POSTERIOR MEANS UNDERESTIMATE DIFFUSION: AUDITABLE IDENTIFICATION OF POSITION-ONLY STOCHASTIC DYNAMICS

Anonymous authors
Paper under double-blind review

ABSTRACT

Learning stochastic dynamics from noisy positions requires estimating forces and diffusion without observing velocity. We show that the standard smooth-then-regress approach systematically underestimates diffusion in our benchmarks because posterior-mean reconstruction discards uncertainty in velocity increments. We introduce OMWG-C, an endpoint-corrected weak-moment estimator that restores this missing variance from smoother cross-time covariances. Drift and diffusion are unknown functions represented in supplied candidate dictionaries; the second-order kinematic structure is assumed. In a frozen synthetic confirmation, OMWG-C lowers diffusion error in every unit, with a substantial median reduction relative to the posterior-mean baseline, while development results show little change in one-step prediction. For data without ground truth, we introduce a pre-set cross-sampling-rate audit that refits the dynamics after downsampling. Endpoint-corrected fits fail this audit in all cell-migration cohorts, despite better prediction in most cohorts, whereas ocean-drifter point estimates pass despite worse prediction. Passing tests rate stability, not physical correctness. These results establish predictive fit, parameter recovery, and sampling-rate stability as distinct dimensions of position-only equation learning.

1 INTRODUCTION

Position trajectories are often easier to measure than the forces that generate them. Microscopy records cell locations, GPS tracks animals or drifters, and imaging systems track particles. In each case, the scientific question concerns the underlying equation of motion. We consider an underdamped diffusion with latent velocity,

$$dQ_t = V_t\,dt, \qquad dV_t = b(Q_t, V_t)\,dt + \Sigma(Q_t, V_t)\,dW_t, \qquad Y_k = Q_{t_k} + \varepsilon_k, \tag{1}$$

where only the noisy positions Y_k are observed. The target is the drift b and diffusion covariance $a = \Sigma\Sigma^{\mathsf{T}}$, not merely a denoised path or the next position.

A reconstructed trajectory can look convincing while giving the wrong diffusion. Prediction may barely register the error. Direct position differences amplify localization noise as Δt^{-1} for velocity and Δt^{-2} for acceleration. Smoothing avoids that explosion, but substituting a posterior mean for an uncertain velocity increment discards its conditional variance (Figure 1a). Kalman filtering and smoothing recover latent states under a specified state-space model (Kalman, 1960; Rauch et al., 1965). Here, the dynamics themselves are unknown. Which uncertainty must be retained when a reconstructed path is used to learn stochastic forcing?

We identify an endpoint velocity-increment variance that can be computed from smoother cross-time covariances and added directly to the diffusion regression. Our estimator incorporates this term into integral weak moments. We call it OMWG-C, the confirmed endpoint-corrected variant of Observation-Marginalized Weak Generator. Against the frozen posterior-mean baseline (hereafter TWOSTAGE), it lowers diffusion error in all 72 confirmation units, with a median per-unit error ratio of 0.421. Yet the median paired change in one-step prediction negative log-likelihood (NLL) is only -0.00147 in the development comparison. Figure 1 connects this omitted uncertainty to attenuation and recovery.

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.

