# ASOF-Bench: Evaluating Temporal Models Under Delayed and Revised Observations

**Anonymous Author(s)**
Affiliation
Address
email

## Abstract

Operational temporal data are not fixed arrays. They are evolving records in which a statement carries both the time a quantity pertains to and the time a system could first know it, and in which values are revised for months. We present ASOF-Bench, a version-aware evaluation suite for such records. It defines a release-event contract over both clocks, builds three public archives against it through adapters sharing one origin and target definition, seals test regions before any score exists, and ships the audits that make a reading checkable, including a training-label availability audit that exposes boundary violations reference-date splits leave behind. Against it we run a reference encoder that reads self- and cross-series release context with both clocks and the revision ordinal as channels, integrates it behind a single removable gate and predicts a persistence residual with an interval, under two controls that hold architecture, parameter count and release timing constant and change only the values. On hospitalisation backfill, authentic release-context values improve prediction over these controls by 38.5% and 41.9% on all five seeds with an interval excluding zero; on a macroeconomic vintage archive the same contrast shows no detectable improvement. Content gains do not establish operational competitiveness: one archive passes both controls yet loses to persistence, and our model is worse than every admitted hub team. Evaluation must preserve not only what happened, but what could have been known when a prediction was made.

## 1 Introduction

An operational forecaster reads a database, not a time series. At time $t$ it may use only the statements published by $t$. Describing one statement takes two timestamps and an ordinal where the usual notation supplies one: the reference time $\tau$ a value describes, the availability time $a$ at which it could first have been acted on, and the position $r$ of the statement in the chain of revisions about that $\tau$. A conventional series keeps only the last element of each chain and discards $a$. Figure 1 makes the gap concrete, and it is not marginal: in the three archives here the median series-and-reference-time pair receives 6, 60 and 5 separate releases, so a model trained and scored on final vintages conditions at every step on statements that did not exist when it would have had to act. A more capable model cannot repair an evaluation protocol that grants access to information the system could not have had.

Revision history is known to improve real-time prediction, and Section 2 places that work. We ask a narrower question: when observations arrive late and are subsequently revised, what was available to the forecaster, and how can the contribution of release-context *values* be separated from release *timing* and from model *capacity*?

We contribute the following.

**C1. Version-aware benchmark construction.** A release-event contract over both clocks, settlement anchored at first availability, chronological splits sealed before any score exists, and three adapters

Submitted to 40th Conference on Neural Information Processing Systems (NeurIPS 2026). Do not distribute.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

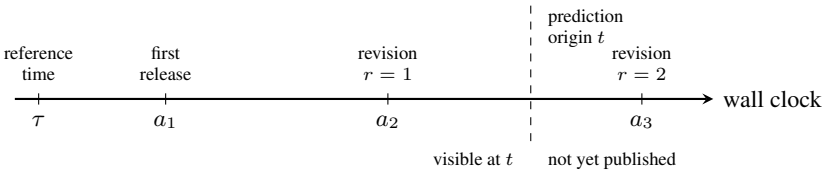



Figure 1: All marks concern one reference time $\tau$. At origin $t$ a forecaster may use the first release and the revision at $a_2$, but not the revision at $a_3$, which revises a quantity pertaining to a past date. A conventional series keeps only the value at $a_3$. Schematic.

emitting one origin and target definition, shipped with point-in-time input filtering, exact-key and completeness checks, per-run provenance, and a training-label availability audit measuring a boundary reference-date splits do not enforce.

**C2. An auditable reference model with matched controls.** A fixed event encoder over self- and cross-series release context, integrating it through query-conditioned attention behind a single removable gate and predicting a persistence residual with an interval, plus two controls that hold architecture, parameter count and release timing constant and change only the values.

**C3. Evidence-tiered evaluation.** We separate incremental value under matched controls, competitiveness against strong and external baselines, and permission to open a confirmation split: two held-out readings against pre-registered expectations, a development case, and a failed successor gate.

## 2 Related work

Revision history is known to help. Back2Future [15] feeds full backfill sequences to a graph-recurrent encoder that refines an existing forecaster, and Delphi-RF [23] predicts the finalised value with quantiles scored by WIS, reporting same-model feature-group ablations that include revision magnitude. The literature therefore already studies revision features while holding the model fixed. We complement it with a timing-preserving shuffle that permutes only the value-to-release pairing, and with an explicit as-of contract fixing what is visible at an origin. GRU-D [6] made missingness patterns predictive and a long line of irregular-sampling architectures inherits that assumption [13, 22]; our arms are one falsifiable form of it. Appendix G gives the full placement, including scoring [5, 10], reliability work on pretrained forecasters [2] and the chain-ladder tradition [18]. The contract does not assume the revised quantity is a past observation. Appendix E audits six further environments spanning corrections, republished forecasts and redisclosures; none carries a model result.

## 3 Task and data

**Contract.** A domain is a table of release events $e = (s, \tau, a, v, r)$: a series identifier, a reference time, an availability time, a value and a revision index. A revision is a new event, not an edit. An origin is a pair $\big((s, \tau), t\big)$: standing at $t$ with exactly those events satisfying $a \le t$ visible, predict the settled value of $(s, \tau)$, the value of the last event with $a \le a_{\text{first}} + D$ for $D$ fixed per domain in advance. The anchor is first availability, not the reference time, so a late first print shifts the whole window. A target whose settle window is not fully covered is dropped rather than truncated, and a value published in a vintage dated $d$ receives $a = d + 1$ day.

The input at an origin is the $E_{\text{cross}}$ most recent events across all series plus the $E_{\text{own}}$ most recent of the target's own series, each filtered by $a \le t$. Standardisation uses events available before the training boundary only.

**Archives.** Table 1 summarises the three archives, chosen for contrasting revision structure rather than subject matter; data cards are in Appendix H. ALFRED [7, 9] has small revisions relative to target dispersion, on a window spanning the COVID shock and therefore out of distribution by construction. Delphi [8, 20] has the deepest chains in the corpus, at median revision depth 60. RKI [21] is a reporting triangle from 1,355 daily snapshots whose revisions are structural backfill, and it

Table 1: The three archives. Settlement is anchored at first availability, never at the reference date. Delphi's test split was materialised and structurally audited but never scored, so no Delphi test number appears in this paper.

| Domain | Target | Settle window | Block | Dev | Test | Test status |
|---|---|---|---|---|---|---|
| ALFRED | vintage level | first avail. + 365 d | month | 1,705 | 1,128 | scored once |
| Delphi | outpatient signal | first avail. + 60 d | ISO week | 24,315 | 49,273 | unscored |
| RKI | hospitalisations | first avail. + 40 d | ISO week | 43,435 | 47,243 | scored once |

is the only archive carrying external comparators, in the form of real-time submissions to a public nowcast hub [24]. Only real-time submissions are admissible, since retrospective directories compare against hindsight.

