ALPHA CONNECTOME

Intraday prediction with simple models

From the observed stream to a defensible prediction score.

Johannes Bleher

University of Hohenheim

28 October 2026

Can simple models use the stream well?

At each row, two linked books arrive.

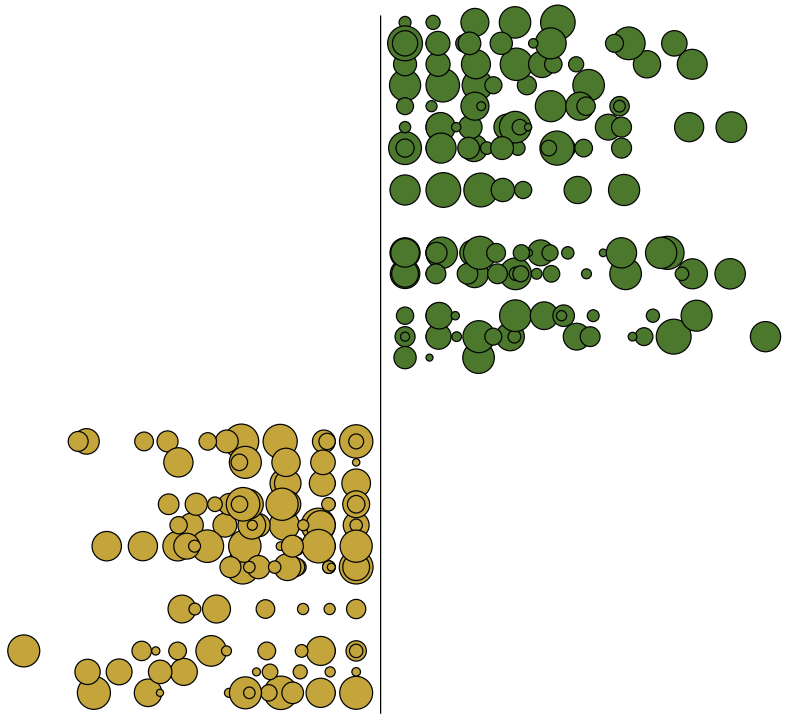
We must return two predictions using only information already available.

One question throughout the talk

Which summaries of the current row and its history improve the official prediction score?

Route: understand the row → define the score → build memory → test added structure.

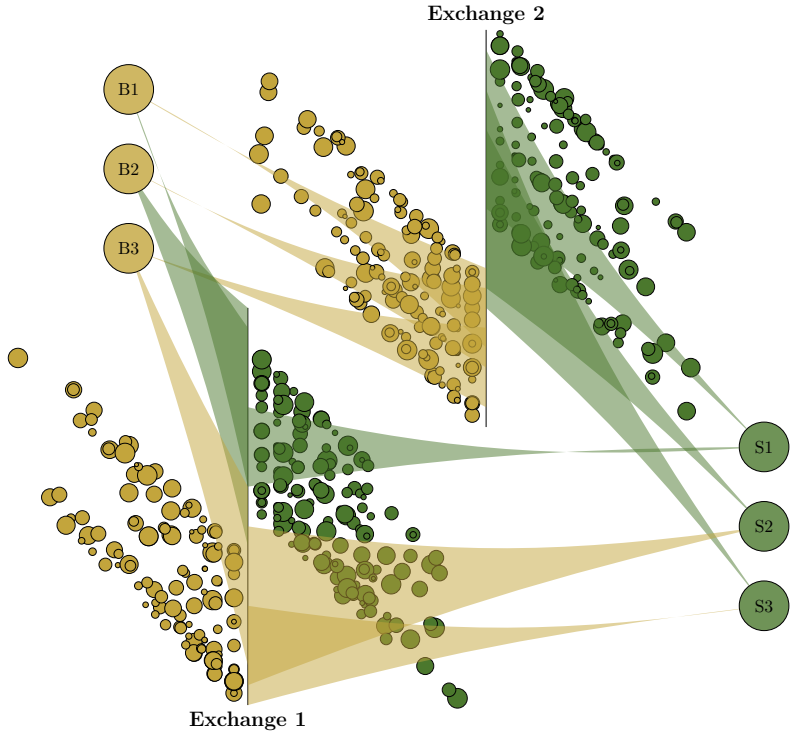
A book records buy and sell orders



- Yellow circles: buy orders, or bids.
- Green circles: sell orders, or asks.
- Prices increase up the vertical axis.
- Order sizes and cancellations affect available liquidity.

LOBAlgebra paper, Figure 1a; unchanged schematic, not challenge observations.

Related books can carry useful signals



- Shared participants can connect different books.
- Connectome provides `i0` and `i1`.
- 52 inputs per book + 8 additional inputs = 112.
- Both targets concern the future of `i0`.

LOBAlgebra paper, Figure 1d; conceptual multiple-book composition, not identified Connectome venues.

One row gives 112 inputs and two training labels

For sequence n and row t, write the supplied input vector as $\boldsymbol{x}_{n,t} \in \mathbb{R}^{112}$.

Input group, in supplied order	Coordinates
`i0`: 44 book-related + 8 trade-related	52
`i1`: 44 book-related + 8 trade-related	52
Additional anonymous features	8

$y_{n,t,k}$ is label `t0` or `t1`, with $k \in \{0, 1\}$.

Both labels describe future movement of `i0`; neither label is an input at inference.

Feature names identify groups, not physical depth

Within each book	Disclosed meaning
p0–p21	Price-like; indices 0–10 bid, 11–21 ask
v0–v21	Volume-like; same bid/ask grouping
dp0–dp3	Trade-price features
dv0–dv3	Trade-volume features

Units, depth order, elapsed time and target horizons are hidden.

For example, p0 is not identified as the best bid. Do not construct a quoted spread from arbitrary slots.

Process every row; reset at each sequence

- $t = 0, \ldots, 19{,}999$ orders rows within sequence n.
- Rows 0–98 warm up the model memory.
- Return $(\widehat{y}_{n,t,0}, \widehat{y}_{n,t,1})$ from row 99 onward.
- Update memory on unscored rows too; reset when n changes.

$\widehat{y}_{n,t,k}$ denotes our prediction of $y_{n,t,k}$. Sequence IDs identify boundaries, not dates.

Each prediction uses the same available history

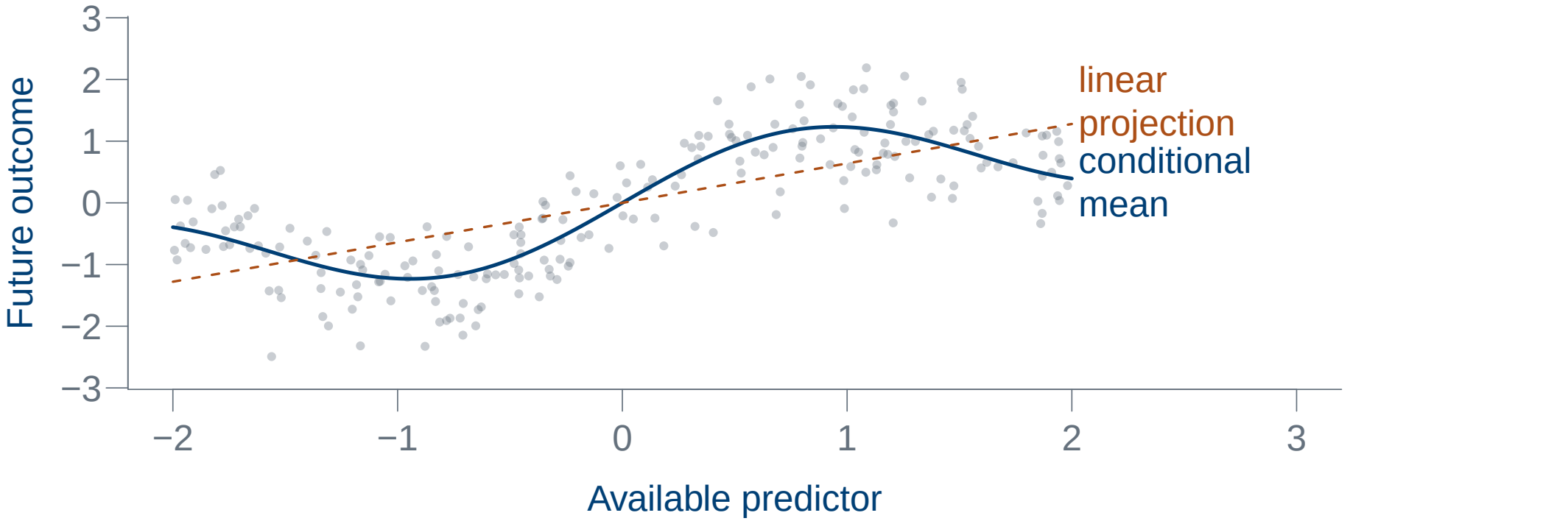
Use the current row and earlier inputs from this sequence:

$$\mathcal{I}_{n,t} = \sigma(\boldsymbol{x}_{n,0}, \ldots, \boldsymbol{x}_{n,t}).$$

- Fit transformations and coefficients on training data; keep them fixed for prediction.
- No future row, target label, or hidden test mask enters inference.