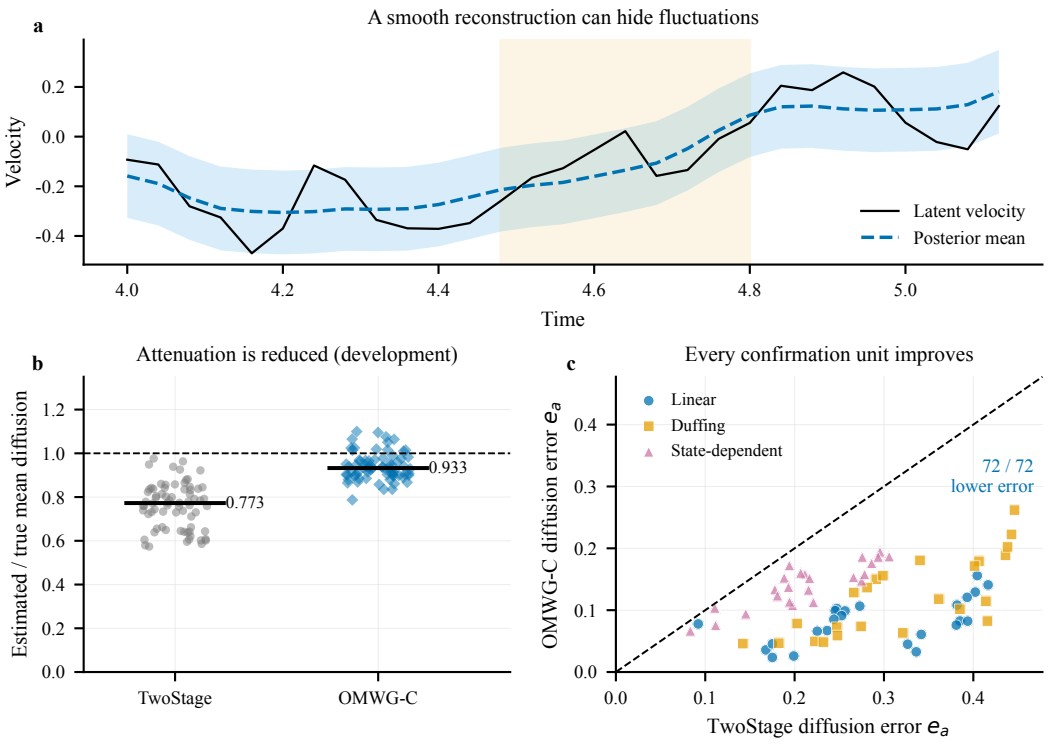


Figure 1: From a smooth trajectory to diffusion recovery. **a**, In a fixed, unselected development example (linear system, seed 128, 2% noise), posterior mean velocity suppresses fluctuations visible in the latent path. Blue shading shows the pointwise 95% posterior band. The highlighted lag-8 block illustrates the endpoint correction and is examined in Figure 5. **b**, Estimated-to-true occupancy-mean diffusion for TwoStage and OMWG-C across all 72 development units. Black bars mark medians; the dashed line marks truth. **c**, All 72 separate frozen confirmation units, spanning three systems, two noise levels, and 12 seeds. Every corrected diffusion error lies below its paired baseline error; the dashed diagonal denotes equal error.

A matched ablation finds that adding the endpoint term at a single lag already recovers much of the benefit. We evaluate its contribution and computational cost. For real trajectories, diffusion ground truth is unavailable. We therefore fit the model again after downsampling. We keep the moment windows fixed in physical time and check whether the estimated diffusion remains stable.

The paper makes three contributions.

- **An explicit correction for diffusion attenuation.** We derive the endpoint posterior variance discarded by smooth-then-regress estimation and a computable correction, together with an exact decomposition of its remaining conditional-moment error.
- **Diffusion recovery with little predictive change.** OMWG-C reduces diffusion error in all 72 frozen confirmation units to a median 0.421 of baseline error. Matched ablations identify the endpoint correction as the main source of improvement; development prediction scores barely change.
- **A cross-sampling-rate audit beyond prediction.** We introduce a pre-set re-identification test of diffusion stability. Its opposite outcomes on cell and drifter trajectories expose a distinction between predictive accuracy and sampling-rate stability; synthetic controls show why stability alone cannot certify recovery. Neural controls reinforce this picture: a supervised recurrent reconstructor inherits the attenuation, while a latent neural SDE passes the audit.

2 PROBLEM SETUP

Finite dictionaries. We parameterize drift and diffusion with fixed, interpretable dictionaries,

$$b_\beta(q,v) = \Phi(q,v)\beta, \qquad a_\gamma(q,v) = \max\{\Psi(q,v)\gamma, 10^{-10}\}. \tag{2}$$

In the one-dimensional experiments, $\Phi = [1, q, v, q^3]$ and $\Psi = [1, q^2]$. The drift terms represent a constant force, linear position and velocity dependence, and cubic confinement; the diffusion terms allow constant and quadratic position dependence. This compact dictionary contains all three synthetic systems in Section 4.1 and keeps the candidate function class fixed when comparing uncertainty corrections. The coefficients and their nonzero pattern are unknown, but the dictionary and the kinematic relation $dQ = V\,dt$ are supplied. We estimate coefficients within this class, without automated dictionary selection or a claim of recovery outside its span. The floor enforces positive predicted diffusion; the coefficient fit itself is an unconstrained linear moment solve (Appendix B).