# 4 Model and controls

**Model.** We use a deliberately fixed reference encoder so that input interventions are auditable and architecture changes cannot confound the contrasts. It is not proposed as a new architecture and it is not tuned. Release events are encoded by a masked Transformer encoder, each token carrying a series embedding, the standardised value, its age $t - a$, its reference lag $t - \tau$, its revision index and a target-series flag. A query built from the target series attends to that context through one cross-attention block, and the attended vector enters the query representation only through a single scalar ReZero gate initialised at zero [3]. The head is persistence-residual and probabilistic: it emits a delta and a log scale, and the predictive mean is the first available value plus the delta. The checkpoint is selected on a train-inner split never used for reporting. Scores are per-origin squared error, averaged over origins and then over seeds. Appendix D gives widths, depths, parameter counts and optimiser settings. A fixed encoder buys auditability, not cross-architecture generality.

Two separate properties are checkable. Context reaches the output only through the ReZero gate, so at gate zero the network reproduces the query-only network bitwise; and the training script zeroes the delta head before the first step, so the predictive mean at initialisation equals the first available value bitwise. The second follows from that initialisation, not from the architecture, since the model class alone uses a default linear head and does not have it. Both are asserted in `tests/test_persistence_nesting.py`. Neither says anything about what the trained model attributes.

**Controls.** A model reading as-of events could improve for reasons unrelated to release content: extra capacity, or the mere presence of more tokens. The *null* arm zeroes the context value channel and leaves availability time, reference lag, age, revision index, mask and series flags bitwise identical, so it removes only the values at matched capacity. The *shuffle* arm permutes values across releases within each series, preserving the release schedule and destroying only the value-to-release pairing. The null addresses the alternative that timing alone suffices, the shuffle that values matter only in aggregate. Both remove release-context values as a whole rather than isolating historical revisions from the latest snapshot, and neither identifies a causal mechanism (Appendix J).

**Reading protocol.** Contrasts are formed per seed across 5 seeds and never between averaged arms, with significance from a two-way bootstrap resampling time blocks and seeds jointly, Holm-adjusted across the two contrasts. Blocks are calendar months of the reference date on ALFRED and ISO weeks on Delphi and RKI; the successor protocol of Section 5 blocks by forecast date instead. 5 seeds is a fixed budget, not a power calculation, so it bounds what a null reading can mean; non-significance is never reported as equivalence or as absence of an effect. Calibration transforms are fitted on the training split only. Appendix D gives the full rule and Appendix F the measurement that forced it.

# 5 Results

We report four empirical case studies, two of them held-out test evaluations: RKI and ALFRED were opened once each, while Delphi and a FluSight successor protocol contribute development-stage evidence.

**Sealed tests.** Development readings are in Appendix C: two of three archives pass the mechanism gate, but S1's own opening condition was not met. We therefore froze a successor protocol, S1-CONFIRM, written after seeing those readings and before any test scoring, registering that ALFRED should fail again and RKI should pass. Each test was opened once and both expectations were met; conclusions are limited to what S1-CONFIRM registered.

Table 2: Held-out test confirmations under S1-CONFIRM, opened for scoring once each; counts in Table 1. Values are mean per-origin squared error of the per-series standardised target over five seeds, comparable within a domain, not across. Contrasts seed-paired, $p$ Holm-adjusted within domain, intervals from the same two-way block bootstrap. "LOO same" counts leave-one-series-out reaggregations, which do not retrain, agreeing with the mean's sign.

| Domain | full | control | control | mean diff. | Holm $p$ | seeds $< 0$ | LOO same |
|---|---|---|---|---|---|---|---|
| RKI | 0.120495 | null | 0.196061 | $-7.557e{-2}$ | <0.0001 | 5/5 | 119/119 |
| | | shuffle | 0.207392 | $-8.690e{-2}$ | <0.0001 | 5/5 | 119/119 |
| ALFRED | 0.013941 | null | 0.013944 | $-3.466e{-6}$ | 1.0000 | 4/5 | 45/47 $*$ |
| | | shuffle | 0.013939 | $+1.858e{-6}$ | 1.0000 | 2/5 | 44/47 $*$ |

On RKI the full arm reaches 0.120495 mean squared error over 47,243 origins, against 0.196061 for the null and 0.207392 for the shuffle, relative reductions of 38.5% and 41.9%. Both hold on all five seeds, show no sign change under any of 119 leave-one-series-out reaggregations, and carry Holm-adjusted $p$ below $10^{-4}$ with a bootstrap interval of $[-1.031e{-1},\ -4.355e{-2}]$ excluding zero. Under this encoder and these matched controls, authentic release-context values have incremental predictive value on this archive.

On ALFRED the arms are separated by $-3.466e{-6}$ over 1,128 origins against a common scale of 0.013941, with a bootstrap interval of $[-2.017e{-5},\ +1.768e{-5}]$ containing zero. We report no detectable improvement under this protocol; no equivalence bound was set in advance, so this does not establish equivalence. In development ALFRED had shown a stable reversed effect agreeing in sign under all 48 leave-one-series-out reaggregations, and none of it survived; that check reaggregates frozen contrasts rather than retraining, so it probes stability within an archive, not transfer.

**Content gains do not establish operational competitiveness.** Delphi passes the mechanism gate and still loses to persistence, at 0.068753 against 0.060862. Against 8 teams filing nowcasts in real time, our mean WIS on the common intersection is 834.4 against a team range of 23.7 to 109.6; we win against no team. That comparison uses a different training and scoring grid: it is a stress test of evaluation compatibility and operational competitiveness, not a like-for-like ranking, and we attribute the gap to no architectural cause (Appendix C). A further pre-registration on influenza hospitalisation reporting failed its own gate, at Holm $p = 0.3040$ on both contrasts over 28 forecast-date blocks. It is a protocol-failure case, not a fourth benchmark build.

## 6 Discussion and limitations

**Limitations.** The contract, adapters, controls and audits apply to any model placed on this interface, and the limitations below are properties our own tools measured. The three archives are not a sample from which universal behaviour follows, every reading uses 5 seeds as a fixed budget rather than a power calculation, and RKI is one country, one disease and one pipeline. Each contrast is a statement about this encoder, so the RKI result is not a domain-level or backbone-independent conclusion.

The contract enforces availability of model *inputs*; training labels are separate, since splits run by reference date and a label can settle after the fit that used it. Our own audit measures this: against the first test origin no RKI fit label settles late and 12 of 34,548 ALFRED ones do, while against the first development origin the rates are 1.6%, 3.3% and 7.8% on ALFRED, RKI and Delphi. The held-out RKI reading is clean on this axis and the development readings are not, so we make no look-ahead-free claim for the pipeline. Standardisation and the checkpoint split are availability-gated (Appendix D). The contrasts should next be repeated on competitive backbones, and against a control ablating historical revisions relative to the latest snapshot rather than to timing, under an availability embargo on training labels.

# References


[1] Rishabh Agarwal, Max Schwarzer, Pablo Samuel Castro, Aaron Courville, and Marc G. Bellemare. Deep reinforcement learning at the edge of the statistical precipice. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. arXiv:2108.13264.

