Bar Time Is Not Information Time

A Causality Contract for Quantitative Trading Pipelines

Edmen Wong
AQP TECH ENTERPRISE
Alpha Quant Pro / Alpha Tick Lab Research

Technical Note 01 - Version 1.0 - 16 August 2026

DOI

10.5281/zenodo.21965788

https://doi.org/10.5281/zenodo.21965788

Canonical research page

https://alphaquantpro.com/research/bar-time-is-not-information-time

Suggested citation

Wong, Edmen. (2026). *Bar Time Is Not Information Time: A Causality Contract for Quantitative Trading Pipelines*. Alpha Quant Pro / Alpha Tick Lab Research, Technical Note 01, Version 1.0. DOI: 10.5281/zenodo.21965788.

Bar Time Is Not Information Time

A Causality Contract for Quantitative Trading Pipelines

Edmen Wong
AQP TECH ENTERPRISE · Alpha Quant Pro / Alpha Tick Lab Research
Technical Note 01 · Version 1.0 · 16 August 2026

Canonical URL: https://alphaquantpro.com/research/bar-time-is-not-information-time
Status: Public technical note; not peer reviewed.
Scope: Research methodology only. No claim of live profitability or investment performance.

Abstract

Look-ahead bias in quantitative research is often reduced to explicit future-row access: a negative shift, a centered rolling window, a globally fitted transformation, or a target that accidentally enters the feature matrix. Those failures are important, but they do not exhaust the ways a backtest can acquire information too early. A pipeline can avoid every obvious future-row operation and still leak if it treats a storage timestamp as the time at which the represented information actually became available.

This note proposes an explicit information-time contract for completed market bars, supervised labels, research selection, walk-forward retraining, and event-driven execution. The core distinction is between *bar time* and *information availability time*. For a completed bar, the final high, low, close, and volume are not known at the bucket open. Accordingly, a causal research system should represent `bar_open_time < bar_close_time = available_at = feature_time` for features derived from the full interval. Supervised targets should additionally carry an explicit `label_end_time`, with training eligibility determined from label maturity rather than row distance alone. Factor selection and other train-owned research decisions should be recomputed inside each fold's training prefix, and market execution should occur strictly after the signal's information time.

The proposed contract is paired with adversarial regression tests. Prices strictly after a cutoff are mutated and the pipeline is recomputed; all feature values at or before the cutoff must remain invariant, while an explicitly future-looking label at the cutoff may change. Validation-only mutations must not alter train-owned selection evidence. Same-timestamp signal execution remains invalid. These tests do not establish economic value. Their narrower purpose is to make causal assumptions observable, attackable, and reproducible across a research-to-execution pipeline.

Keywords: look-ahead bias; information leakage; backtesting; bar timestamps; information availability; label maturity; walk-forward validation; financial machine learning; algorithmic trading.

---

1. The leakage problem is broader than future-row access

The most familiar examples of look-ahead bias are syntactic. Code reviewers search for constructs such as `shift(-1)`, centered windows, future joins, or a scaler fitted on the full sample. These are useful checks because they expose direct access to later observations. However, they also encourage a misleading mental model: if the feature function never reads a row with a later timestamp, the feature must be causal.

That conclusion depends on what the timestamp means.

Consider a 15-minute OHLC bar covering the interval from 10:00:00 through 10:14:59. At 10:00:00 the eventual high, low, close, and volume of that interval do not exist. If a data system stores the completed bar under the label `10:00:00`, a feature function can consume only rows whose labels are less than or equal to 10:00:00 and still use

information observed almost fifteen minutes later. The code contains no negative shift, yet the decision clock is wrong.

This is a semantic form of leakage. The failure is not that the pipeline reads a future row; the failure is that a row has been assigned an information time earlier than the observations it contains.

Recent work reinforces the importance of decision-time semantics. Zhang, Li, Peng, and Chen (2026) report that same-day-open execution using post-open daily-bar information can create large and stable inflation even under otherwise paired backtest protocols. Mir and Shrestha (2026) likewise demonstrate that leakage mechanisms interact materially with walk-forward evaluation and can alter apparent backtest performance. These findings complement the earlier literature on backtest overfitting by Bailey and co-authors: evaluation integrity depends not only on model fitting, but on the entire protocol through which information reaches a simulated decision.

2. Four meanings of time for a completed bar

A research system should avoid overloading a single timestamp with incompatible meanings. For a completed interval bar, at least four concepts are useful:

1. **bar_open_time** — when the aggregation interval begins;
2. **bar_close_time** — when the interval is complete;
3. **available_at** — when the complete information represented by the row is allowed to enter a decision system;
4. **feature_time** — the decision-time authority assigned to features derived from that row.

For a feature that consumes the full completed bar, a conservative invariant is:

```
bar_open_time < bar_close_time = available_at = feature_time
```

The exact equality between close time and availability time is an operational convention, not a claim that every real feed has zero latency. A production feed may require `available_at > bar_close_time` because of transport or processing delay. The important property is monotonicity: information must never be available before it exists.