Why compare physical scales? For locally constant acceleration drift and diffusion, a short position increment satisfies

$$\mathrm{Var}(Y_{t+h} - Y_t - V_t h \mid V_t) \simeq \frac{a h^3}{3} + 2r, \tag{3}$$

where $r = \mathrm{Var}(\varepsilon_k)$. At a single h, this variance confounds physical diffusion and localization noise unless one is separately calibrated. Across horizons, the diffusion contribution scales as h^3, while the noise contribution stays constant. We use noise calibration to separate these contributions and posterior covariances to retain uncertainty in reconstructed velocity.

3 ENDPOINT-CORRECTED WEAK-MOMENT ESTIMATION

We first specify the missing posterior term, then describe the noise calibration, smoother, and weak moments that supply its inputs. The full procedure is summarized in Appendix B.

3.1 ENDPOINT POSTERIOR CORRECTION

The key change is to retain uncertainty in the velocity increment, rather than regress on its posterior mean alone. Let

$$D = V_{t+h} - V_t, \qquad B = \int_t^{t+h} b_\beta(X_s)\,ds, \qquad R = D - B. \tag{4}$$

Writing $X_s = (Q_s, V_s)$ and $\mu_s = \mathbb{E}[X_s \mid Y]$, define the posterior-mean residual and the implemented corrected moment by

$$\widetilde{R}_h = \mu^V_{t+h} - \mu^V_t - \int_t^{t+h} b_\beta(\mu_s)\,ds, \qquad \widetilde{M}_h = \widetilde{R}_h^2 + \mathrm{Var}(D \mid Y), \tag{5}$$

and let $\delta_h = \mathbb{E}[R \mid Y] - \widetilde{R}_h$ denote the conditional-mean plug-in error.

Proposition 1 (Conditional residual identity)**.** *If D and B are square-integrable conditional on Y, then*

$$\mathbb{E}[R^2 \mid Y] = \mathbb{E}[R \mid Y]^2 + \mathrm{Var}(R \mid Y), \tag{6}$$

$$\mathrm{Var}(R \mid Y) = \mathrm{Var}(D \mid Y) + \mathrm{Var}(B \mid Y) - 2\mathrm{Cov}(D, B \mid Y). \tag{7}$$

Consequently, the exact discrepancy from the implemented correction is

$$\mathbb{E}[R^2 \mid Y] - \widetilde{M}_h = 2\widetilde{R}_h\delta_h + \delta_h^2 + \mathrm{Var}(B \mid Y) - 2\mathrm{Cov}(D, B \mid Y). \tag{8}$$

The posterior-mean baseline keeps only a plug-in version of the squared conditional mean. OMWG-C additionally restores the endpoint term, which the RTS posterior supplies as

$$\mathrm{Var}(V_j - V_i \mid Y) = P^{vv}_j + P^{vv}_i - 2P^{vv}_{ij}, \tag{9}$$

where $P_i = \mathrm{Var}(X_i \mid Y)$ and $P_{ij} = \mathrm{Cov}(X_i, X_j \mid Y)$. Figure 5a in Appendix A.2 illustrates this identity on one fixed development block. The shaded band shows posterior state uncertainty. The printed add-back is computed directly from posterior covariances.

Scope of the correction. The identity holds under a specified conditional law. Our implementation uses means and covariances from a surrogate Gaussian smoother. Error in this surrogate posterior is separate from the omitted terms in Equation 8. Appendix A.2 gives a sufficient endpoint-dominance bound and the exact affine finite-grid expression. Under this surrogate, omitted-to-endpoint ratios are small on the linear and Duffing development blocks (Appendices A.2 and A.6); these diagnostics do not assess calibration against the true nonlinear posterior.