[2] Abdul Fatir Ansari, Lorenzo Stella, Caner Turkmen, Xiyuan Zhang, et al. Chronos: Learning the language of time series. *Transactions on Machine Learning Research*, 2024. arXiv:2403.07815.

[3] Thomas Bachlechner, Bodhisattwa Prasad Majumder, Huanru Henry Mao, Garrison W. Cottrell, and Julian McAuley. ReZero is all you need: Fast convergence at large depth. In *Uncertainty in Artificial Intelligence (UAI)*, 2021. arXiv:2003.04887.

[4] Xavier Bouthillier, Pierre Delaunay, Mirko Bronzi, et al. Accounting for variance in machine learning benchmarks. In *Proceedings of Machine Learning and Systems (MLSys)*, 2021. arXiv:2103.03098.

[5] Johannes Bracher, Evan L. Ray, Tilmann Gneiting, and Nicholas G. Reich. Evaluating epidemic forecasts in an interval format. *PLOS Computational Biology*, 17(2):e1008618, 2021.

[6] Zhengping Che, Sanjay Purushotham, Kyunghyun Cho, David Sontag, and Yan Liu. Recurrent neural networks for multivariate time series with missing values. *Scientific Reports*, 8:6085, 2018.

[7] Dean Croushore and Tom Stark. A real-time data set for macroeconomists. *Journal of Econometrics*, 105(1):111–130, 2001.

[8] Delphi Research Group, Carnegie Mellon University. Delphi Epidata API. Public API serving issue-dated (as-of) epidemiological indicators, `https://cmu-delphi.github.io/delphi-epidata/`, 2026. Accessed 2026-09; issue dates supply the availability timestamps and the revision chains used here.

[9] Federal Reserve Bank of St. Louis. ALFRED: Archival Federal Reserve Economic Data. Public vintage archive of FRED macroeconomic series, `https://alfred.stlouisfed.org/`, 2026. Accessed 2026-09; vintage dates supply the availability timestamps used here.

[10] Tilmann Gneiting and Adrian E. Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102(477):359–378, 2007.

[11] Sture Holm. A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*, 6(2):65–70, 1979.

[12] Hospitalization Nowcast Hub. Hospitalization Nowcast Hub: archived real-time nowcasts of seven-day COVID-19 hospitalization incidence in Germany. Public repository `KITmetricslab/hospitalization-nowcast-hub`, 2026. Accessed 2026-09. Real-time submissions read in place from `data-processed`; the repository carries no `LICENSE` file, so no submission file is redistributed with this work.

[13] Ziniu Hu, Yuxiao Dong, Kuansan Wang, and Yizhou Sun. Heterogeneous graph transformer. In *Proceedings of The Web Conference (WWW)*, 2020. arXiv:2003.01332.

[14] Zijie Huang, Yizhou Sun, and Wei Wang. Learning continuous system dynamics from irregularly-sampled partial observations. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. arXiv:2011.03880.

[15] Harshavardhan Kamarthi, Alexander Rodríguez, and B. Aditya Prakash. Back2Future: Leveraging backfill dynamics for improving real-time predictions in future. In *International Conference on Learning Representations (ICLR)*, 2022. arXiv:2106.04420.

[16] Hans R. Künsch. The jackknife and the bootstrap for general stationary observations. *The Annals of Statistics*, 17(3):1217–1241, 1989.

[17] Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. In *International Conference on Learning Representations (ICLR)*, 2018. arXiv:1707.01926.

[18] Thomas Mack. Distribution-free calculation of the standard error of chain ladder reserve estimates. *ASTIN Bulletin*, 23(2):213–225, 1993.

[19] David Picard. torch.manual_seed(3407) is all you need: On the influence of random seeds in deep learning architectures for computer vision, 2021. arXiv:2109.08203.

[20] Alex Reinhart, Logan Brooks, Maria Jahja, Aaron Rumack, et al. An open repository of real-time COVID-19 indicators. *Proceedings of the National Academy of Sciences*, 118(51): e2111452118, 2021.

[21] Robert Koch Institute. COVID-19 hospitalization incidence reporting triangle for Germany. Daily public snapshot files of reported hospitalization incidence, `https://github.com/robert-koch-institut/COVID-19-Hospitalisierungsinzidenz-in-Deutschland`, 2026. Accessed 2026-09; snapshot dates supply the availability timestamps used here.

[22] Yulia Rubanova, Ricky T. Q. Chen, and David Duvenaud. Latent ODEs for irregularly-sampled time series. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019. arXiv:1907.03907.

[23] Jingjing Tang, Aaron Rumack, Bryan Wilder, and Roni Rosenfeld. Real-time forecasting of data revisions in epidemic surveillance streams. *PLOS Computational Biology*, 21(11):e1013709, 2025. doi: 10.1371/journal.pcbi.1013709. Code: `https://github.com/cmu-delphi/DelphiRF`.

[24] Daniel Wolffram, Sam Abbott, Matthias an der Heiden, Sebastian Funk, Felix Günther, et al. Collaborative nowcasting of COVID-19 hospitalization incidences in Germany. *PLOS Computational Biology*, 19(8):e1011394, 2023. doi: 10.1371/journal.pcbi.1011394.

# A Frozen provenance chain

Table 3 lists every artifact that governs the reading reported in this paper, with the first sixteen hexadecimal digits of its SHA256 digest. The protocol, the split and seal table, the gate register with its pass conditions and fixed reading order, the context-selection rule, the comparator admission rule and the comparison grid were all written, hashed and set read-only *before* any development number existed; the development result of record cites that set by hash, and the next-decision memo records what the result does and does not license. The hashes printed in the table are not transcribed: the script that produces this paper's tables recomputes them from the files themselves on every build, so a drifted artifact would change the printed digest rather than pass silently.

Three protocols appear in this paper and they are not nested versions of one document. **S1** is the original pre-registration governing the three archives and the development reading; its own authorisation condition for opening a test split was not met, so the S1 chain contains no test artifact. **S1-CONFIRM** is a successor protocol, written after the S1 development results had been seen and frozen before any test was scored, registering what ALFRED and RKI should show if those results were real. It is therefore not part of the original S1 registration, and conclusions drawn from it are limited to what it registered. **S2** is a separate pre-registration on FluSight, whose gate failed (Section 5).

The Delphi and FluSight test splits have status TEST_LABELS_MATERIALIZED_STRUCTURALLY_AUDITED_UNSCORED. Their labels were separated and were not used for fitting, selection, calibration, prediction scoring or aggregate evaluation. They were materialised and structurally audited, so we do not describe them as unread or as never touched. The seeds reserved for a confirmation run, the split boundaries and the authorisation rule are all inside the artifacts above, so a successor can open those splits exactly once under a pre-registration written before the reading.