From here, suppress n within one sequence: $\boldsymbol{x}_t$, $y_{t,k}$, $\hat{y}_{t,k}$ and $\mathcal{I}_t$ keep these meanings.

A useful prediction summarizes the expected outcome



Synthetic example. $m_k(\mathcal{I}_t) = \mathrm{E}[y_{t,k} \mid \mathcal{I}_t]$: conditional mean, the squared-error optimum. Global WP uses a different criterion.

Predict every required row; score selected rows

Let $\mathcal{M}$ contain all evaluated pairs (n, t) with both `need_prediction` and `is_scored` true.

- Pool these rows across all sequences before computing the score.
- The validation mask is supplied for evaluation.
- The hidden test mask is applied after inference.

Compare methods on the same $\mathcal{M}$. Averaging sequence-level correlations gives a different metric.

Global WP rewards weighted co-movement

On scored rows, clip the label and our prediction to $[-2, 2]$:

$$c_{t,k} = \mathrm{clip}(y_{t,k}, -2, 2), \qquad p_{t,k} = \mathrm{clip}(\widehat{y}_{t,k}, -2, 2), \qquad w_{t,k} = |c_{t,k}|.$$

Compute the weighted Pearson correlation $\rho_{k,w}$ for each target.

$$s_{\mathrm{WP}} = \tfrac{1}{2}(\rho_{0,w} + \rho_{1,w}).$$

Larger clipped label movements receive more weight. Higher is better; the score lies in $[-1, 1]$.

Scale inputs once before building features

For input coordinate $j = 1, \ldots, 112$, define

$$z_{t,j} = \operatorname{clip}\left(\frac{x_{t,j} - \widehat{\mu}_j}{\widehat{\sigma}_j}, -8, 8\right).$$

- $\widehat{\mu}_j$ and $\widehat{\sigma}_j > 0$: fitting-sample center and scale.
- $\boldsymbol{z}_t$ always denotes these 112 transformed inputs.
- A first regression is $\widehat{y}_{t,k} = \widehat{a}_k + \widehat{\boldsymbol{b}}_k' \boldsymbol{z}_t$.

Keep training preprocessing fixed. Input clip: 8; score clip: 2. $\hat{a}_k$: intercept; $\hat{\boldsymbol{b}}_k$: slopes; prime: transpose.

A moving average turns past inputs into features

For half-life ℓ rows, update the exponential moving average:

$$r_\ell = 2^{-1/\ell}, \qquad \boldsymbol{e}_{\ell,t} = r_\ell \boldsymbol{e}_{\ell,t-1} + (1 - r_\ell)\boldsymbol{z}_t.$$

Initialize $\boldsymbol{e}_{\ell,0} = \boldsymbol{z}_0$ at every sequence boundary.

$$\boldsymbol{d}_{\ell,t} = \boldsymbol{z}_t - \boldsymbol{e}_{\ell,t}, \qquad \mathcal{H} = \{2, 8, 32, 128, 512\}.$$

$\boldsymbol{d}_{\ell,t}$ measures current deviation from recent history. Half-life ℓ is in rows, not seconds or a target horizon.

Stack current inputs and five memory scales

Collect the features in the column vector

$$\boldsymbol{m}_t = (\boldsymbol{z}_t', \boldsymbol{d}_{2,t}', \boldsymbol{d}_{8,t}', \boldsymbol{d}_{32,t}', \boldsymbol{d}_{128,t}', \boldsymbol{d}_{512,t}')' \in \mathbb{R}^{672}.$$

$$\widehat{y}_{t,k} = \widehat{a}_k + \widehat{\boldsymbol{b}}_k' \boldsymbol{m}_t, \qquad k \in \{0, 1\}.$$

- 112 current inputs + 5×112 deviations = 672 slopes.
- Ordinary least squares (OLS) minimizes squared label errors on fitting rows.

Same information $\mathcal{I}_t$, richer summary. Refit coefficients for each target and feature basis.

Nonlinear features can keep a linear readout

Append 612 nonlinear features $\boldsymbol{r}_t$ to the memory basis:

$$\boldsymbol{\phi}_t = (\boldsymbol{m}_t', \boldsymbol{r}_t')' \in \mathbb{R}^{1284}, \qquad \widehat{y}_{t,k} = \widehat{a}_k + \widehat{\boldsymbol{b}}_k' \boldsymbol{\phi}_t.$$

- Per input: $|z_{t,j}|$, $z_{t,j}^2$, $z_{t,j}|z_{t,j}|$; 336 features.
- 52 paired-slot products and 224 deviations divided by recent input scales.
- Linear in fitted coefficients; nonlinear in the available inputs.

Refit coefficients for each basis. The appendix defines the products and recent-scale normalization.

Choose features and fitting loss separately

Estimator	What fitting emphasizes
OLS	Minimize squared errors; raw or clipped labels are separate candidates
Weighted LS	Weight squared clipped-label errors by $w_{t,k} = \lvert c_{t,k} \rvert$
Ridge	Add a squared-slope penalty to stabilize the fitted coefficients

Choose the basis, loss and penalty on development data using s_{WP}.

These fitting objectives are surrogates. None automatically maximizes clipped weighted correlation.

Stable distributions and representative averages differ

- **Stationarity:** shifting all time indices leaves the joint distribution unchanged.
- **Ergodicity:** under a stationary law, long-time averages recover population averages.
- A stable fitted VAR or generator does not establish either property for market data.
- Use causal history/scale features, then assess transfer across separate sequences.

We now audit 64 existing fitting sequences; formal definitions, measured diagnostics and remedies are in the appendix.

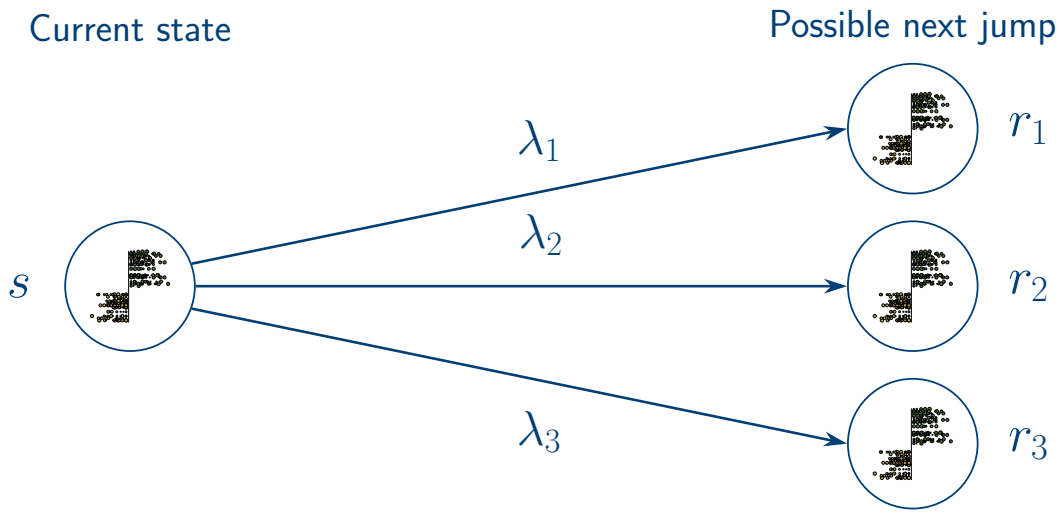
Fix choices before judging added structure

Fitting sequences	Learn scaling, representations and coefficients
Development sequences	Choose features, penalties and corrections
Assessment	Test on reserved sequences

Compare paired predictions on the same rows with s_{WP}.

95% intervals below resample whole sequences and recompute pooled score differences; they condition on fitted and selected models.