This distinction also resolves a common resampling error. Many data libraries label a resampled interval by its left boundary by default. That label may be perfectly acceptable as a bucket identifier while being unsafe as a decision-time authority. Storage identity and information availability should therefore be represented separately whenever the default index semantics are ambiguous.

3. Supervised labels require explicit maturity

Financial-machine-learning targets intentionally use future outcomes. A future-return label is not a bug merely because it depends on a later price. The bug occurs when a row containing that target is admitted to training before the future outcome would have matured relative to the validation boundary.

Each supervised row should therefore record an explicit `label_end_time`. The basic relation is:

```
feature_time = available_at
label_end_time > feature_time
```

A row is eligible for a training fold only if its target has fully matured before the validation period begins:

```
training_row.label_end_time < validation_start
```

This makes the causal dependency visible in the data contract. Instead of inferring maturity from a configured horizon that may be interpreted differently by downstream modules, the row carries the end of the information interval consumed by its target.

Explicit maturity is particularly useful for event-driven labels. A trade outcome may terminate early at a stop, remain open until a barrier, or expire at a maximum horizon. Two neighboring feature rows can therefore have different target end times. Row number alone does not capture that information geometry.

4. Why fixed purge distance is weaker evidence

Purging a fixed number of bars before validation is a useful approximation when every label has an identical, reliable horizon and the time grid is regular. It is not a universal proof of independence.

Suppose the nominal horizon is 16 bars. A split implementation may remove the last 16 training rows and assume that all remaining labels are mature. That assumption can fail when bars are missing, sessions are irregular, labels terminate on events, data are aligned across timeframes, or a target's true observation interval differs from its nominal row count.

When `label_end_time` is available, the stronger rule is to test the timestamp directly. Fixed-bar purging can remain a defensive parameter, but it should not override explicit evidence about when the target became knowable.

This leads to a broader principle: **causality belongs to information intervals, not array positions.**

5. Selection leakage: chronological fitting is not enough

A model can be fitted only on past rows and still inherit validation information through upstream research decisions. Factor ranking, correlation filtering, feature selection, principal-component fitting, hyperparameter search, threshold selection, and ensemble design all have the potential to observe data that the final estimator never sees directly.

For an outer research split, the ownership boundary should be explicit:

```
outer train rows
    -> factor statistics
    -> FactorSet / feature selection
    -> model fit

validation rows
    -> excluded from train-owned statistics and selection
```

The same logic applies within walk-forward retraining. Each fold should recompute the statistics and selection state owned by that fold's training prefix. Reusing a feature set chosen from the complete research period can turn walk-forward into chronological refitting on top of globally informed selection.

The practical test is adversarial. Mutate only the validation window. If a train-owned selection artifact changes, validation information reached the selection process. Mutate the training prefix instead; the input and selection identities should change. This paired test establishes both *insensitivity to unauthorized evidence* and *sensitivity to authorized evidence*.

6. Signal time and execution time are different authorities

Research signals introduce another timing boundary. A signal derived from a completed bar becomes available only after the information used by its feature computation is available. The order then needs an execution event that occurs after that decision.

For a conservative event-driven backtest:

```
execution_time > signal_time
```

This rule avoids the circular convention in which a strategy sees the completed state of an event and is simultaneously filled using the state of that same event. Depending on the market and order type, a more specific simulator may model queue priority, latency, spread, or intrabar sequencing. The strict-later invariant is not a complete microstructure model; it is a lower-bound causal condition.

A content hash or immutable run identifier does not repair incorrect time semantics. If a result is rehashed after setting `execution_time == signal_time`, it is still causally invalid. Identity proves which bytes were evaluated; it does not prove that their semantics were correct.

7. Adversarial causality regression

Static inspection should be complemented by behavioral tests that deliberately attack the future.

7.1 Feature prefix invariance

Choose a cutoff time `T`. Compute the full feature matrix. Then multiply, replace, or otherwise perturb market prices strictly after `T` by a large amount and recompute the features. For every feature timestamp `t <= T`, the values should remain identical.

An explicit future label at `T` is allowed to change because its future dependency is declared. This asymmetry is desirable: features must be prefix-invariant, while labels should respond to the future outcomes they are designed to encode.

7.2 Bar-availability adversary

Construct a bar whose intrabar high or closing value differs materially from the state visible at the bucket open. The complete OHLC row must not become accessible at `bar_open_time`. If the research system can consume that final state at the open timestamp, the test fails even if no future index was accessed.

7.3 Selection-isolation adversary

Mutate only validation rows and verify that train-owned factor statistics and selection identities remain unchanged. Then mutate training rows and verify that those identities do change. This prevents a false pass from a selection procedure that simply ignores all inputs.

7.4 Execution-time adversary

Generate a valid research signal and resolve the next executable market event. Then alter the result so that execution occurs at the same timestamp as the signal and recompute its identity. The system should still reject it on semantic grounds.

Together these tests shift the question from “does the code look causal?” to “can information from the future influence an earlier decision under controlled attack?”

8. Compact causality contract

The proposed contract can be summarized as four rules:

```
1. bar_open_time < bar_close_time <= available_at
2. feature_time = available_at and label_end_time > feature_time
```

```
3. training rows require label_end_time < validation_start; train-owned selection is fold-
local
4. execution_time > signal_time
```

A stricter system can add feed latency, embargo periods, exchange calendars, asynchronous source releases, or event-time watermarks. The value of the compact contract is that it creates a common minimum across dataset construction, model research, validation, and execution replay.

The contract does **not** establish that a strategy is profitable, robust, or economically meaningful. A perfectly causal model can still overfit noise. It can suffer survivorship bias, bad transaction-cost assumptions, regime instability, multiple-testing bias, or production slippage. Causality is one necessary layer of evaluation integrity, not a substitute for the rest.

9. Implementation provenance in Alpha Tick Lab

The methodology described here is implemented as a bounded research-validation contract in Alpha Tick Lab. The current adversarial gate checks:

- official interval-bar information is exposed at bar availability time rather than bucket-open time;
- feature prefixes remain invariant when market values strictly after a cutoff are mutated;
- every supervised research row carries explicit label maturity;
- outer factor selection uses train-owned evidence only;
- fold-local walk-forward selection is invariant to validation-only mutations;
- fold training mutations change the corresponding research identities;
- same-timestamp signal execution is rejected.

This implementation evidence is intentionally narrower than a production claim. It does not grant live execution authority, establish model quality, or prove that any strategy will earn positive returns. The purpose is to enforce a temporal contract inside a reproducible research workflow.

10. Limitations

First, different data vendors and exchanges use different timestamp conventions. Some label bars by interval start, some by interval end, and some expose multiple timestamps. The contract requires the researcher to identify the actual semantics rather than assume one convention.

Second, real information availability can be later than the nominal market timestamp because of network, processing, publication, or synchronization delay. Macroeconomic and fundamental data can also be revised after first release. Such systems may require separate event time, release time, receipt time, and revision-vintage fields.

Third, this note focuses on temporal and selection leakage. It does not cover every source of backtest overfitting. Multiple testing, repeated strategy search, cross-sectional survivorship, universe construction, execution-cost calibration, and discretionary researcher intervention need separate controls.

Finally, the adversarial tests described here prove bounded invariants under the tested transformations. They do not constitute formal verification of an arbitrary research codebase.

11. Discussion

The practical disagreement raised by this note is not whether obvious future shifts are wrong. It is whether a timestamp should be trusted as sufficient evidence of information availability.

Our position is no. A timestamp is a label until the pipeline defines what event it represents.

This leads to several questions worth debating:

1. Should completed OHLC bars always be relabeled to close time, or is a separate `available_at` field preferable?
2. Is fixed-bar purging acceptable when explicit label-end timestamps can be computed?
3. Can a walk-forward result be called leakage-controlled if factor selection was global?
4. What is the minimum causal execution convention for a signal generated from a completed bar?
5. Should research frameworks ship adversarial future-mutation tests as standard validation tools?

These are methodological questions rather than product claims. Different systems may choose different representations, but the information boundary should be explicit enough to test.

12. Conclusion

A quantitative backtest can cheat without a single obvious `shift(-1)`. The failure can begin when a row containing later interval information is labeled as if that information existed earlier. Once that semantic error enters the dataset, otherwise disciplined feature code, walk-forward fitting, and immutable artifacts can faithfully reproduce the wrong timeline.

The remedy is to make time authority explicit. Represent when a bar becomes available, when a supervised target matures, which rows own selection decisions, and when a signal can execute. Then attack those boundaries with future mutations rather than trusting naming conventions or code appearance.

Time labels are not information guarantees. Causal research needs both.

References

1. Bailey, D. H., Borwein, J., López de Prado, M., & Zhu, Q. J. (2015). *The Probability of Backtest Overfitting*. Journal of Computational Finance / SSRN. https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2326253
2. Bailey, D. H., Borwein, J., López de Prado, M., Salehipour, A., & Zhu, Q. J. (2016). *Backtest Overfitting in Financial Markets*. Automated Trader / SSRN. https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2731886
3. Mir, Z. A., & Shrestha, D. (2026). *Quantifying Backtest Overfitting from Information Leakage: A Walk-Forward and Embargo-Based Diagnostic Framework Applied to Deep Learning Return Forecasts*. SSRN. https://papers.ssrn.com/sol3/papers.cfm?abstract_id=7029819
4. Zhang, F., Li, Z., Peng, S., & Chen, Y. (2026). *When Alpha Disappears: A One-Switch Benchmark for Decision-Time Leakage in Financial Backtests*. arXiv:2605.23959. https://arxiv.org/abs/2605.23959

Suggested citation

Wong, Edmen. (2026). *Bar Time Is Not Information Time: A Causality Contract for Quantitative Trading Pipelines*. Alpha Quant Pro / Alpha Tick Lab Research, Technical Note 01, Version 1.0. https://alphaquantpro.com/research/bar-time-is-not-information-time

Disclosure: This document is a non-peer-reviewed technical note about quantitative research methodology. Alpha Quant Pro is software, not a broker, bank, investment adviser, or guaranteed-return service. Historical, simulated, or methodological discussion is not a promise of future investment performance.