# B Corrections register

Table 4 lists the corrections issued against the frozen S1 protocol artifacts of Table 3. It is deliberately scoped: it is not a register of every error found in the project. The scorer normalisation error and the rebuild it forced are recorded in Appendix F, and the training-label availability audit is recorded in its

Table 3: **Frozen provenance chain.** SHA256 prefixes (first 16 hex digits), recomputed from each artifact at build time by the same script that generates the result tables. In the upper (S1) block the first eleven rows were frozen before any development number existed and the remainder are the single reading and the records it produced; the lower block is the S2 (FluSight) pre-registration and its failed gate (Section 5). No sealed-test artifact appears in this S1 chain, because S1's own opening condition was not met; the two held-out readings of Section 5 were produced later under the separate S1-CONFIRM protocol and are recorded there.

| artifact | SHA256[:16] | role |
|---|---|---|
| S1 protocol | `cb321e6a86598abb` | gates, reading order, authorisation rule |
| as-of event contract | `93ffd8ea3a64b17b` | event schema, origins, settled value, $D{+}1$ rule |
| splits and seals | `74aea29b36adca60` | train / inner / development / sealed test per domain |
| gate register | `f62cc14ac73f28c3` | G0–G14, pass conditions, consequence on failure |
| context rule | `ec2a40652038849f` | candidate set and mechanical selection of $(E_{\text{cross}}, E_{\text{own}})$ |
| comparator admission | `6d53b77db9776497` | which hub directories and teams are admissible |
| grid binding | `e490302ab691cf54` | key grid, target definition, truth source |
| grid binding addendum | `7c4d04664c735b07` | amendment to the key grid |
| comparison manifest V2 | `026755353fe247dc` | the 11 real-time model directories, machine-readable |
| hash binding | `f1a4d699fbc7405c` | binds the gate register hash into the protocol set |
| freeze manifest | `62e93b6fb9b5947e` | code, environment, RNG and launch commands |
| development result of record | `8bb69f2fdc263570` | the single reading; verdict MIXED_EVIDENCE |
| next-decision memo | `974b4423a422a6c3` | what may and may not be said; S2 re-pre-registration |
| mechanism gate record | `2a818cd9b3e3e9c8` | development reading, source of Table 7–9 |
| baseline guard record | `622bad94798a72ec` | development reading |
| competitiveness record | `284fe09ba380d966` | development reading |
| robustness diagnostic record | `29e7744dc1c58c39` | development reading |
| S2 protocol | `97a0902f1161ad8f` | splits, seeds, gates, opening rule; frozen before any S2 training |
| S2 splits and seals | `b4428c53d2870405` | train / selection / sealed test, FluSight |
| S2 score-comparability gate | `18d3334284ae3a05` | which score each gate may use; frozen before training |
| S2 development failure record | `4633d3284c427046` | the single selection-season reading; verdict `DEV_FAIL_NO_TEST_OPENING` |
| S2 dev gate record | `7301afba96c45138` | selection-season reading, source of Table 10 |
| S2 baseline score record | `464cf9267a549087` | selection-season reading, source of Table 10 |

own artifact. None of these was repaired by editing the artifact that contained it. A correction is a new document, hashed separately, that travels with the ones it corrects, so the record of what was believed at freeze time stays intact and auditable.

**Disclosure attached to correction 04.** While re-verifying the ALFRED revision-depth figure, a structural count was computed over the whole of the ALFRED event file, whose rows extend into the sealed ALFRED test period. That computation was a `groupby(...).size()` over release events: no target was formed, no label was read, no loss, model output or split-conditional score was computed, and no quantity derived from sealed-period rows appears anywhere in this paper or in any decision behind it. The re-verification deliberately reports the training and development windows separately and reports no 2023–2024 quantity. The Delphi and RKI event files were not opened by that audit. We record this because a structural count over a file that also contains sealed rows is close enough to the boundary that silence about it would be the wrong choice; the ALFRED test split was unread at the time of that audit. It was opened once on 2026-09-06 under S1-CONFIRM, after the audit and after the retrained arms reproduced the frozen development artifacts bitwise.

# C Guard and hub comparison tables

Table 8 carries the baseline guard summarised in Section 5. Table 9 carries the complete competitiveness reading of Section 5: the common-intersection means (Panel A), the pre-registered per-team seed-paired contrasts (Panel B), and the post-hoc robustness diagnostic (Panel C), which is not a criterion and cannot become one retroactively. The reading is unchanged under the robust view: the model is worse than every admitted team on both statistics, and no win over any team is claimed.

# D Reading protocol in full

Contrasts are formed per seed across 5 seeds and never between averaged arms. Significance comes from a two-way bootstrap resampling time blocks and seeds jointly, Holm-adjusted across the two

Table 4: **Selected corrections to the S1 protocol artifacts.** These are the corrections issued against the frozen S1 documents listed in Table 3. They are not a register of every error in the project: the scorer normalisation error and its rebuild are recorded separately in Appendix F, and the audit of Appendix D is recorded in its own artifact. No frozen artifact was edited; a correction is a new, separately hashed document that travels with the ones it corrects. Corrections 01–03 were issued before any development aggregate existed, and 04 after the reading and before submission, touching descriptive prose only.

| # | SHA256[:16] | what was wrong, and what it does and does not affect |
|---|---|---|
| **01** | `c96ded0d18096636` | Five artifacts stamped 2026-09-05 were written 2026-09-04.<br>*Scope.* Stamps only. Ordering is independently checkable and unchanged; cite these artifacts with the corrected date. |
| **02** | `0d96181d7265dcd5` | A pre-reading note claimed 96 ALFRED events span months; measured span is 1.9 calendar days.<br>*Scope.* The claim was reasoned from series counts, not measured. The ALFRED / Delphi context-span contrast is withdrawn; the frozen $E$ decision stands, and the span audit is the only admissible source. |
| **03** | `90aec1c29921aeed` | The shuffle guard bounded pooled value correlation, which equals the between-series variance share and not the pairing the protocol names.<br>*Scope.* Implementation guard, not a frozen threshold. Replaced by a stricter within-series bound; the decomposition is printed as a diagnostic. No artifact invalidated; only the five blocked runs re-executed, same seeds. |
| **04** | `ca43d240775d83b6` | Two prose counts were wrong: ALFRED revision depth quoted as 5, and hub `data-processed` quoted as 12 model directories.<br>*Scope.* Depth is 1.0 over all reference dates and 6.0 within either the training or the development window, and the window must be named; the admissible set is 11 model directories, of which 8 are WIS-eligible. Both errors are confined to descriptive prose: the governing manifest already held 11 entries and no gate, threshold or admission depended on either figure. The same note discloses a structural read (see below). |

contrasts within a stratum. Leave-one-series-out reaggregations of the frozen per-origin contrasts, which do not retrain, are reported for sign stability, and at least four of five seeds must agree in direction. Averaging arms before testing is what this protocol prevents: in an earlier study on other data, ensembling five seeds and then testing gave $p < 10^{-4}$ where the seed-paired contrast on the same runs gave $p = 0.24$. Non-significance is reported as non-significance and never as equivalence.

**Architecture and optimisation.** Tokens are embedded by summing a learned series embedding of width 64 with a linear projection of the five numeric fields. A two-layer pre-norm Transformer encoder of width 64, four heads, feed-forward width 128 and dropout 0.1 encodes the token set with padding masked. A query built from the target series embedding and four query features by a two-layer tanh network attends to the encoded context through one four-head cross-attention block. The model has 95,875 parameters on ALFRED, 99,331 on Delphi and 100,419 on RKI, differing only in the embedding table. Training minimises Gaussian negative log likelihood with Adam at $10^{-3}$, batch 256, for 20 epochs.

