators, and sender-keyed precision is bounded by the size of the verified positive set rather than by the handle, so neither number would measure what it appears to measure. The population is instead reported with its coverage and scope (Section 6.5).

Appendix C: Operator grouping and its code test

Single-pool vault families appear as separate owners under the mint handle and are assembled by grouping on the shared mint sender (Section 10). Grouping on a shared sender alone cannot distinguish a platform keeper running many vaults from an automation service acting for unrelated customers, so each assembled component is put to a second, independent test on the deployed code of its members. A component counts as one operator family when its members share an implementation or, where per-pool vaults compile their configuration into the runtime code, a common product interface.

The five largest components, holding two hundred and eleven of the grouped owners and all but two of the verified operators, resolve under that test, three on shared bytecode above nine tenths of members and two on a common product interface. A residue of small components, none larger than four owners, is left unattributed, one because it resolves under neither test and the rest because they fall below the sampling threshold at which the test was run.

The test also disciplines the grouping itself, since the components it links must be checked for custody layers admitted as owners by mistake. A venue position manager left in the owner set chains every keeper that mints through it into one spurious component, and the code test identifies such a bridge by name. This is not hypothetical, since an earlier grouping run that filtered three custody addresses where the population carries eleven produced exactly that artefact, a single component of a hundred and fifty-three owners that the corrected filter dissolves.

Appendix D: Data-quality sweeps

The recovery depends on two guards and one exclusion, and each is swept, not assumed. The staleness guard bounds how far before a rebalance the last swap may sit for its tick to be taken as the firing tick. The comparison band sets the width just inside and just outside the candidate line over which the step ratio is formed. The full-range exclusion drops near-full-range legs before a width or a fraction is read.

On the control operator, whose ground truth is known, the recovered fraction is 0.25 at every staleness guard from a hundred blocks to unbounded, admitting between 85,076 and 89,379 firing positions as the guard relaxes, with the inside-to-outside ratio drifting only from 2.05 to 2.15. Widening the comparison band from 0.03 to 0.05 leaves the recovery at 0.25, and 0.08 moves it by one grid step to 0.26, which is the resolution effect Section 4 describes rather than a change in the underlying value.

Across the discovered population the same guards move the clean-cell count by a few cells without changing the shape of the recovered distribution, and the full-range exclusion is inert until it is tightened far enough to delete pools outright (Table 8), whose baseline is the wider population gate of Section 6.1, one cell above the reported count. At a hundred thousand ticks the exclusion is no longer removing near-full-range legs but removing pools, and the surviving count reflects that rather than any instability in the recovery.

Table 8. Population clean-cell count by staleness guard and by full-range-exclusion threshold.

Setting	Clean cells
Staleness guard (blocks)	
225	101
450 (sweep baseline)	103
900	103
1,800	105
unbounded	107
Full-range exclusion (ticks)	
887,000 (sweep baseline)	103
443,000	103
100,000	21