We regress $\widetilde{M}_h/h$ on the interval-average diffusion dictionary for lags $4, 8, 16$. For five subsequent refits, the smoother uses the frozen linear drift terms $\widehat{\beta}_q q + \widehat{\beta}_v v$ and the current diffusion intercept $s = \max\{\widehat{\gamma}_0, 10^{-3}\}$ as a constant process intensity. The residual and diffusion regression retain the full supplied dictionaries, including the cubic drift and q^2 diffusion terms. Only the smoother uses the linear, constant-intensity approximation. Drift coefficients remain frozen. This resembles EM (Dempster et al., 1977), but optimizes weak moments instead of the expected complete-data likelihood. Appendix A proves Proposition 1, Corollary 1, Proposition 2, and Lemma 1.

3.2 Noise calibration and latent-state reconstruction

The smoother needs an observation variance without using test trajectories. For every interior observation, let

$$e_k = Y_k - w_k^- Y_{k-1} - w_k^+ Y_{k+1}, \quad w_k^- = \frac{t_{k+1} - t_k}{t_{k+1} - t_{k-1}}, \quad w_k^+ = 1 - w_k^-. \tag{10}$$

We estimate

$$\widehat{r} = \frac{1}{N} \sum_k \frac{e_k^2}{1 + (w_k^-)^2 + (w_k^+)^2}. \tag{11}$$

Lemma 1 (Finite-step chord-statistic bias)**.** *Consider three equally spaced observations with interval Δ. Suppose acceleration drift and white acceleration intensity are constant over the two intervals at b and a, respectively, and localization errors are independent of the latent process and each other, zero mean, and have variance r. Then the statistic in Equation 11 satisfies*

$$\mathbb{E}[\widehat{r}] = r + \frac{a\Delta^3}{9} + \frac{b^2\Delta^4}{6}. \tag{12}$$

This local chord statistic is a difference-based noise estimator in the spirit of Rice (1984). Lemma 1 shows how finite-step dynamics contaminate this estimate. In the constant-coefficient model, stochastic acceleration enters at order Δ^3 and deterministic curvature at order Δ^4. For state-dependent coefficients, this is a local approximation; additional terms can contribute at order Δ^4. Appendix A.7 gives the proof. In the frozen confirmation, the pooled median is $\widehat{r}/r = 1.001$. Table 2 reports the condition-level values.

Given $\widehat{r}$, we use the constant-velocity working model

$$X_{k+1} = F_k X_k + \eta_k, \qquad F_k = \begin{bmatrix} 1 & \Delta t_k \\ 0 & 1 \end{bmatrix}, \quad \eta_k \sim \mathcal{N}\left(0, s \begin{bmatrix} \Delta t_k^3/3 & \Delta t_k^2/2 \\ \Delta t_k^2/2 & \Delta t_k \end{bmatrix}\right). \tag{13}$$

The process intensity s is selected on training trajectories. This working model links velocity to position through $dQ = V\,dt$ without specifying a force law. It treats unmodeled acceleration as process noise. RTS smoothing yields posterior means μ_k, marginal covariances P_k, and gains from which we obtain cross-time covariances $P_{ij} = \mathrm{Cov}(X_i, X_j \mid Y)$.

3.3 Integral instrumental moments

For an interval $[t, t+h]$, integration of Equation 1 gives

$$V_{t+h} - V_t = \left(\int_t^{t+h} \Phi(X_s)\, ds \right) \beta + M_{t,h}, \tag{14}$$

where $M_{t,h}$ is a martingale increment. For the complete state, the start feature $\Phi(X_t)$ is an adapted instrument. The implementation substitutes smoothed-state features, uses left-endpoint quadrature, and forms non-overlapping blocks within each lag. Because smoothed features depend on future observations, we use approximate instrumental moments. Appendix A.1 details the resulting dependence and quadrature errors. The frozen baseline, TwoStage, estimates drift and diffusion on posterior means with lag 8.

4 RECOVERY, MECHANISM, AND COMPUTATIONAL TRADE-OFFS

We first test whether the correction improves diffusion recovery, then isolate its contribution, and finally compare accuracy and computational cost. Unless marked as confirmation, results on the development grid are post-hoc diagnostics.