**Training-label availability audit.** The contract enforces availability of model inputs at the origin: context tensors admit only events with $a \leq t$. Training labels are governed separately, because splits are assigned by reference date and the settled label of a reference-legal row may only become available later. We therefore audited, for every row the model is fit or checkpoint-selected on, whether its label's availability time precedes the fitting cutoff, taking the cutoff to be the first prediction origin of the evaluation split.

Relative to the first *test* origin, no RKI fit label settles late, no Delphi fit label settles late, and 12 of 34,548 ALFRED fit labels do, the latest by 1273 days. Relative to the first *development* origin the rates are 557 of 34,548 on ALFRED, 2,162 of 64,617 on RKI and 5,948 of 75,917 on Delphi. The held-out RKI reading, which carries the positive result, is therefore clean on this axis; the development readings are not, and no label-maturity embargo was imposed. ALFRED's held-out reading is a non-detection and its twelve late labels can only add information, so that exposure runs opposite to the reported conclusion rather than producing it.

Table 5: Release environments audited against the contract. Evidence status is what was completed, not what is planned. No entry in this table has a model result, and none supports a performance claim.

| Domain | What changes | Availability clock | Target | Evidence completed | Main limitation |
|---|---|---|---|---|---|
| Elexon | B: official demand forecast | publish time, minute | measured outturn | data/protocol audited | no model run |
| IBES | B: analyst EPS estimate | recorded timestamp | realised EPS | engineering smoke only | licence-restricted |
| SEC EDGAR | C: redisclosed fact | filing acceptance time | final restated value | data/protocol audited | versions often identical |
| ComCat | C: event record update | product update time | catalogue magnitude | data/protocol audited | no model run |
| RSV | A: surveillance backfill | report week | settled count | data/protocol audited | few real-time teams |
| CAS | *none per cell* | none recoverable | ultimate loss | contract test failed | out of contract scope |

Two related boundaries are clean. Per-series standardisation is computed from events with $a$ at or before the train boundary, which precedes every evaluation cutoff above, and the predictive scale is initialised from training-split residuals only. The checkpoint is selected on the inner split and never on a reporting split. We do not claim on this basis that the pipeline as a whole is free of look-ahead; we claim the input-availability contract, and we report the label boundary as audited rather than as guaranteed.

# E Release mechanisms beyond the four case studies

The contract in Section 3 is written over release events and does not assume that the revised quantity is a past observation. Table 5 records six further environments we audited against it. **None of them carries a model result.** The evidence column states what was actually completed: a data and protocol audit, or an engineering smoke test on training rows only. Reporting them as validated domains would misrepresent engineering preparation as empirical evaluation.

Three distinct mechanisms appear, and they are not interchangeable. In **type A** a past observation is corrected or completed as late reports arrive, which is the mechanism in the four case studies of the main text and in RSV. In **type B** a forecast about a future target is republished as that target approaches; the revised object is an official prediction, not a measurement of the past, and completion-delay intuitions from public health do not transfer to it. In **type C** the same fact is redisclosed or an event record is updated, where a new version need not carry a new value at all.

Two entries deserve emphasis because they mark the boundary of the contract rather than extending it. **SEC EDGAR** satisfies the contract structurally, since filings carry an acceptance timestamp and a fact may be restated across filings, but a redisclosure frequently repeats the previous value, so multiple versions do not by themselves imply usable revision content. **CAS** loss-development triangles do not satisfy the contract at all in the data we examined: a development cell is observed once, so there is no within-cell revision chain and no availability clock to read. We keep it as a negative case. It shows that "a table that grows over time" is not the same object as "a record that is revised".

**ComCat** refers to the ANSS earthquake-product archive. It is unrelated to the USGS streamflow gauge network, which shares the agency name.

# F WIS normalisation check

**The identity we rely on.** For a symmetric quantile grid containing the median and using the standard Bracher weights [5], the weighted interval score equals $2\times$ the unweighted mean pinball loss over the corresponding $2K + 1$ quantile levels, where $K$ counts central intervals. The numerator is the sum of pinball losses over those levels and the denominator is $K + \frac{1}{2}$, not the sum of the interval weights. This is a restatement of the standard definition and we claim it as an implementation check, not as a contribution. $K$ depends on the grid: the 23-level grid used by the hub and the 7-level grid used elsewhere in this paper have different $K$ and therefore different denominators, so the identity must be applied per grid rather than with one constant.

**The error we made.** Our first scorer divided by the sum of the interval weights rather than by $K + \frac{1}{2}$, printing the constant as 8.258528. Because the mistake was a fixed factor per scoring chain, rankings, signs, $p$-values and every gate verdict are unaffected, but absolute WIS values computed

that way are not comparable with published ones. A three-way internal check passed because all three implementations shared one misreading of the definition. The error was caught by an external anchor supplied from outside that loop: the case $y = 1$ with all quantiles at $0$, for which WIS must equal $1$ and ours returned 4.129264. An internal consistency check cannot catch a definitional error, and the external anchor takes one line to run. Every WIS number in this paper is produced by the rebuilt V3 chain; no V1 or V2 WIS value is reported anywhere.

**External agreement.** The rebuilt scorer was checked against 12,878 quantile rows shipped by the Delphi-RF authors [23]. Scoring the quantiles exactly as supplied, which is those authors' own native policy, all 12,878 rows agree to a maximum absolute difference of $7.11 \times 10^{-15}$, which is floating-point agreement rather than bitwise equality. The distinction matters here: 70.9% of those rows contain quantile crossing, and applying non-crossing rearrangement first drops agreement to 3,753 of 12,878 rows. Rearrangement is therefore a separate transformation with its own effect on the score. We do not apply it and then describe the result as native reproduction, and no number in this paper mixes the two policies.

**Calibration split.** Fitting a baseline's residual quantiles on the season being reported improved its apparent WIS by 14.6% with point forecasts unchanged. That figure is one measured instance on one baseline and one season, not a general rate of self-calibration optimism, and it should not be transferred to other settings. It is the reason our protocol admits only calibration transforms fitted on the training split: any protocol comparing a model against a quantile-calibrated baseline must fix the calibration split in advance.

# G Related work in full

**Forecasting from revision history.** Back2Future [15] feeds the full backfill sequence of every signal to a graph-recurrent encoder and refines an existing forecaster's real-time prediction toward the stable target. Delphi-RF [23] predicts the finalised value directly, conditions on the full revision sequence, and scores quantiles with WIS on a rolling report-date origin. Both establish that revision history helps, and that result is theirs rather than ours. Delphi-RF additionally reports same-model feature-group ablations, including one over revision magnitude, so the literature does study revision features while holding the model fixed. Our contribution is complementary: a timing-preserving release-value shuffle, which keeps the release schedule intact and permutes only which value was attached to which release, together with an explicit as-of contract that fixes what is visible at an origin. The shuffle asks a question a feature-group ablation cannot, namely whether the value matters through its pairing with a particular release.