A state model offers another summary of history



Schematic jump destinations; arrows show rates, not probabilities.

- A regime summarizes current book conditions.
- Transition rates describe changes between regimes.
- Filtering updates regime probabilities using the new inputs.

Schematic: s, r_1, r_2, r_3 are states; $\lambda_1, \lambda_2, \lambda_3$ are jump rates, not probabilities.

Two four-regime books give 16 joint states

- Learn four statistical regimes per instrument from fitting inputs.
- Let the other book's regime affect each transition rate.
- Update 16 state probabilities using current inputs and past probabilities.
- Use these probabilities and input interactions as regression features.

Same targets, same Global WP comparison.

A predictive adaptation of LOBAlgebra; anonymous features do not identify literal order-arrival rates.

Better input forecasts need not improve target forecasts

- Keep $\boldsymbol{z}_t$ continuous and forecast its levels and squares.
- Compare current-row and history-based maps at lags $h = 1, 8, 32$ rows.
- Measure input error by mean squared error (MSE), scaled by fitting-sample increment variation; lower is better.
- Test past forecast surprises as extra target-regression features.

Input MSE and target s_{WP} answer different questions. Target definitions and physical horizons remain hidden.

The selected correction uses plain input lags

- Compare $\boldsymbol{z}_t$ with $\boldsymbol{z}_{t-h}$ at lags $h = 1, 8, 32$ rows.
- Use these differences, scale-normalized differences and log scales as extra features.
- Fit OLS to the richer regression's remaining label errors on fitting data.
- Add one quarter of that predicted error; development selects this shrinkage for both targets.

Unchanged weighted-ridge base + OLS residual correction. The final predictor is not a plain OLS model.

Rich regression is the benchmark to beat

Procedure	Global WP s_{WP}
Supplied recurrent neural network (GRU; reported)	0.6171
Previous 672-feature regression	0.6217
1,284-feature regression, raw-label OLS	0.6337
Selected richer regression (weighted ridge/blend)	0.6446
Richer regression + state correction	0.6454
Richer regression + selected lag correction	**0.6458**

Reused full validation: 1,873 sequences, same official mask. GRU is organizer-reported; local methods were evaluated here.

The fresh assessment leaves the extra gain uncertain

The selected lag correction adds **0.0012** Global WP on reused validation.

On 64 new training-file sequences, its paired gain is

$$\Delta s_{\mathrm{WP}} = 0.00135, \qquad \text{95\% interval } [-0.00022,\ 0.00296].$$

- This interval includes zero; a gain is not established there.
- Better input forecasts did not win the target development comparison.

Δs_{WP}: correction minus richer regression on the same rows. New TRAIN assessment uses all required rows.

Turn the benchmark into a reproducible workflow

- Inspect metadata and one complete sequence first.
- Use `01_data`, numbered scripts in `02_code`, and saved results in `04_output`.
- Ask for one bounded change and an explicit check.
- Read the code, results, and unresolved assumptions.
- Keep an execution log so another person can reproduce the run.

The README lists every numbered preparation, fitting, selection, and evaluation stage.

Fit in R, verify the same predictor in Python

```
X <- cbind(1, causal_features)
fit <- lm.fit(x = X, y = targets)
B <- fit$coefficients
```

- Build every history feature in sequence order.
- Export feature order, transformations, and coefficients.
- Check R/Python prediction parity and sequence resets.

Plain OLS example; the implemented weighted/ridge candidates are separate estimators.

Package, test, then submit

- Put `solution.py` and its model files at the ZIP root.
- Return two finite outputs after warm-up; reset every sequence.
- Test deterministic inference on the complete callback.
- Budget: one CPU, 16 GB RAM, and 60 minutes.
- ZIP size at most 20 MB; retain scored candidates for final selection.

No internet is available during inference; five submissions per day.

Keep two deadlines in mind

- **Team changes:** 1 November 2026, 23:59 UTC.
- **Submissions:** 15 November 2026, 23:59 UTC.
- You can enter individually; teams have at most five members.
- Check announcements and the required final-selection procedure.

Official timeline; Berlin deadlines fall just after midnight the next day.

Let added structure earn its place.

- Start with rich features from available history.
- Fit, score and compare the same prediction task.
- Keep a separate assessment for the chosen model.

wundernn.io/connectome/docs/quick_start

Use the same clipping and weights in every score

For $(n,t) \in \mathcal{M}$ and target k, keep $c = \mathrm{clip}(y,-2,2)$, $p = \mathrm{clip}(\widehat{y},-2,2)$ and $w = |c|$. Define total weight ω_k and the weighted means:

$$\omega_k = \sum_{(n,t)\in\mathcal{M}} w_{n,t,k}, \quad \bar{c}_{k,w} = \frac{\sum_{(n,t)\in\mathcal{M}} w_{n,t,k} c_{n,t,k}}{\omega_k}, \quad \bar{p}_{k,w} = \frac{\sum_{(n,t)\in\mathcal{M}} w_{n,t,k} p_{n,t,k}}{\omega_k}.$$

$$\rho_{k,w} = \frac{\sum_{(n,t)\in\mathcal{M}} w_{n,t,k}\left(c_{n,t,k} - \bar{c}_{k,w}\right)\left(p_{n,t,k} - \bar{p}_{k,w}\right)}{\sqrt{\left(\sum_{(n,t)\in\mathcal{M}} w_{n,t,k}\left(c_{n,t,k} - \bar{c}_{k,w}\right)^2\right)\left(\sum_{(n,t)\in\mathcal{M}} w_{n,t,k}\left(p_{n,t,k} - \bar{p}_{k,w}\right)^2\right)}}.$$

$$s_{\mathrm{WP}} = \tfrac{1}{2}\left(\rho_{0,w} + \rho_{1,w}\right).$$

Pooled rows, separate targets. Float64 arithmetic; official zero-weight/zero-variance rules apply.

Weighted squared loss targets a different conditional mean

Fix a target k and a row t; its clipped label is $c_{t,k}$ and its weight is $w_{t,k} = |c_{t,k}|$. The best weighted squared-loss prediction is

$$a^*_{t,k} = \arg\min_a \mathrm{E}[w_{t,k}(c_{t,k} - a)^2 \mid \mathcal{I}_t] = \frac{\mathrm{E}[|c_{t,k}| c_{t,k} \mid \mathcal{I}_t]}{\mathrm{E}[|c_{t,k}| \mid \mathcal{I}_t]},$$

when the denominator is positive.

It generally differs from the unweighted mean $\mathrm{E}[c_{t,k} \mid \mathcal{I}_t]$.

A fitting-loss optimum need not maximize the clipped-correlation score.

Separate outcome noise from variability in the fitted rule

For a fixed feature vector $\boldsymbol{\phi}$, let $\mu_k(\boldsymbol{\phi}) = \mathrm{E}[y_{t,k} \mid \boldsymbol{\phi}_t = \boldsymbol{\phi}]$. Let $\mathcal{D}$ denote a random fitting sample and $\widehat{f}_{\mathcal{D},k}$ its fitted prediction rule.

$$\begin{aligned}\mathrm{E}_{\mathcal{D},y}[(y_{t,k} - \widehat{f}_{\mathcal{D},k}(\boldsymbol{\phi}))^2 \mid \boldsymbol{\phi}_t = \boldsymbol{\phi}] = \mathrm{Var}[y_{t,k} \mid \boldsymbol{\phi}_t = \boldsymbol{\phi}] \\ + \left(\mathrm{E}_{\mathcal{D}}[\widehat{f}_{\mathcal{D},k}(\boldsymbol{\phi})] - \mu_k(\boldsymbol{\phi})\right)^2 \\ + \mathrm{Var}_{\mathcal{D}}[\widehat{f}_{\mathcal{D},k}(\boldsymbol{\phi})].\end{aligned}$$

Outcome noise + squared bias + fitting variance. Requires test outcome independent of fitting data given the feature vector.

Fixing a model and testing it are different steps

- Training data estimate transformations and coefficients.
- Development data choose features, penalties and model variants.
- Fix all these choices before a separate assessment.
- A reused development score remains a development result.
- New sequences test transfer; shuffled IDs do not establish a future-date test.

Our full-validation benchmark includes reused development. The new TRAIN assessment was excluded from fitting and selection.

Hold out sequences before judging predictive value

FIT + TUNE WITH GROUPED FOLDS

sequence 72 | sequence 8 | sequence 105 | sequence 31

HOLD OUT COMPLETE SEQUENCES

sequence 94 | sequence 16 | sequence 3 | sequence 51

Order rows within each sequence; do not sort sequences by their ID.