4.1 EXPERIMENTAL SETUP

Synthetic systems. We simulate (i) a linear oscillator $b = -q - 0.4v$, $a = 0.36$; (ii) a Duffing system $b = q - 0.5v - q^3$, $a = 0.25$; and (iii) state-dependent diffusion $b = -q - 0.45v$, $a = 0.16 + 0.08q^2$. We simulate with Euler–Maruyama at step 0.002 and observe every 0.04 time units, giving 240 observations per trajectory. The localization-noise standard deviation is 2% or 5% of the position standard deviation. Each unit has 40 fit trajectories, 8 validation trajectories, and independent test paths. Development uses seeds 128–139 plus a targeted state-dependent study on 156–165. The frozen one-shot confirmation uses untouched seeds 144–155, giving $3 \times 2 \times 12 = 72$ units.

Baselines and metrics. Our position-only comparators include TWOSTAGE, central-difference Kramers–Moyal regression, and an RBF Gaussian-process smoother with the same weak-IV backend. We also evaluate the published linear Underdamped Langevin Inference (ULI) estimator (Brückner et al., 2020) and the parametric force/diffusion path of Stochastic Force Inference (SFI). Exact-discrete affine CAR(2) Kalman maximum likelihood provides a linear reference. A parameter-matched GRU predictor and a true-state weak-IV reference complete the comparison. ULI uses its official SFI 2.0.3 implementation (Ronceray, 2026), the same drift and diffusion dictionaries, and its own train-position noise calibration. Function recovery uses relative occupancy-weighted L_2 errors e_b and e_a, with joint error $e_{\text{joint}} = \sqrt{e_b e_a}$. Prediction is natural-log observation NLL per predicted position.

4.2 DOES THE CORRECTION RECOVER DIFFUSION?

The correction lowers diffusion error in all 72 frozen confirmation units (Figure 1c). The median per-unit diffusion-error ratio is 0.421. Drift is frozen by construction and its error ratio is 1.000. The median joint-error ratio is 0.649 (geometric mean 0.629); 69 units improve joint error by at least 10%, and none degrades. A seed-clustered bootstrap gives a 95% interval $[0.571, 0.703]$ for the median joint-error ratio. The weakest condition is state-dependent diffusion at 2% noise, with a median joint-error ratio of 0.865. Appendix C.2 shows every joint-error point by condition.

Recovery changes while prediction barely moves. The diffusion improvement has little effect on one-step prediction: the median paired NLL difference between OMWG-C and TWOSTAGE is only -0.00147, with medians -2.183 and -2.181; CAR(2) and the GRU sit at -2.173 and -0.550. Evaluating predictive fit alone would obscure the estimator's principal benefit. Appendix C.2 also shows Duffing function recovery over the occupied state range (Figure 7).

The attenuation extends beyond RTS in our reconstruction control. With the same plug-in backend, a supervised bidirectional GRU underestimates diffusion in all 24 development units. Its median estimated-to-true ratio is 0.716, compared with 1.018 for the true-state reference. Its velocity RMSE is nevertheless comparable to RTS (Table 6).

4.3 WHICH COMPONENT PRODUCES THE GAIN?

A matched 72-unit development ablation freezes the same 40-trajectory fit split, reconstructed starting states, drift, and scoring rule (Figure 2; numerical values in Table 5). Multi-lag pooling alone increases diffusion error; iterating without the endpoint term increases it further. A single-lag correction already removes much of the attenuation and is a competitive simplification. The full method has the lowest pooled median error and improves 69/72 development units. Appendix D.1 gives condition-level values and paired seed-cluster intervals, including the cases favoring the simpler correction.

The bias direction supports this result. The median estimated-to-true occupancy-mean diffusion ratio is 0.773 for TWOSTAGE, 1.037 for the single-lag correction, and 0.933 for OMWG-C (Figure 1b and Figure 2a). The single-lag variant provides this correction without five refits.