**Irregular sampling and informative missingness.** GRU-D [6] observed that missingness patterns are themselves predictive and encoded that through decay terms, and a long line of irregular-sampling architectures inherits the assumption. Our null and shuffle arms are the falsifiable form of the same idea, asking per archive and under a registered pass rule whether structure informs beyond values and whether values inform through their pairing with releases. Cross-series context follows the multivariate graph-sequence lineage [17].

**Continuous-time models.** Latent ODEs [22], graph variants [14] and typed-event transformers [13] model irregular arrivals. The as-of axis is an unusual regime for that family because arrivals are overwhelmingly tied: one vintage or snapshot publishes many statements at a single timestamp. On the interleaved availability axis 98.90% of adjacent ALFRED events and 99.92% of adjacent RKI events carry a zero gap, and even within a single series the rates are 93.58% and 90.64%.

A high tie rate does not by itself show that a continuous-time model has nothing to gain, and we do not claim it does. The distinct release instants in these archives are spaced a median of two days apart on ALFRED and one day on RKI, so a flow still has a span to integrate over, in few large steps. Tie rate, spacing between distinct instants, and total elapsed span are three different quantities and can give three different answers on one archive.

We have a bounded synthetic observation on this, and state it as such. On a frozen sweep of two mechanisms, two tie modes, five context depths, seven zero-gap rates and three seeds, an ODE-RNN was compared against its own de-flowed ablation, the same module with the flow never applied at

Table 6: Data cards: properties of the registered build, not of any model. Revision depth is the median releases per $(\text{series}, \tau)$, ALFRED's over its development window (correction `ca43d240`). $R = \mathbb{E}|z_{\text{settled}} - z_{\text{first}}|/\text{sd}(z_{\text{settled}})$, train only.

| | **ALFRED** | **Delphi** | **RKI** |
|---|---|---|---|
| Availability rule | vintage $+1$d | issue $+1$d | snapshot $+1$d |
| Settle horizon | first release $+365$d | $+60$d | $+40$d |
| Series | 48 | 102 | 119 ($17 \times 7$) |
| Revision depth (median) | 6 (dev window) | 60 | 5 |
| $R$ (train only) | 0.068 | 0.367 | — |
| Development refs | 2020-01..2022-12 | 2022-07..2023-02 | 2022-10..2023-09 |
| Development origins | 1,705 | 24,315 | 43,435 |
| Test refs (SEALED) | 2023-01..2024-12 | 2023-03..2024-06 | 2023-10..2024-10 |

identical parameter count. Over 140 settings the paired premium was $+0.00017$ nMSE, with 129 settings indistinguishable from zero and 6 favourable against 5 unfavourable, where 7 false positives are expected at the 5% level. The same ODE-RNN beat a vanilla Transformer by $+0.0194$ nMSE, and the de-flowed variant retained $+0.0193$ of that margin. The separation is between model families, not between continuous and discrete time. This is synthetic, involves no archive in this paper and no held-out split, and is reported because it is the same misattribution the arms of Section 4 are built to prevent, arising in a second literature.

**Scoring.** The weighted interval score [5, 10] is the standard probabilistic metric in epidemic forecast hubs. Our score-comparability gate extends reliability work on pretrained forecasters [2]: one widely used forecaster carries a fixed nine-level quantile head that cannot express a 23-level hub grid, and its wrapper silently clamps the outer levels. The reporting triangle connects to the actuarial chain-ladder tradition [18].

### G.1 Hub comparison in detail

We compare the RKI model with 8 teams that filed nowcasts to a hospitalisation hub in real time, admitted by a written rule before any score existed. The competitiveness gate fails: on the common intersection our mean WIS is 834.4 against a team range of 23.7 to 109.6, significantly worse than 5 of 8 teams. We claim no win over any team. Those with nominally negative contrasts are exactly the teams whose own files carry extreme quantile values, none of those contrasts is significant, and under a post-hoc median diagnostic we are worse than every team. Two observed limitations complicate interpretation. Our predictive intervals are much narrower than the teams', which is an observed property of our forecasts. Our head is trained on first-availability origins and scored on the hub's forecast-date grid, which is an observed design difference between the two setups. We ran no controlled experiment isolating either, so neither is established as a cause of the gap, and we do not attribute the gap to any architectural property. The comparison is an evaluation-compatibility diagnostic rather than a like-for-like benchmark; the negative result stands regardless.

## H Data cards

Table 6 records properties of the registered build for each archive. They describe the data as built, not any model.

## I Result tables

Table 7 gives the development mechanism gate, Table 8 the baseline guard, Table 9 the full hub comparison and Table 10 the failed S2 FluSight gate.

## J The shuffle-guard correction in detail

The shuffle arm's run-time guard needed a correction (correction 03, Table 4), which we report in full because the diagnostic it replaces is a natural thing to write and is wrong. A pooled-correlation

Table 7: **Mechanism gate G12 (development evidence).** Per-series standardised MSE of the settled target, seed-averaged; lower is better. Values are in units of each series' own training-period standard deviation, so they are comparable *within* a row and must not be compared in absolute size across domains. Origin and series counts per domain are in Table 1. Arms as defined in §4. Contrasts are full−control (negative favours the authentic release-context values), seed-paired two-way block bootstrap with 5000 resamples over the blocks named in §4, Holm corrected within domain across the two contrasts; the interval is the 2.5–97.5 percentile of that bootstrap and a bootstrap value of exactly zero is reported as $< 10^{-3}$. "seeds" counts seeds favouring *full*; "LOO" counts leave-one-series-out reaggregations of the frozen per-origin contrasts, which do not retrain, keeping the full−null sign.

| | | | arm MSE | | | full − null | | | full − shuffle | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| domain | origins | series | full | null | shuffle | $\Delta$ | Holm $p$ | seeds | $\Delta$ | Holm $p$ | seeds | LOO | G12 |
| ALFRED | 1,705 | 48 | 3.417939 | 3.417546 | 3.417474 | $+3.92 \times 10^{-4}$ | 0.0244 | 1/5 | $+4.65 \times 10^{-4}$ | $< 10^{-3}$ | 0/5 | 0/48 | FAIL |
| Delphi | 24,315 | 102 | 0.068753 | 0.091210 | 0.080825 | $-2.25 \times 10^{-2}$ | $< 10^{-3}$ | 5/5 | $-1.21 \times 10^{-2}$ | $< 10^{-3}$ | 5/5 | 102/102 | PASS |
| RKI | 43,435 | 119 | 0.207174 | 0.240871 | 0.249784 | $-3.37 \times 10^{-2}$ | 0.0120 | 4/5 | $-4.26 \times 10^{-2}$ | 0.0024 | 5/5 | 119/119 | PASS |

Table 8: **Baseline guard G13a (development evidence).** Same metric and splits as Table 7. $M_0$ is plain persistence on the most recent available value; $M_1$ is a persistence-nested gradient-boosting model tuned by rolling validation inside the training period only. The gate asks whether *full* is not worse than $M_1$; $\Delta$ is full $- M_1$ and "seeds" counts seeds favouring *full* over $M_1$. The best arm in each row is in bold and named in the penultimate column: the mechanism gate and this guard answer different questions, and Delphi passes the former while still losing to persistence.

| domain | full | $M_1$ (GBM) | $M_0$ (persist.) | $\Delta$ | $p$ | seeds | best arm | G13a |
|---|---|---|---|---|---|---|---|---|
| ALFRED | 3.417939 | 3.437160 | **3.417933** | $-1.92 \times 10^{-2}$ | $< 10^{-3}$ | 5 | persistence | PASS |
| Delphi | 0.068753 | 0.067922 | **0.060862** | $+8.31 \times 10^{-4}$ | 0.90 | 3 | persistence | PASS (tied) |
| RKI | **0.207174** | 0.267961 | 0.553505 | $-6.08 \times 10^{-2}$ | 0.0012 | 5 | full | PASS |

bound of $0.15$ between the original and shuffled value channel aborted all five reporting-triangle runs at $+0.2484$. Decomposing the statistic explains it: the residual pooled correlation equals the between-series share of value variance, to the digit, in all three domains ($+0.0390$ vs. $3.8\%$; $+0.0735$ vs. $7.4\%$; $+0.2484$ vs. $24.8\%$), while the within-series correlation – the pairing the protocol requires to be destroyed – is $+0.0006$, $-0.0001$ and $+0.0000$. A within-series permutation leaves each series' own level intact by construction, so a pooled bound penalises heterogeneous series, not a defective shuffle. The guard now asserts $|\text{within-series corr}| < 0.02$ and prints the decomposition as a diagnostic rather than a pass condition; no artifact was invalidated, and only the five blocked runs were re-executed with the same seeds.

# K Reproducibility

**Code.** The benchmark build, the model, the arms, the run-time assertions, the gate scripts, the WIS V3 scorer, the training-label availability audit and the table generators accompany this submission as anonymised supplementary material, with an environment file and run instructions. Two fixtures run offline, read no split and report no performance number: the persistence and gate-off assertions of Section 4, and the WIS anchors of Appendix F, which recover a score of exactly 1 on the $y = 1$ all-zero case on both the 23-level and 7-level grids. No result number in the paper, in its tables or in its figures is typed by hand: each is read at build time from a frozen result record, and each generated file carries the digests of the records it was made from in its own header.

**Data.** All three domains are built from public sources [8, 9, 21] by adapters released with the code; the adapters take the raw vintage archive, issue-dated API responses and daily snapshot files and emit the release-event tables the contract defines. Availability timestamps come from the sources' own vintage, issue and snapshot dates under the uniform $D{+}1$ rule of Section 3, so the build is reproducible from a clean download without any private artifact.

**Why the comparator files are not shipped.** The competitiveness layer scores the nowcasts 8 teams filed in real time to the German hospitalisation nowcast hub [12, 24]. Those submissions are read *in place* from a user-performed clone of the hub's public repository and are not redistributed

Table 9: **Competitiveness against admitted real-time hub submissions, G13b (development evidence).** Weighted interval score, lower is better; model scores are averaged over the 5 seeds. Comparators are the submissions the teams filed in real time, read only from the hub's `data-processed` directory; no team was added or removed on the basis of its score, and we redistribute no submission file. **Panel A** scores every system on the common intersection of all admitted teams and the model ($n = 19{,}691$ keys). **Panel B** is the pre-registered seed-paired contrast on each team's own overlap; $\Delta$ is model $-$ team, Holm corrected across teams, "seeds" counts seeds favouring the model. The model is significantly worse than 5 of 8 teams and *no win over any team is claimed*: the teams whose $\Delta$ is negative are exactly the teams whose own submitted quantiles contain extreme values, and Panel C shows that a handful of rows carries nearly the whole of those teams' means. **Panel C** is a post-hoc robustness diagnostic, *not* a pre-registered criterion: "flagged" rows are the extreme-value rows recorded in the frozen diagnostic. Under the robust view the reading is unchanged, the model being worse than every team by its median as well.

**Panel A.** Common intersection, $n = 19{,}691$; mean WIS.

| system | mean WIS |
|---|---|
| LMU_StaBLab-GAM_nowcast | 23.7 |
| NowcastHub-MedianEnsemble | 24.1 |
| NowcastHub-MeanEnsemble | 25.8 |
| SU-hier_bayes | 27.0 |
| ILM-prop | 34.5 |
| KIT-simple_nowcast | 37.2 |
| RIVM-KEW | 42.2 |
| SZ-hosp_nowcast | 109.6 |
| **as-of event model (ours, dev)** | **834.4** |

**Panel B.** Per-team overlap, seed-paired (pre-registered).

| team | $n$ | model | team | $\Delta$ | Holm $p$ | seeds | reading |
|---|---|---|---|---|---|---|---|
| ILM-prop | 46,487 | 633.2 | 26.0 | 607.2 | $< 10^{-3}$ | 0/5 | model worse |
| SU-hier_bayes | 83,375 | 357.0 | 15.2 | 341.8 | $< 10^{-3}$ | 0/5 | model worse |
| SZ-hosp_nowcast | 147,407 | 320.0 | 42.7 | 277.3 | $< 10^{-3}$ | 0/5 | model worse |
| NowcastHub-MedianEnsemble | 182,758 | 268.8 | 9.9 | 259.0 | $< 10^{-3}$ | 0/5 | model worse |
| KIT-simple_nowcast | 206,770 | 251.3 | 11.5 | 239.8 | $< 10^{-3}$ | 0/5 | model worse |
| NowcastHub-MeanEnsemble | 182,758 | 268.8 | $8.76 \times 10^{8}$ | $-8.76 \times 10^{8}$ | 0.78 | 5/5 | n.s. |
| RIVM-KEW | 193,407 | 266.0 | 1,422.8 | $-1{,}156.8$ | 1.00 | 5/5 | n.s. |
| LMU_StaBLab-GAM_nowcast | 165,938 | 279.1 | $2.90 \times 10^{9}$ | $-2.90 \times 10^{9}$ | 1.00 | 5/5 | n.s. |

**Panel C.** Post-hoc robustness diagnostic (not a criterion).

| team | team median | model median | flagged rows | share of keys | share of team mean |
|---|---|---|---|---|---|
| ILM-prop | 5.0 | 136.8 | 2 | 0.0043% | 0.5% |
| SU-hier_bayes | 3.3 | 94.8 | 3 | 0.0036% | 0.7% |
| SZ-hosp_nowcast | 10.5 | 83.6 | 28 | 0.0190% | 1.3% |
| NowcastHub-MedianEnsemble | 2.4 | 60.0 | 5 | 0.0027% | 0.8% |
| KIT-simple_nowcast | 2.3 | 54.7 | 1 | 0.0005% | 0.1% |
| NowcastHub-MeanEnsemble | 2.6 | 60.0 | 76 | 0.0416% | 100.0% |
| RIVM-KEW | 3.2 | 60.6 | 92 | 0.0476% | 98.9% |
| LMU_StaBLab-GAM_nowcast | 3.2 | 63.1 | 19 | 0.0115% | 100.0% |

here, for a stated reason: the repository contains no `LICENSE` file. Its README notes that the US, EU and German/Polish forecast hubs it draws from are MIT licensed, but makes no licence statement about itself, so we have no grant permitting redistribution and do not assume one. The hub also ships no scoring code; the weighted interval score used here is our own implementation of the published definition [5, 10], validated against the closed form on synthetic inputs before any comparator score was computed. We did not re-run any hub team's method and make no claim of having reproduced one. The comparison is therefore reproducible from the hub's own public archive together with our admission rule, key grid and scoring code, but not from a copy shipped with this paper.

**Environment and determinism.** The frozen manifest in Table 3 records the interpreter, library versions, random-number discipline and launch commands used for every arm. Arms within one

Table 10: **S2 FluSight development gate: selection-season scores, read once.** WIS over the 23 hub quantile levels, 5 seeds; the primary stratum $h = -1$ is the only one carrying a partial observation. Contrasts are full−control, seed-paired, bootstrapped over (forecast-date block $\times$ seed), Holm-corrected. Neither mechanism contrast meets the frozen $\alpha = 0.05$: verdict `DEV_FAIL_NO_TEST_OPENING`, sealed test unopened.

| layer | full | control | $\Delta$ | 95% CI | Holm $p$ | seeds | LOO / reading |
|---|---|---|---|---|---|---|---|
| *Mechanism gate (pre-registered):* FAIL | | | | | | | |
| WIS $h=-1$, vs. null | 0.5110 | 0.5440 | $-3.30 \times 10^{-2}$ | [-0.086, +0.010] | 0.3040 | 5/5 | 53/53 |
| WIS $h=-1$, vs. shuffle | 0.5110 | 0.5430 | $-3.20 \times 10^{-2}$ | [-0.088, +0.017] | 0.3040 | 4/5 | 53/53 |
| *Baseline guards* (point guard pre-registered; $p$ uncorrected as frozen) | | | | | | | |
| MSE $q_{0.5}$, vs. persistence | 10.2779 | 11.3009 | $-1.00 \times 10^{0}$ | — | 0.2384 | — | PASS |
| WIS all $h$, vs. mono. compl. | 1.4602 | 1.6242 | $-1.64 \times 10^{-1}$ | — | — | — | reported only |

seed slot are seeded identically and therefore start from identical weights, so they differ only in the context tensor; the shuffle permutation uses its own fixed generator, independent of the training seed. One seed runs entirely on one device and same-seed outputs are never merged across devices. Every per-origin artifact carries seed, device, library versions, and code, input and transform digests, and a completeness audit requiring exact key agreement across all arms and seeds runs before any aggregate is formed.

**Architectures considered and not run.** Continuous-time latent-state models for irregularly sampled series [22] are a natural alternative to the event-token encoder used here, and the gated residual pathway follows [3]. No architectural change was made after the reading reported in Section 5: substituting a model in response to a failed gate would convert a pre-registered test into a search. Such changes are deferred to a successor study with its own pre-registration and its own unread splits.

# L Use of LLMs and AI assistants

We follow the NeurIPS 2026 main-track policy. No LLM or agent is an author, and the author is responsible for all text, figures, claims and references. Beyond routine editing and basic code assistance, which the policy does not require us to disclose, assistants were used in two non-routine ways. They produced first drafts of protocol text, including candidate split boundaries and gate thresholds, all of which the author reviewed and froze before the corresponding reading. They were also used as adversarial readers of our own frozen documents, which is how the count errors in correction 04 of Table 4 surfaced. Two properties are mechanically checkable rather than matters of trust: every reported number is read at build time from a frozen record by a generator holding no result constants, and the gate conditions were fixed in writing before each reading. We do not claim every assistant-produced defect was caught: the scorer normalisation error of Appendix F passed three internal checks that shared one misreading and was caught only by an external fixture.

# M Ethics and broader impact

**Data.** All three domains are built entirely from public, aggregate, non-personal sources: a public macroeconomic vintage archive, a public epidemiological indicator API, and official public daily reporting-triangle snapshots. The comparator nowcasts are likewise public. No individual-level, personal, proprietary or restricted data were obtained, used or produced, and every source is reachable by any reader from its publisher. The comparator forecasts are the public real-time submissions of identified teams; they are read in place and are not redistributed, for the licence reason set out in Appendix K.

**Naming other teams.** Section 5 reports named teams' published nowcasts and, in one place, records that a small number of extreme values in three teams' own submitted quantile files dominate their mean under a non-robust statistic. That is stated as a property of a statistic applied to public files, not as a criticism of those teams, and it is stated precisely because ignoring it would have let us appear to

beat them. We claim no win over any team, and the one direction in which the arithmetic flattered us is the one we examined hardest.

**Use in public health and policy settings.** The system evaluated here is *not* suitable for deployment in surveillance or macroeconomic decision-making. Its intervals are too narrow for its own errors, and on the one domain where an external, operationally meaningful comparison exists it is worse than every real-time submission it was compared against. Reporting that clearly is part of the contribution: a benchmark whose failures are legible is more useful to a deployment decision than one whose headline is favourable. The positive result in this paper concerns whether authentic release-context values are used at all, which is not a licence to deploy.

**Environmental cost.** The development budget is held small: 5 seeds per arm per domain, three arms, three domains. That is a fixed budget rather than a power calculation, and it bounds what a null reading can mean. The ten-seed confirmation run was not authorised and was not executed.

# N Reproducibility statement

The evidence in this paper is a single reading of development splits that had never been read, followed by two held-out test readings under the successor S1-CONFIRM protocol. All of them ran under a protocol, a gate register, a split-and-seal table, a context rule, a comparator admission rule and a comparison grid frozen and hashed beforehand; Table 3 pins each of them, and Table 4 records the corrections issued against them. The build is reproducible from public sources through released adapters; the numbers are reproducible from the released result records through a generator that holds no result constants; the comparator layer is reproducible from the hub's own public archive together with our admission rule, key grid and scoring code, but not from any copy of their files shipped by us. Uncertainty is reported from a seed-paired two-way block bootstrap [16] with a Holm correction [11] across contrasts, over 5 seeds, which is a development budget and not a characterisation of seed variance [1, 4, 19]. The Delphi and FluSight test splits have status TEST_LABELS_MATERIALIZED_STRUCTURALLY_AUDITED_UNSCORED: their labels were separated and were not used for fitting, selection, calibration, prediction scoring or aggregate evaluation. We do not describe them as unread, because structural audits over files that contain those rows were run and are disclosed in Appendix B. The ALFRED and RKI test splits were opened for scoring once each on 2026-09-06 under S1-CONFIRM, whose opening records and results are cited in Section 5; each was scored exactly once, after per-origin artifacts were written and after an exact-key and finiteness audit.