Illustrative sequence IDs; they are categorical identifiers, not dates.

A credible comparison includes stability and runtime

- Fix all model choices before comparing on separate assessment sequences.
- Report both target correlations as well as their mean.
- Resample whole sequences and recompute the global score for a dependence-aware diagnostic.
- Examine additional moments to detect validation-mask specialization.
- Benchmark the full callback loop on one CPU; include all state updates.

A private leaderboard remains an external test of generalization.

A sequence bootstrap relies on the sampling interpretation of sequences; it cannot establish robustness to every future regime.

An exported model includes its transformation contract

Save	Verify in both languages
Feature names and order	Identical positions and numeric component order
Training preprocessing	Identical centering, scaling and missing-value rules
Intercept and coefficients	Same raw-scale prediction for each target
Dynamic state rules	Identical initialization, lags and reset rules
Known input/output fixture	Predictions agree within a documented tolerance

Compare full sequential predictions, not just a coefficient file.

Plain coefficient and scaling files can keep linear-model inference small and transparent.

The richer basis adds shapes and paired-slot products

For the same transformed inputs $\boldsymbol{z}_t$, define the 52-vector of products

$$\chi_{t,j} = z_{t,j} z_{t,j+52}, \qquad j = 1, \ldots, 52.$$

Append shapes, products and two scaled-memory blocks:

$$\boldsymbol{r}_t = (|\boldsymbol{z}_t|', \ (\boldsymbol{z}_t^{\odot 2})', \ (\boldsymbol{z}_t \odot |\boldsymbol{z}_t|)', \ \boldsymbol{\chi}_t', \ (\boldsymbol{d}_{8,t}^{\mathrm{sc}})', \ (\boldsymbol{d}_{128,t}^{\mathrm{sc}})')'.$$

336 shapes + 52 products + 224 scaled deviations = 612 additions. Products pair corresponding slots across books; no causal interpretation is assumed.

Scale a memory deviation by recent input variability

Keep the earlier mean $\boldsymbol{e}_{\ell,t}$ and decay r_ℓ, with $\ell \in \{8, 128\}$.

Update a second-moment average, starting at $\boldsymbol{e}^{(2)}_{\ell,0} = \boldsymbol{z}_0^{\odot 2}$:

$$\boldsymbol{e}^{(2)}_{\ell,t} = r_\ell \boldsymbol{e}^{(2)}_{\ell,t-1} + (1 - r_\ell)\boldsymbol{z}_t^{\odot 2}, \qquad \boldsymbol{\sigma}^2_{\ell,t} = \max(\boldsymbol{e}^{(2)}_{\ell,t} - \boldsymbol{e}^{\odot 2}_{\ell,t}, 0).$$

Then normalize the already defined deviation $\boldsymbol{d}_{\ell,t}$:

$$\boldsymbol{d}^{\mathrm{sc}}_{\ell,t} = \mathrm{clip}\Big(\boldsymbol{d}_{\ell,t} \oslash \sqrt{\boldsymbol{\sigma}^2_{\ell,t} + 0.01}, -8, 8\Big).$$

$\boldsymbol{\sigma}^2_{\ell,t}$ measures recent input variability. All operations are coordinatewise; 0.01 stabilizes division. No target labels enter.

Stationarity means invariance under time shifts

Let $\boldsymbol{W}_t = (\boldsymbol{x}_t', y_{t,0}, y_{t,1})'$ collect inputs and targets.

Strict stationarity: for every finite collection of times and shift h,

$$(\boldsymbol{W}_{t_1}, \ldots, \boldsymbol{W}_{t_m}) \overset{d}{=} (\boldsymbol{W}_{t_1+h}, \ldots, \boldsymbol{W}_{t_m+h}).$$

Weak stationarity: finite second moments and

$$\mathrm{E}[\boldsymbol{W}_t] = \boldsymbol{\mu}_W, \qquad \mathrm{Cov}(\boldsymbol{W}_t, \boldsymbol{W}_{t-h}) = \boldsymbol{\Gamma}_W(h).$$

- The mean and covariance do not depend on the time origin.
- Standardizing or clipping inputs does not guarantee stationarity.

Breiman, Probability, chapter 6. Indices describe rows within one sequence.

Ergodicity lets one path represent its population

Assume a stationary law.

Events unchanged by shifting the path have probability zero or one.

For any integrable function g of a finite window,

$$\frac{1}{T}\sum_{t=1}^{T} g(\boldsymbol{W}_t, \ldots, \boldsymbol{W}_{t+m}) \xrightarrow{\text{a.s.}} \mathrm{E}[g(\boldsymbol{W}_0, \ldots, \boldsymbol{W}_m)].$$

- A long dependent sequence can still have representative averages.
- Stationarity alone is insufficient; persistent hidden regimes can prevent this convergence.
- Ergodicity alone does not justify a central limit theorem or our bootstrap intervals.

Integrability means $\mathrm{E}\,|g| < \infty$; a.s. means almost surely. Mixing and cross-sequence sampling need separate assumptions.

A persistent sequence effect survives a long average

Draw one random level U per sequence, independently of iid noise ε_t with mean zero:

$$V_t = U + \varepsilon_t, \qquad \mathrm{Var}(U) > 0.$$

$$\frac{1}{T}\sum_{t=1}^{T} V_t \xrightarrow{\text{a.s.}} U, \qquad \mathrm{E}[V_t] = \mathrm{E}[U].$$

- This process is stationary, but one path retains its random level.
- More rows cannot replace observing more independently drawn regimes.
- Pooling many suitable sequences can still estimate the mixture law.

An illustrative counterexample; no claim that this is the Connectome data-generating process.

OLS can average moments across sequences

Let $\boldsymbol{\Phi}_n$ contain m fitting rows, including an intercept, and $\boldsymbol{Y}_n$ their two labels.

$$\boldsymbol{G}_n = \frac{1}{m}\boldsymbol{\Phi}_n'\boldsymbol{\Phi}_n, \qquad \boldsymbol{H}_n = \frac{1}{m}\boldsymbol{\Phi}_n'\boldsymbol{Y}_n, \qquad \widehat{\boldsymbol{B}} = \left(\sum_{n=1}^{N} \boldsymbol{G}_n\right)^{+} \sum_{n=1}^{N} \boldsymbol{H}_n.$$

- Within-sequence rows may be strongly dependent.
- A sequence-level law of large numbers can support pooled estimation.
- It still needs suitable cross-sequence dependence, moments and an identifiable feature span.
- Transfer to the test population is an additional requirement.

N: sequences; $+$: generalized inverse. Whole-sequence resampling preserves internal dependence, not unknown dependence between sequences.

Local features reduce moment variation, not all persistence

Representation	Mean range / SD	Variance ratio	Lag128 corr.
Clipped levels	0.415	2.336	0.174
EMA128 deviations	0.060	2.308	0.008
Scaled deviations	0.066	1.206	0.009
Raw targets	0.490	2.600	0.403
Regression residuals	0.490	2.579	0.369

- Four contiguous blocks per path; table entries are medians.
- Labels and residuals remain persistent; the cause is not identified.

64 existing fitting sequences; required rows only. Descriptive checks, not stationarity or ergodicity hypothesis tests.

Solve the invariant law of each fixed generator

$$\boldsymbol{\pi}_\star \boldsymbol{Q} = \mathbf{0}', \qquad \boldsymbol{\pi}_\star \mathbf{1} = 1, \qquad \boldsymbol{\mu}(t) = \boldsymbol{\mu}(0)\exp(t\boldsymbol{Q}).$$

$$d(t) = \max_i \frac{1}{2}\sum_j \left|[\exp(t\boldsymbol{Q})]_{ij} - \pi_{\star,j}\right|, \qquad \tau_{.05} = \inf\{t : d(t) \le .05\}.$$

Fixed fitted generator	$\tau_{.05}$ in row units
Earlier coupled model	34.1
Refitted static model	27.4
Four fixed clock contexts	23.5–54.4

Fixed-chain calculations: solve a constrained linear system, then bracket/bisect the distance. Lalley: finite-chain results. No filter warm-up guarantee.

Adapt input features while retaining local context

Use past-only location $\boldsymbol{m}_{t-1}$ and second moment $\boldsymbol{s}_{t-1}$:

$$\boldsymbol{u}_t = (\boldsymbol{z}_t - \boldsymbol{m}_{t-1}) \oslash \sqrt{\max(\boldsymbol{s}_{t-1} - \boldsymbol{m}_{t-1}^{\odot 2}, 0) + \epsilon}.$$

For input-only regime weights $\omega_{r,t} \geq 0$, $\sum_r \omega_{r,t} = 1$, fit

$$\widehat{y}_{t,k} = \sum_r \omega_{r,t}(a_{r,k} + \boldsymbol{b}'_{r,k}\boldsymbol{u}_t) + \boldsymbol{c}'_k \boldsymbol{m}_{t-1}.$$

- Update local moments and gates from inputs; fit coefficients on training labels.
- Existing models already use EMA deviations and variability scaling.
- This combined specification is a next candidate, not a measured new WP gain.

OLS still fits linear coefficients on known transformed features. Local normalization mitigates shifts; it does not guarantee stationarity.

An exact generator need not describe a stationary LOB

$$\boldsymbol{Q}_t = \kappa_{0,t-1}\boldsymbol{Q}_0 + \kappa_{1,t-1}\boldsymbol{Q}_1, \qquad \kappa_{j,t-1} > 0.$$

- A generator describes evolution; stationarity additionally concerns its law and initialization.
- Existing positive activity clocks let past inputs change transition speeds.
- Add relevant state/context when one fixed transition law is inadequate.
- Each fixed context's invariant law does not establish stationarity of the switched process.

The activity-clock model is implemented; it did not establish a clear target-score advantage over simpler state gating.

A joint factor/target VAR must infer its target coordinates

Let $\boldsymbol{f}_t^{\mathrm{VAR}} \in \mathbb{R}^4$ be four PCA input factors, with loadings learned on fitting data. Define the six-dimensional joint state

$$\boldsymbol{u}_t = ((\boldsymbol{f}_t^{\mathrm{VAR}})', y_{t,0}, y_{t,1})', \qquad \boldsymbol{u}_t = \boldsymbol{c} + \boldsymbol{A}\boldsymbol{u}_{t-1} + \boldsymbol{\varepsilon}_t.$$

Here $\boldsymbol{c}$ is an intercept, $\boldsymbol{A}$ a 6×6 lag matrix, and $\boldsymbol{\varepsilon}_t$ a mean-zero innovation.

At inference only the four factor coordinates are observed.

Vector autoregression, one row lag: VAR(1). Fit true adjacent pairs; reset the input-only filter at each sequence.

The filter updates the joint state using input factors only

Observe $\boldsymbol{f}_t^{\mathrm{VAR}} = \boldsymbol{H}\boldsymbol{u}_t$ with $\boldsymbol{H} = [\boldsymbol{I}_4 \ \boldsymbol{0}_{4\times 2}]$. With posterior mean $\widehat{\boldsymbol{u}}_{t-1|t-1}$ and covariance $\boldsymbol{C}_{t-1}$,

$$\widehat{\boldsymbol{u}}_{t|t-1} = \boldsymbol{c} + \boldsymbol{A}\widehat{\boldsymbol{u}}_{t-1|t-1}, \qquad \boldsymbol{V}_t = \boldsymbol{A}\boldsymbol{C}_{t-1}\boldsymbol{A}' + \boldsymbol{\Sigma}_\varepsilon.$$

The innovation covariance is $\boldsymbol{\Sigma}_\varepsilon = \mathrm{Var}[\boldsymbol{\varepsilon}_t]$; the update gain is

$$\boldsymbol{K}_t = \boldsymbol{V}_t\boldsymbol{H}'(\boldsymbol{H}\boldsymbol{V}_t\boldsymbol{H}')^{-1}, \qquad \widehat{\boldsymbol{u}}_{t|t} = \widehat{\boldsymbol{u}}_{t|t-1} + \boldsymbol{K}_t(\boldsymbol{f}_t^{\mathrm{VAR}} - \boldsymbol{H}\widehat{\boldsymbol{u}}_{t|t-1}).$$

$$\boldsymbol{C}_t = \boldsymbol{V}_t - \boldsymbol{K}_t\boldsymbol{H}\boldsymbol{V}_t.$$

$t|t-1$: before seeing row t; $t|t$: after seeing it. The last two posterior-mean coordinates predict $y_{t,0}, y_{t,1}$.

Fit in R, then score the same procedure in Python

```
# current and lagged contain true adjacent TRAINING pairs.
model <- fit_simple_var(
  current, lagged, feature_columns, k = 4
)
# Export coefficients, scaling, covariance and filter gains.
jsonlite::write_json(model, "simple_var.json", digits = 16)
```

64 fit sequences; 16 separate development sequences; 64 validation sequences.

The four factors retain 33.8% of standardized input variation.

Numbered stages 06–09 prepare, fit, evaluate and report; validation never selects this specification.

The small VAR pilot does not set the performance ceiling

Reference	Global WP	Evaluation sample
Our filtered VAR(1)	0.2552	64 validation sequences
Supplied GRU	0.6171	Full validation; organizer-reported
Public leaders	About 0.67	Hidden public test data

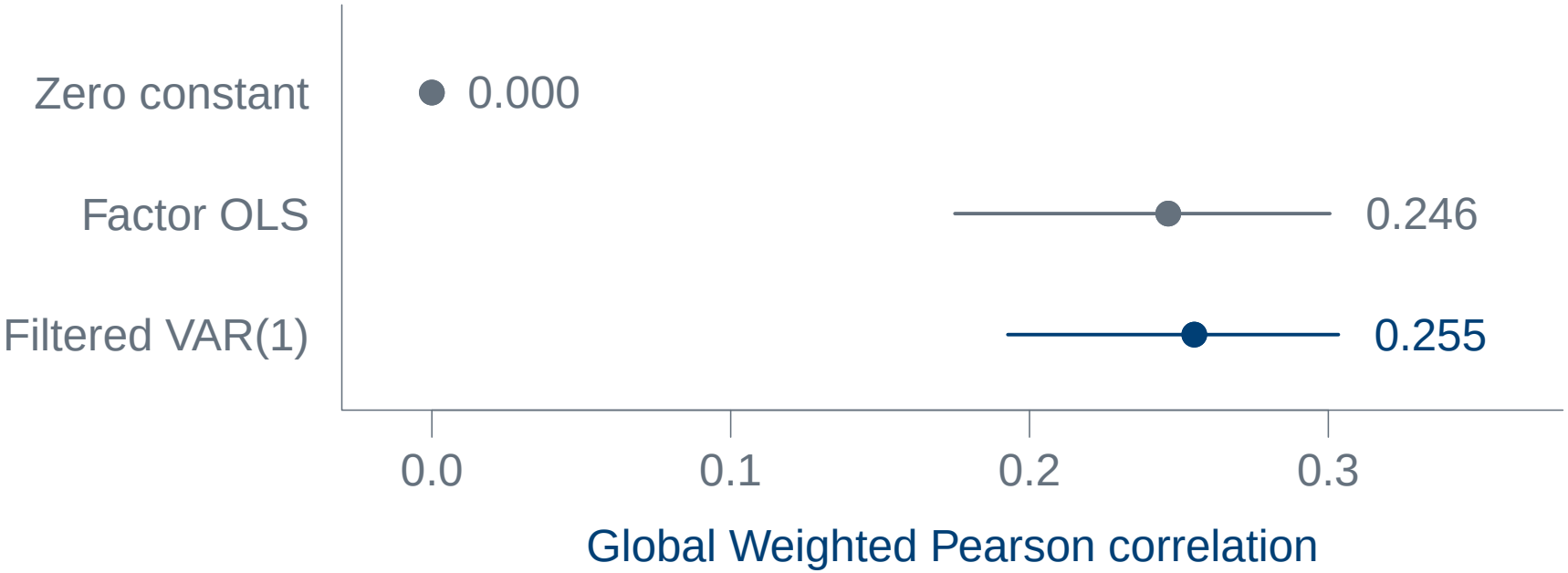
Only four factors and one lag have been fitted in our first example.

Later benchmarks test richer memory and nonlinear features.

Scores on different samples provide context.

Official public leaderboard; supplied baseline README | checked 27 October 2026

The paired score difference remains uncertain



Measured 64-sequence validation subset; bars are 95% sequence-bootstrap intervals conditional on the fitted models.

LOBAlgebra supplies a richer book-dynamics framework

Model structure

Placements, cancellations and matching.

Partial fills and interacting book states.

State-dependent, cross-book event rates.

Required observation map

Relate latent book states to anonymized features.

Declare how target predictions are obtained.

Keep physical-rate claims separate from predictive fit.

The existing simulator expects a price grid, orders and event information.

Use the coupled-book structure; establish its mapping to the supplied data.

Johannes's LOBAlgebra: book engine, E3/E4 coupled-book experiments, and latest theory/data assessment.

Jump rates specify the generator matrix

Let S_τ be a finite-state continuous-time working model; s and r are states.

$$q_{sr} = \lim_{\delta \downarrow 0} \frac{\mathrm{P}(S_{\tau+\delta} = r \mid S_\tau = s)}{\delta}, \quad r \neq s, \qquad q_{ss} = -\sum_{r \neq s} q_{sr}.$$

- $\boldsymbol{Q} = (q_{sr})$ has nonnegative off-diagonal entries and zero row sums.
- Source states index rows; destination states index columns.
- One unit of τ is normalized to one observed row interval.

A time-homogeneous working model. Rates per second are not identified by the anonymous row index.

Event mechanics and event rates jointly determine dynamics

Event $e \in \mathcal{E}$ maps state s to $T_e(s)$ at nonnegative rate $\lambda_e(s)$. For a scalar observable $v(s)$, stack its values in a column vector $\boldsymbol{v}$.

$$(\boldsymbol{Q}\boldsymbol{v})_s = \sum_{e \in \mathcal{E}} \lambda_e(s) \left(v(T_e(s)) - v(s) \right).$$

- $\mathcal{E}$: allowed arrivals, cancellations and matching events.
- T_e: the change made to the represented book state.
- λ_e: how often that event occurs per model time unit.

LOBAlgebra separates state-change rules from state-dependent intensities.

Exponentiating rates gives a transition law

For lag $h \geq 0$, the transition matrix is

$$\boldsymbol{P}(h) = \exp(h\boldsymbol{Q}), \qquad P_{sr}(h) = \mathrm{P}(S_{\tau+h} = r \mid S_\tau = s).$$

If $\boldsymbol{\pi}(\tau)$ is a **row vector** of state probabilities,

$$\boldsymbol{\pi}(\tau + h) = \boldsymbol{\pi}(\tau)\boldsymbol{P}(h), \qquad \mathrm{E}[v(S_{\tau+h}) \mid S_\tau = s] = (\boldsymbol{P}(h)\boldsymbol{v})_s.$$

Rows of $\boldsymbol{P}(h)$ sum to one. The exponential includes paths with multiple jumps; h is not a disclosed target horizon.

Coupling makes rates depend on both books

A joint state $(a, b) \in \{1, 2, 3, 4\}^2$ records the regimes of `i0` and `i1`.

$$q_{(a,b),(a',b)} = \lambda_0(a \to a' \mid b), \qquad a' \neq a,$$
$$q_{(a,b),(a,b')} = \lambda_1(b \to b' \mid a), \qquad b' \neq b.$$

- $\boldsymbol{Q} = \boldsymbol{Q}_0 + \boldsymbol{Q}_1$; one book changes per instantaneous jump.
- Other off-diagonal entries are zero; diagonals are negative row sums.
- Coupled: 96 rate parameters; independent-book restriction: 24.

Statistical regimes learned from inputs; neither literal queue states nor identified causal cross-impact.

Filter probabilities before constructing the target readout

Let $\boldsymbol{\ell}_t$ be the row vector of Gaussian input likelihoods.

For $t \geq 1$, update filtered state probabilities $\boldsymbol{\pi}_t$ using $\boldsymbol{P} = \exp(\boldsymbol{Q})$:

$$\boldsymbol{\pi}_t = \frac{(\boldsymbol{\pi}_{t-1}\boldsymbol{P}) \odot \boldsymbol{\ell}_t}{((\boldsymbol{\pi}_{t-1}\boldsymbol{P}) \odot \boldsymbol{\ell}_t)\mathbf{1}_{16}}.$$

Here $\mathbf{1}_{16}$ is a column of ones; $\odot$ multiplies coordinates.

$$\boldsymbol{\psi}_t^{\text{state}} = (\boldsymbol{c}_t', \boldsymbol{\pi}_t, \boldsymbol{\pi}_t \otimes \boldsymbol{f}_t')', \qquad \widehat{y}_{t,k} = \widehat{a}_k + \widehat{\boldsymbol{b}}_k' \boldsymbol{\psi}_t^{\text{state}}.$$

$\boldsymbol{f}_t$: eight PCA input factors; $\boldsymbol{c}_t$: 112 inputs + 16 factor-memory deviations.

$\otimes$: pairwise state/factor products; 272 readout features.

Let past activity change the generator

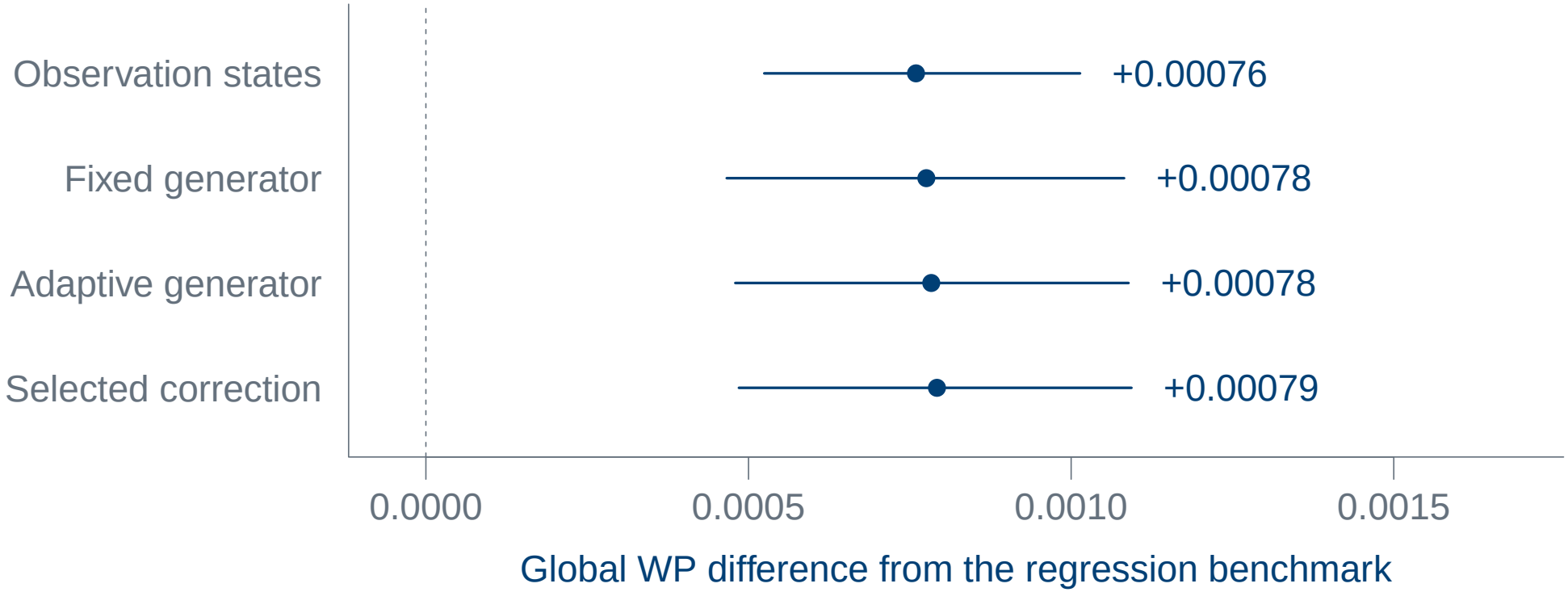
$$\boldsymbol{Q}_t = \kappa_0(a_{0,t-1})\boldsymbol{Q}_0 + \kappa_1(a_{1,t-1})\boldsymbol{Q}_1, \qquad \kappa_b(a) > 0.$$

$$\widehat{y}_{t,k} = \widehat{y}_{t,k}^{\text{base}} + \eta_k(\widehat{\alpha}_k + \widehat{\boldsymbol{\gamma}}_k' \boldsymbol{\psi}_t^{\text{clock}}).$$

- $a_{b,t-1}$: low/high activity in book $b \in \{0, 1\}$, observed one row earlier.
- Clock multipliers κ_b preserve a valid generator.
- $\boldsymbol{\psi}_t^{\text{clock}}$: 240 filtered-state and state/direction features.
- Fit intercept $\widehat{\alpha}_k$, slopes $\widehat{\boldsymbol{\gamma}}_k$ and select shrinkage η_k; $\eta_k = 0$ keeps the benchmark.

Fitted low/high clocks: book 0, 0.70/1.12; book 1, 0.44/1.34. Base prediction: the unchanged richer regression.

State gating nearly matches the generator corrections



Gain over 0.6446; conditional paired 95% sequence intervals. Same reused full validation and mask.

The official mask selects a different score population

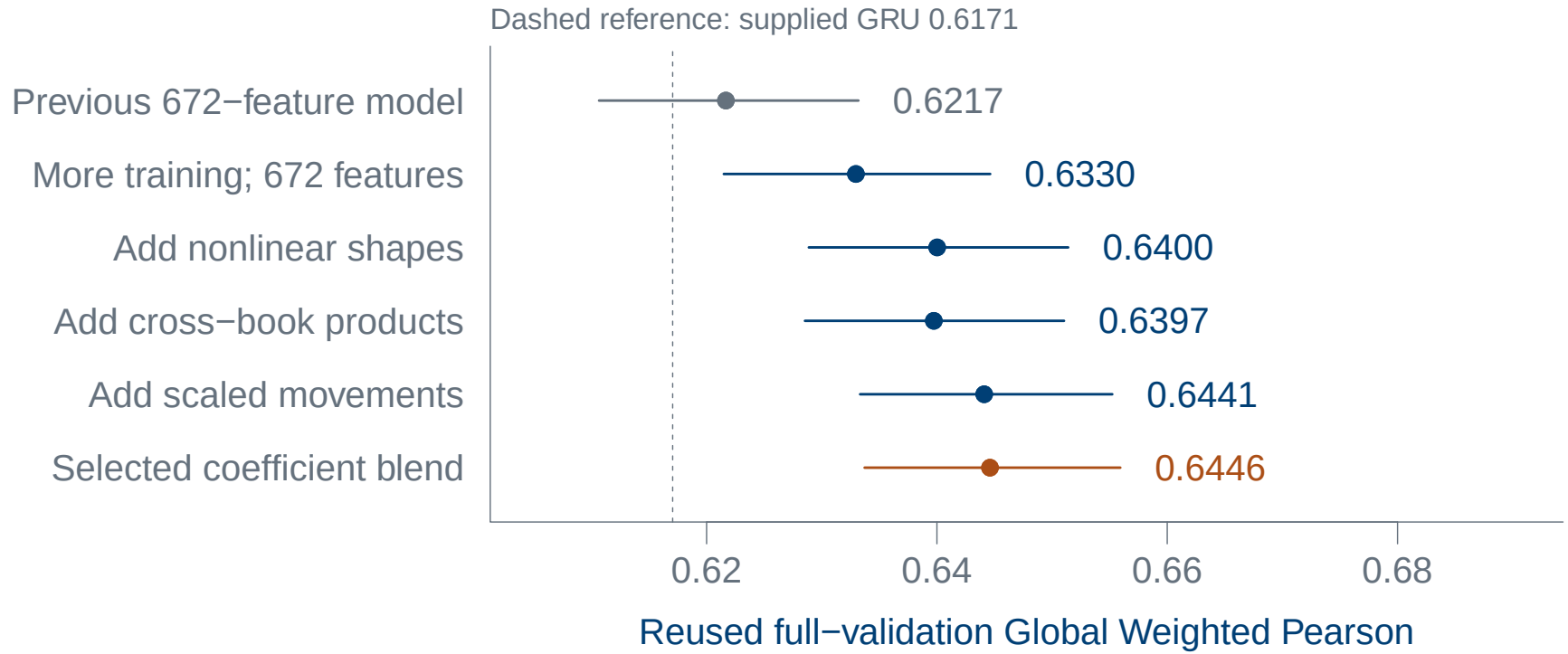
Same coupled procedure	Global WP
Development: all required rows	0.4110
Fresh holdout: all required rows	0.4019
Fresh holdout: official selected rows	**0.6002**

The callback receives no scoring mask and predicts every required row.

Compare scores on the same moments before explaining a gap.

Same predictions and sequences; no model or penalty changed after assessment.

More training and nonlinear features improve prediction



Full supplied validation, including development. Bars: conditional 95% sequence-bootstrap intervals.

More parameters can improve generalization again

Beyond an exact training fit

Richer classes can admit smoother fitted functions.

Regularization and averaging determine which prediction emerges.

The first LOB prototype

272 readout features; 131,072 fitting rows.

Conditional ridge effective degrees of freedom: 245.7 and 245.5.

This run does not cross the readout's interpolation threshold.

Assess capacity, inductive bias and averaging with unseen data.

Effective degrees of freedom = trace of the fitted-value map, holding design and weights fixed; double-descent perspective: (Belkin et al.2019).

Generator averaging and prediction averaging differ

For m_0 generators $\boldsymbol{Q}_m$ with nonnegative weights α_m on the **same aligned states and time unit**,

$$\overline{\boldsymbol{Q}} = \sum_{m=1}^{m_0} \alpha_m \boldsymbol{Q}_m, \qquad \alpha_m \geq 0, \quad \sum_{m=1}^{m_0} \alpha_m = 1$$

is another valid generator, but generally

$$\exp(\overline{\boldsymbol{Q}}) \neq \sum_{m=1}^{m_0} \alpha_m \exp(\boldsymbol{Q}_m).$$

A third construction averages the final target predictions.

Test whether averaging improves held-out prediction.

Proposed next experiments; no generator ensemble was fitted or scored in this run.

Match the score and combine complementary predictions

Features and objective

Start with OLS on rich causal features.
Tune each target separately.
Use the exact clipped, weighted score.
Check output scale where clipping matters.

Capacity and averaging

Compare nonlinear readouts and richer states.
Blend models with different errors.
Choose ensemble weights on development data.
Share feature calculations to save CPU time.

One-CPU estimates: regression 36.6 min; selected lag correction 13.3 min.

Use an untouched final holdout and measure the complete callback.

Official FAQ: hidden scoring mask; one CPU; 60-minute inference limit | checked 27 October 2026

The same observed summary can hide different futures

Let X_τ be the full Markov book state and $R_\tau = \mathcal{R}(X_\tau)$ its observed summary. The conventional capitals here denote states, not numeric vectors.

$$\mathrm{E}[v(R_{\tau+h}) \mid X_\tau] = [\exp(h\mathcal{L})(v \circ \mathcal{R})]\,(X_\tau).$$

- $\mathcal{L}$: full-state generator; v: scalar function of the summary.
- $v \circ \mathcal{R}$: apply the summary map, then the observable.
- A closed summary model requires this expectation to depend only on R_τ.

Different hidden order composition can imply different futures at the same summary. Closure of our 16 regimes was not established.

Forecast changes in levels and squares of the same inputs

For the already defined $\boldsymbol{z}_t \in \mathbb{R}^{112}$, define the observable map

$$\boldsymbol{v}(\boldsymbol{z}) = \begin{pmatrix} \boldsymbol{z} \\ \boldsymbol{z} \odot \boldsymbol{z} \end{pmatrix}, \qquad \boldsymbol{\delta}_{h,t} = \boldsymbol{v}(\boldsymbol{z}_{t+h}) - \boldsymbol{v}(\boldsymbol{z}_t) \in \mathbb{R}^{224}.$$

Fit separate maps at $h \in \{1, 8, 32\}$ rows:

$$\widehat{\boldsymbol{\delta}}^{\mathrm{cur}}_{h,t} = \widehat{\boldsymbol{a}}^{\mathrm{cur}}_h + \widehat{\boldsymbol{B}}^{\mathrm{cur}}_h \boldsymbol{\phi}^{\mathrm{cur}}_t,$$
$$\widehat{\boldsymbol{\delta}}^{\mathrm{hist}}_{h,t} = \widehat{\boldsymbol{a}}^{\mathrm{hist}}_h + \widehat{\boldsymbol{B}}^{\mathrm{hist}}_h \boldsymbol{\phi}_t.$$

$\boldsymbol{\phi}^{\mathrm{cur}}_t$: 500 current-input functions; $\boldsymbol{\phi}^{\mathrm{hist}}_t = \boldsymbol{\phi}_t$: 1,284 history features. Both maps retain at most 32 fitted output directions; this rank is not a state count.

Compare the current input with an earlier forecast

For either fitted map, split $\widehat{\boldsymbol{\delta}}_{h,t-h}$ into level block $\widehat{\boldsymbol{\delta}}^{(1)}_{h,t-h}$ and squared-level block $\widehat{\boldsymbol{\delta}}^{(2)}_{h,t-h}$.

$$\widehat{\boldsymbol{\mu}}_{h,t-h} = \mathrm{clip}(\boldsymbol{z}_{t-h} + \widehat{\boldsymbol{\delta}}^{(1)}_{h,t-h}, -8, 8), \qquad \boldsymbol{\epsilon}_{h,t} = \boldsymbol{z}_t - \widehat{\boldsymbol{\mu}}_{h,t-h}.$$

$$\boldsymbol{\nu}_{h,t-h} = \mathrm{clip}(\boldsymbol{z}^{\odot 2}_{t-h} + \widehat{\boldsymbol{\delta}}^{(2)}_{h,t-h} - \widehat{\boldsymbol{\mu}}^{\odot 2}_{h,t-h}, .01, 64).$$

$\boldsymbol{\nu}$ is a numerical variance proxy, not a calibrated predictive variance.

$$\boldsymbol{\psi}^{\mathrm{surp}}_t = \mathrm{stack}_{h\in\{1,8,32\}} \left(\boldsymbol{\epsilon}_{h,t},\ \mathrm{clip}(\boldsymbol{\epsilon}_{h,t} \oslash \sqrt{\boldsymbol{\nu}_{h,t-h}}, -8, 8),\ \log(1 + \boldsymbol{\nu}_{h,t-h})\right).$$

1,008 features. $\odot 2$: elementwise square; $\oslash$: elementwise division; clipping, roots and logs act elementwise.

Plain lags provide a correction without a forecast map

For each $h \in \{1, 8, 32\}$, let $\boldsymbol{d}_{h,t}^{\mathrm{lag}} = \boldsymbol{z}_t - \boldsymbol{z}_{t-h}$ when $t \geq h$, and zero otherwise. Starting from zero, update squared differences with half-life 128:

$$\boldsymbol{s}_{h,t} = r_{128}\boldsymbol{s}_{h,t-1} + (1 - r_{128})(\boldsymbol{d}_{h,t}^{\mathrm{lag}})^{\odot 2}, \qquad \boldsymbol{\nu}_{h,t}^{\mathrm{lag}} = \mathrm{clip}(\boldsymbol{s}_{h,t}, .01, 64).$$

$$\boldsymbol{\psi}_t^{\mathrm{lag}} = \mathrm{stack}_{h\in\{1,8,32\}} \left(\boldsymbol{d}_{h,t}^{\mathrm{lag}},\ \mathrm{clip}(\boldsymbol{d}_{h,t}^{\mathrm{lag}} \oslash \sqrt{\boldsymbol{\nu}_{h,t}^{\mathrm{lag}}}, -8, 8),\ \log(1 + \boldsymbol{\nu}_{h,t}^{\mathrm{lag}}) \right).$$

Fit OLS to $y_{t,k} - \widehat{y}_{t,k}^{\mathrm{base}}$, keeping the richer regression base unchanged.

1,008 features; add 0.25 times the fitted residual prediction for each target, as selected on development data.

Normalize squared input errors before comparing maps

For horizon h, coordinate j and assessment origin (n, t), the level increment is

$$\delta^{(1)}_{n,h,t,j} = z_{n,t+h,j} - z_{n,t,j}.$$

Let $\widehat{\sigma}^{\Delta}_{h,j} > 0$ be its scale estimated on fitting data only. For the set $\mathcal{A}$ of assessment origins, define

$$\mathrm{MSE}^{\mathrm{level}}_{h} = \frac{1}{112|\mathcal{A}|} \sum_{(n,t)\in\mathcal{A}} \sum_{j=1}^{112} \left(\frac{\delta^{(1)}_{n,h,t,j} - \widehat{\delta}^{(1)}_{n,h,t,j}}{\widehat{\sigma}^{\Delta}_{h,j}} \right)^2 .$$

Lower is better. Squared-level MSE uses the second block and its own fitting scales. Persistence predicts a zero increment.

History lowers MSE for the tested input forecasts

Horizon	Persistence	Current row	With history
1 row	0.9053	0.6995	0.6799
8 rows	0.9147	0.7538	0.7277
32 rows	0.9000	0.7103	0.6837

- The preceding normalized level-increment MSE; lower is better.
- All six level/square comparisons favor history with paired intervals below zero.
- This tests these working bases, rather than proving a full-state Markov claim.

32 new input-assessment sequences; 512 origins each. No target labels used for these comparisons.

Plain lags win the development comparison

Fixed model	Reused valid.	New TRAIN
Richer regression	0.6446	0.3662
Earlier state correction	0.6454	0.3676
Plain lags (selected)	**0.6458**	**0.3675**
Current-row map correction	0.6447	0.3670
History-map correction	0.6450	0.3670
History map, EMA scale	0.6456	0.3677

A gain is not established on the new TRAIN assessment.

Global WP: 1,873 reused masked-validation sequences; 64 new TRAIN sequences, all required rows. Compare within each column.

Extend the basis while keeping OLS coefficients

Let $\mathcal{B}$ collect the four added blocks below.

Refit all coefficients jointly:

$$\widehat{y}_{t,k} = \widehat{a}_k + \widehat{\boldsymbol{b}}_k^{\text{base}\prime}\boldsymbol{\phi}_t + \sum_{b\in\mathcal{B}} \widehat{\boldsymbol{b}}_k^{(b)\prime}\boldsymbol{\phi}_t^{(b)}.$$

- $\boldsymbol{\phi}_t$: the same 1,284 input/history features.
- $\boldsymbol{\phi}_t^{\text{lag}}$: changes over 1 and 16 rows; 224 terms.
- $\boldsymbol{\phi}_t^{\text{spline}}$: $(z_{t,j} - 1)_+$ and $(-z_{t,j} - 1)_+$; 224 terms.
- $\boldsymbol{\phi}_t^{\text{cross}}$: cross-book current-input/memory products; 224 terms.
- $\boldsymbol{\phi}_t^{\text{activity}}$: current inputs times two book-activity gates; 224 terms.

2,180 slopes at most. $(u)_+ = \max(u, 0)$. Activity gates: mean squared trade inputs, mapped to $a/(1+a)$.

Extra features do not help this small pilot

Basis	Columns	Raw OLS	Best ridge
Existing	1,284	0.6252	0.6564
Add lags	1,508	0.6213	0.6527
Add splines	1,508	0.6172	0.6522
Add lead–lag	1,508	0.6130	0.6510
Add activity	1,508	0.6190	0.6530
All additions	2,180	0.5951	0.6402

Test added capacity together with sample size and shrinkage.

Global WP on 32 reused development sequences. All fits: 256 training sequences, 65,536 labeled rows.

The existing larger-sample model stays ahead

Model	Reused valid.	New TRAIN
Pilot raw OLS	0.6023	0.3110
Pilot selected ridge	0.6369	0.3564
Existing 1,024-sequence fit	0.6492	0.3932

- Pilot choices were recorded before these assessments.
- Neither selector chose an added feature block.
- No replacement model is promoted from this experiment.

64 reused masked-validation sequences; 32 new TRAIN sequences, all required rows. Compare within columns.

Use the precise documentation alongside the lecture ideas

- Quick start, data overview, submission guide.
- Current rules, timeline, FAQ; checked 27 October 2026.
- Starter pack: `README.md`, `METRIC.md`, `utils.py` and supplied baseline.
- James et al. (2021), Chapters 2, 5, 6 and 7.
- Official glmnet vignette; Friedman et al. (2010).
- Stationarity and ergodicity: Breiman, chapter 6; fixed generators: Lalley.

Local foundations: IntroDS statistical-learning, linear-model and resampling decks; ESDA/SBDA and macro_sdf presentation conventions.

References

Belkin, M., Hsu, D., Ma, S., and Mandal, S. (2019). Reconciling modern machine-learning practice and the classical bias–variance trade-off. *Proceedings of the National Academy of Sciences*, 116(32):15849–15854.

Friedman, J., Hastie, T., and Tibshirani, R. (2010). Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1):1–22.

James, G., Witten, D., Hastie, T., and Tibshirani, R. (2021). *An Introduction to Statistical Learning: with Applications in R*. Springer, 2 edition.