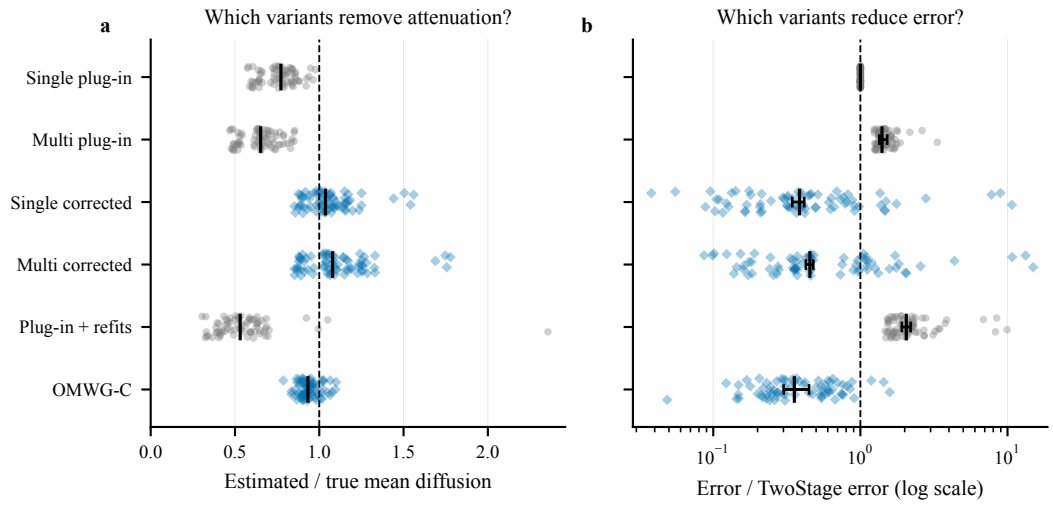


Figure 2: The endpoint term, rather than additional lags or refits alone, produces the gain. All six variants use the same 72 development units and fixed drift. **a**, Estimated-to-true occupancy-mean diffusion; the dashed line is truth. **b**, Per-unit diffusion-error ratios to TWOSTAGE on a log scale; the dashed line denotes equal error. All points, including large failures, are retained. Gray circles omit the correction and blue diamonds include it; black ticks are medians. Horizontal intervals in **b** are descriptive 95% intervals from 10,000 paired seed-cluster bootstrap resamples.

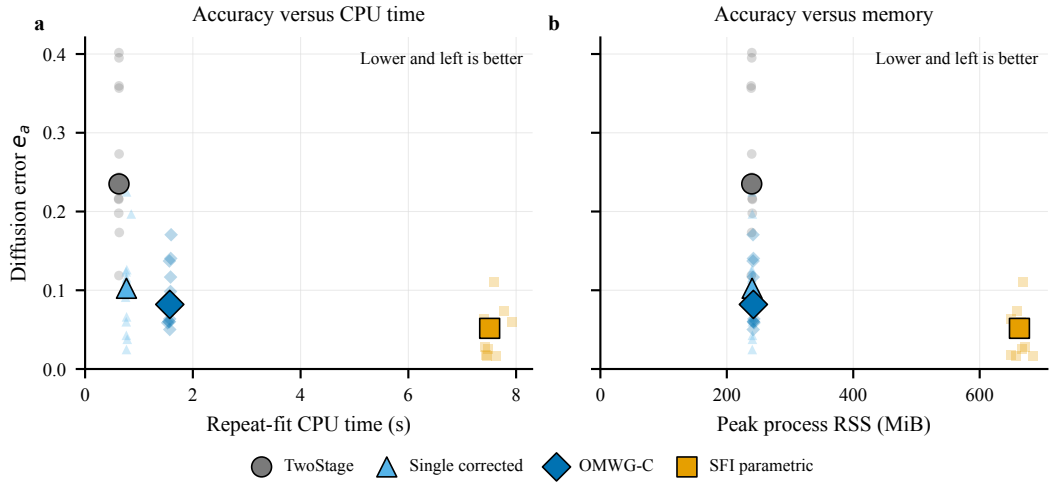


Figure 3: Accuracy–resource trade-off on the same 12 development units. Diffusion error versus **a**, repeat-fit CPU time and **b**, peak process RSS including runtime libraries and compilation caches. Faint symbols show individual units; outlined symbols pair marginal medians and need not represent an individual fit. Measurements retain native runtime parallelism; Appendix D.5 reports first/repeat wall time and joint error.

4.4 ACCURACY AND RESOURCE USE AGAINST STRONG BASELINES

Accuracy against the strongest matched comparator. On 72 matched development units, SFI parametric has lower median joint error than OMWG-C (0.072 versus 0.093); OMWG-C wins 21/72 paired units. On the initial 12-unit subset, the latent neural-SDE median is 0.107, versus 0.081 for OMWG-C and 0.052 for SFI (Appendices D.5 and D.4).

Precision–cost trade-off. On the matched subset, OMWG-C uses less CPU time and peak memory than SFI parametric, although its repeat-fit wall-time advantage narrows after compilation reuse (Figure 3). Appendix D.5 reports full timing and accuracy tables, runtime parallelism, and operation counts.

Other estimators and model specification. On the full development grid, OMWG-C has lower joint error than the official closed-form ULI path in all 72 units (pooled medians 0.093 versus 0.489). Appendix C.2 retains the complete external-baseline table and ULI calibration details.

Correctly specified CAR(2) wins 21/24 linear-system units, whereas OMWG-C wins all 24 Duffing units; they are close on state-dependent diffusion. The nonlinear dictionary thus helps when dynamics fall outside the affine likelihood model (Figure 6b; Table 4).

4.5 Transfer and multidimensional boundary

Transition-covariance recovery improves at unseen horizons, but shortest-horizon mean recovery fails the pre-set gate. Long-time gains are modest. Two-dimensional add-back helps constant diffusion but fails on the tested non-polynomial state-dependent system; a state-aware refinement is also rejected because it degrades linear-system recovery. Appendix C.4 reports these development-only results and their numerical limits.

5 From Recovery to Real-Data Interpretation

The synthetic results show why prediction alone can miss a diffusion-recovery error. For real trajectories, ground truth is unavailable. We instead check whether the fitted diffusion remains stable when the sampling interval doubles. We apply the same test to the baseline and corrected fits.

Real trajectories. The cell-collider data are the three public one-dimensional paired-cell cohorts sd01, sd02, and sd03 from Brückner et al. (2021), sampled every 10 minutes. Whole pairs are split deterministically 60/20/20, producing train/validation/test counts of 54/18/19, 93/31/31, and 51/17/18. Because the model omits the partner's state, it describes marginal effective dynamics. The second dataset comes from the NOAA Global Drifter Program (GDP) quality-controlled, interpolated hourly product (Elipot et al., 2016; 2022). It contains 94 drogued subtropical North Atlantic segments from 2016–2018, each with at least 480 hourly observations. We split them into 56/18/20 train/validation/test segments. One buoy appears in both validation and test partitions, but none crosses from training into either; GDP uses no validation-based fitting or selection. The two-dimensional model is fit independently at one- and two-hour sampling.

Estimator variants. The synthetic confirmation fixes five diffusion refits and defines OMWG-C. The cell pipeline selects the add-back fit or one of iterations 1–5 by validation NLL, with a TwoStage fallback. GDP uses two-dimensional endpoint add-back without those refits. Figure 4 therefore labels these real-data fits as endpoint-corrected variants, not as identical executions of the confirmed procedure. Both estimators retain their original train/validation splits and are re-identified at each rate. Appendix D.2 specifies the resampling and selection rules.

5.1 Cross-sampling-rate audit

We retain every second observation and re-identify every component, including $\widehat{r}$, the state posterior, and dynamics. Lags are changed from $(4, 8, 16)$ to $(2, 4, 8)$ so their physical durations remain fixed. If a_{eff} is diffusion averaged over the training occupancy distribution, the audit statistic is

$$\rho_{\mathrm{rate}} = \frac{a_{\mathrm{eff}}^{\mathrm{stride}\ 2}}{a_{\mathrm{eff}}^{\mathrm{native}}}. \tag{15}$$

Before either real-data analysis, we fixed the stability interval to $[0.8, 1.25]$. Acceptance denotes stability of occupancy-averaged diffusion across the two sampling rates. Smoothing can itself create autocorrelation and conditional structure. We therefore compare residual diagnostics with matched synthetic trajectories passed through the same smoother. Figure 4 compares audit decisions with predictive differences.

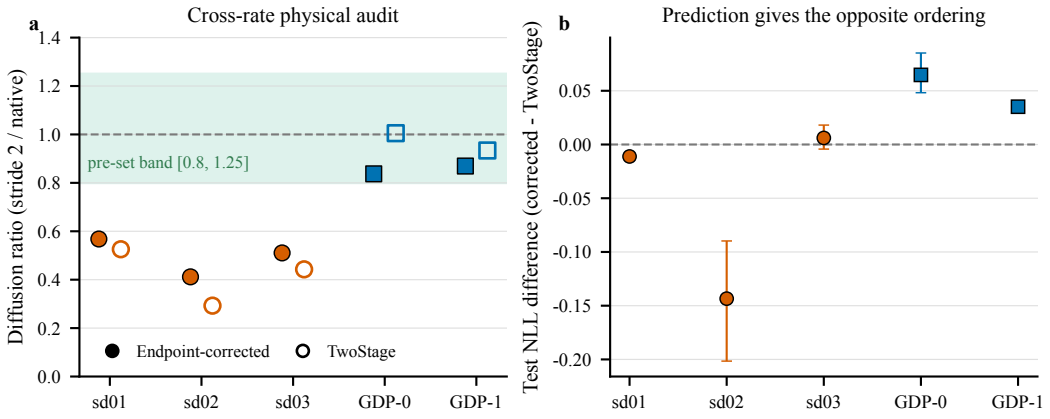


Figure 4: Sampling-rate stability and prediction on real data. **a**, Frozen point estimates of effective-diffusion ratios after re-identification at twice the sampling interval. Filled markers denote endpoint-corrected fits (validation-selected iterates for cells; two-dimensional add-back only for GDP), and open markers TwoStage. Circles are cell cohorts and squares are ocean-drifter coordinates. Shading is the pre-set $[0.8, 1.25]$ interval, not an uncertainty interval. **b**, Native-rate test NLL differences. Negative favors the corrected fit; error bars are 95% bootstrap intervals over cell pairs or the 20 frozen GDP test segments.

We now supplement these point-estimate decisions with paired full-refit bootstrap intervals, evaluating both fitted functions on a common native training-position grid (Appendix D.2). The audit cannot detect bias shared by both rates. In a 72-unit synthetic development diagnostic, 31/32 TwoStage fits with diffusion error above 25% still pass (Appendix D.3).

A neural audit control. A latent neural SDE passed all 26 units (16 primary plus 10 probes; $\rho_{\text{rate}} = 0.926$–$1.036$), with NLL medians within 0.045 of the EKF references. Some state-dependent fits retained 20–24% maximum edge error in σ_θ despite passing the audit (Table 7).

5.2 Sampling-dependent diffusion in cell trajectories

Figure 9 (Appendix C) illustrates the raw positions and reconstructed velocity on an unselected held-out sd01 trajectory.

Prediction favors the corrected fit in two of three cell cohorts. Native-rate test NLL differences (corrected minus TwoStage) are -0.0111 for sd01 and -0.1434 for sd02, with pair-bootstrap intervals excluding zero, but $+0.0061$ for the untouched sd03 replication, where the interval crosses zero (Figure 4b; Appendix C.5).

The rate audit reveals instability that predictive fit does not expose. Downsampling ratios for the endpoint-corrected fit are 0.568, 0.412, and 0.511, all below the pre-set 0.8 lower bound (Figure 4a). The corresponding TwoStage ratios are 0.526, 0.293, and 0.443. All three cohorts fail under both estimators. Coarse sampling with independent noise, fixed omitted partner coupling, and static parameter heterogeneity fail to reproduce this collapse in pre-registered stress tests. Matched reconstruction nulls also explain several initially suggestive residual diagnostics. A white-noise fit passed through the same smoother reproduces residual autocorrelation and partner-conditioned structure. The surviving deviation is short-lag residual variance.

The full-refit pair bootstrap gives 95% intervals $[0.450, 0.649]$, $[0.325, 0.484]$, and $[0.381, 0.580]$, respectively, all below the original lower bound (Appendix D.2).

Fast colored velocity forcing with correlation time near one sampling interval reproduces the collapse in simulation. However, even a correctly specified colored-force likelihood under-recovers the correlation time after downsampling, and a two-rate joint fit fails all 32 recovery units. For the estimators tested here, the evidence points to two obstacles. