Stage Gating for Robust FX Strategy Research

A Purged Walk-Forward + Bootstrap Framework, with Microstructure Entry Refinement as Stage 2

Lucitech Computer Solutions — Quant Research

Author: Sean Plows

Date: 3 February 2026

Version: 1.1 (living document)

Online hub: https://lucitech.co.uk/lucitech-quant-research/

Contents

1 Motivation and research goal 4

2 Data and instrumentation 4

2.1 Event sources 4

2.2 Trade representation 4

3 Notation and core definitions 4

3.1 Returns and excursions 5

3.2 Event labels 5

3.3 Gates 5

4 Validation protocol (time-series first) 6

4.1 Purged and embargoed walk-forward 6

4.2 Block bootstrap for uncertainty 6

4.3 Multiple testing controls 6

5 Stage gating research design 7

5.1 Stage A — Candidate screening (cheap) 7

5.2 Stage B — Walk-forward validation (proper) 7

5.3 Stage C — Stress and realism checks 7

6 Interpretable machine learning in the research loop 7

6.1 ML as a candidate generator (shallow trees) 8

6.2 ML as quantification (regularised logistic regression) 8

6.3 Why ML usage is deliberately constrained 8

7 Stage 2: microstructure entry refinement (overview) 8

7.1 Theoretical earlier entry 9

7.2 Microstructure features around τ 9

7.3 Stage 2 evaluation rule 9

8 Engineering stack and reproducibility **10**
8.1 Pipeline principles 10
8.2 Architecture sketch (conceptual) 10
8.3 Why this matters 10

9 Reporting policy: what I publish (and what I don't) **10**

10 Current status (at time of writing) **11**

11 Roadmap for Part 2 **11**

A Checklist for each experiment **11**

Abstract

Retail trading research often fails for one reason: it confuses *found patterns* with *robust evidence*. This paper outlines a framework for systematic FX research designed to avoid "holy grail" hunting by enforcing: (i) decision-time feature constraints, (ii) purged + embargoed walk-forward validation, (iii) block bootstrap uncertainty, and (iv) safeguards against multiple testing.

The research programme is structured as a two-stage funnel:

1) **Stage gating:** discover and validate interpretable veto rules that remove low-quality regions of feature space and improve conditional outcome probabilities out-of-sample.
2) **Microstructure entry refinement (Stage 2):** only after Stage 1 is stable, test whether tick-level microstructure variables can bring entry forward and improve trade quality without increasing adverse excursion.

Part 1 focuses on methodology, notation, and the research stack. Later parts will publish empirical findings as the programme progresses.

1 Motivation and research goal

FX markets are noisy and adaptive. Any research pipeline that tests enough ideas will eventually discover something that looks good in-sample. The goal here is not to find a single magic indicator, but to build a repeatable process that answers:

> *Is there a robust conditional edge that survives realistic time-series validation and costs?*

The most reliable lever I've found in systematic research is **gating**: selectively avoiding trades in conditions that consistently degrade expectancy. This is a conservative approach: it does not require forecasting precisely; it requires identifying *where not to trade*.

Only once gating is robust do I consider Stage 2: whether entry timing can be improved using microstructure variables.

2 Data and instrumentation

2.1 Event sources

The strategy generates:

- **Orders submitted** and **filled trades** (timestamps, direction, prices, lifecycle events).
- **Market data:** bar data (e.g., 1-minute) and tick data (bid/ask, and volumes where available).

2.2 Trade representation

Each trade i is defined by:

- Decision time t_i (when a trade is committed/filled; defined consistently per experiment).
- Direction $d_i \in \{+1, -1\}$ (LONG/SHORT).
- Entry price p_i.
- A vector of **decision-time features** $x_i \in \mathbb{R}^k$.

Key discipline: no feature may use information after t_i.

3 Notation and core definitions

Let P_t be a price process (bid, ask, mid, or another consistent convention).

3.1 Returns and excursions

For trade i entered at t_i with entry price p_i, define an evaluation window $[t_i, t_i + H]$.

Directional signed move:

$$\Delta P_i(t) = d_i \cdot (P_t - p_i). \tag{1}$$

Maximum favourable excursion (MFE) over horizon H:

$$\mathrm{MFE}_i(H) = \max_{t \in [t_i, t_i + H]} \Delta P_i(t). \tag{2}$$

Maximum adverse excursion (MAE) over horizon H:

$$\mathrm{MAE}_i(H) = \min_{t \in [t_i, t_i + H]} \Delta P_i(t). \tag{3}$$

(Excursions are measured consistently in pips or price units.)

3.2 Event labels

A common trap is using "MFE is positive" as a tradability claim. Many losing trades briefly go positive. Instead, define event labels that reflect tradability.

Definition 1 (Hit-first event label)**.**

$$y_i = \mathbb{1}\{\textit{price hits } +X \textit{ before } -Y \textit{ within } H\}. \tag{4}$$

Where X and Y are thresholds in pips, typically tied to volatility/spread constraints such as:

$$X = \max(\kappa \cdot \mathrm{spread}, \, \mu \cdot \mathrm{ATR}). \tag{5}$$

This makes the label more resistant to "predicting noise".

3.3 Gates

A **gate** is a boolean function of decision-time features:

$$g(x_i) \in \{0, 1\}, \tag{6}$$

where $g(x_i) = 1$ means "trade allowed", and $g(x_i) = 0$ means "veto".

A key quantity of interest is uplift in a target metric, for example hit probability:

$$\Delta = \mathbb{E}[y \mid g(x) = 1] - \mathbb{E}[y]. \tag{7}$$

Or lift:

$$\text{Lift} = \frac{\mathbb{E}[y \mid g(x) = 1]}{\mathbb{E}[y]}. \tag{8}$$

4 Validation protocol (time-series first)

4.1 Purged and embargoed walk-forward

Time-series validation must avoid leakage from temporal dependence and overlapping trades. I use walk-forward validation with:

- Training window: T_{train} days
- Test window: T_{test} days
- **Embargo:** remove samples within an embargo period around fold boundaries (e.g., 24 hours) to reduce contamination from adjacent time segments.

This enforces a realistic "train on past, test on future" regime.

4.2 Block bootstrap for uncertainty

Financial outcomes are autocorrelated and heavy-tailed. I use **block bootstrap** (e.g., by day) to estimate uncertainty for fold outcomes and uplift metrics.

If Z is a statistic (lift, win-rate uplift, profit-factor proxy, mean return), estimate a distribution:

$$\{Z^{(b)}\}_{b=1}^{B}, \tag{9}$$

from which confidence intervals and stability measures can be derived.

4.3 Multiple testing controls

If hundreds of candidate gates are tested, some will appear significant by chance. To reduce selection by noise, I apply multiple-testing discipline and require **walk-forward stability** rather than single-period wins.

Practical promotion standards:

- Evidence must appear in multiple folds, not one.
- Gates should remain interpretable and operationally simple.

5 Stage gating research design

This programme uses a funnel:

5.1 Stage A — Candidate screening (cheap)

Goal: rapidly identify candidate gates that show uplift.

Candidate generation from:

- simple thresholds (spread bucket, ATR bucket, session/hour, regime flags, confidence bins),
- small combinations of conditions,
- "pocket" discovery (bins and intersections).

Evaluation includes uplift and retention (how many trades remain), plus minimum trade-count and minimum hits in test segments.

5.2 Stage B — Walk-forward validation (proper)

Goal: ensure uplift persists out-of-sample.

- Apply purged/embargoed walk-forward.
- Estimate uncertainty via block bootstrap.
- Promote only gates that show consistent benefit across folds.

5.3 Stage C — Stress and realism checks

Goal: ensure robustness is not an artefact.

- Costs/spread/slippage stress (sensitivity analysis).
- Regime drift checks (by year/session/volatility regimes).
- Failure mode analysis: where does the gate break?

The output of Stage 1 is not "the strategy". It is a **policy constraint**: a compact set of veto rules that reduces exposure to adverse conditions.

6 Interpretable machine learning in the research loop

In parts of the pipeline I use machine learning as a **research tool**, not as a deployable black-box trading model.

Principle 1 (ML as hypotheses)**.** *ML outputs are treated as* hypotheses *(candidate gates or quantified relationships). Promotion depends on out-of-sample stability under the validation protocol above.*

6.1 ML as a candidate generator (shallow trees)

To efficiently search for compact, interpretable veto rules, I use **shallow decision trees** as a rule generator. The tree is trained on decision-time features x_i with an appropriate label (e.g., y_i or a loss-event proxy), and then distilled into human-readable conditions.

A typical distilled gate resembles:

> "If spread is high and the market regime/session is unfavourable, veto."

Importantly:

- the tree itself is not the final model,
- rules are extracted and then re-tested independently via purged walk-forward + bootstrap,
- only compact rules that remain stable are promoted.

This provides a practical compromise: algorithmic discovery with auditable outputs.

6.2 ML as quantification (regularised logistic regression)

Where useful, I use **regularised logistic regression** to quantify associations and monitor stability/drift-like behaviour under feature constraints. Regularisation reduces degrees of freedom and helps avoid fitting noise when features are correlated.

These fits inform prioritisation and hypothesis testing; they are not treated as "alpha" unless supported by out-of-sample evidence.

6.3 Why ML usage is deliberately constrained

The primary risk with ML in finance is not implementation complexity; it is overfitting under multiple testing. By limiting ML to interpretability-first roles (candidate generation and quantification), and enforcing strict out-of-sample validation, the pipeline stays aligned with the goal: robust conditional probability rather than fragile curve-fit.

7 Stage 2: microstructure entry refinement (overview)

Once gating is stable, I test a separate hypothesis:

Conditional on "good" gated conditions, can tick-level variables reliably improve entry timing and trade quality?

7.1 Theoretical earlier entry

For each trade, define a local bar-based "cycle window" around the actual entry time. Over that window, compute a range:

$$R = P_{\text{high}} - P_{\text{low}}. \tag{10}$$

Define a theoretical entry level using an entry fraction $\alpha \in (0, 1)$:

$$\text{LONG:} \quad p^\star = P_{\text{low}} + \alpha R, \tag{11}$$

$$\text{SHORT:} \quad p^\star = P_{\text{high}} - \alpha R. \tag{12}$$

Then use tick data to find the earliest timestamp τ where the mid price crosses $p^\star$.

7.2 Microstructure features around τ

At/around τ, compute microstructure features such as:

- spread statistics in short windows pre/post τ,
- short-horizon signed move statistics,
- volume imbalance proxies (when volume is available),
- measures of choppiness vs directional persistence,
- time-to-cross and subsequent short-horizon MFE/MAE.

7.3 Stage 2 evaluation rule

Stage 2 is only "successful" if earlier entry improves outcomes **without increasing risk**:

- improves probability of reaching $+X$ before $-Y$ within H,
- improves MFE distribution and does not worsen MAE beyond tolerance,
- remains stable across walk-forward folds.

This is deliberately conservative: fragile signals are not promoted.

8 Engineering stack and reproducibility

8.1 Pipeline principles

- **Reproducible runs:** each experiment tagged with a run identifier (e.g., `run_id`).
- **Decision-time enforcement:** features must be computable at t_i without future leakage.
- Separation of concerns: execution/telemetry, dataset construction, validation harness, reporting artifacts.

8.2 Architecture sketch (conceptual)

```
Strategy execution (orders/trades/events)
        |
        v
PostgreSQL telemetry + market data
        |
        v
Dataset builder (labels + features)
        |
        +--> Stage 1: gating / WF / bootstrap / promotion
        |
        +--> Stage 2: theoretical entry + tick microstructure features
        |
        v
Reports: notes + charts + CSV summaries
```

8.3 Why this matters

The stack is not “the edge”. The stack prevents self-deception: it makes it easy to test hypotheses quickly, hard to leak future information accidentally, and makes negative results useful (they close doors).

9 Reporting policy: what I publish (and what I don’t)

I publish:

- methodology and validation discipline,
- stability evidence and limitations,
- engineering approach and research notes.

I do not publish:

- live deployable parameters or rule sets,
- security-sensitive operational details,
- anything presented as guaranteed performance.

10 Current status (at time of writing)

- Stage gating research is ongoing, with emphasis on out-of-sample stability.
- Microstructure entry refinement is tested only after gating reduces the hypothesis space.
- The stack evolves to improve iteration speed, auditability, and repeatability.

11 Roadmap for Part 2

Part 2 will focus on empirical results from Stage 2, including:

- whether earlier entry opportunities exist conditionally,
- which microstructure variables (if any) survive walk-forward,
- how entry improvements interact with risk controls (MAE and tail behaviour).

A Checklist for each experiment

1) Define label y_i (horizon H, thresholds X, Y, cost assumptions).
2) Define features x_i and confirm they are decision-time valid.
3) Stage A screening with conservative minimum sample sizes.
4) Stage B walk-forward with embargo and block bootstrap.
5) Apply multiple-testing discipline; promote only compact gates.
6) Stress test costs, regime segmentation, and drift.
7) Write a short research note: what worked, what failed, what's next.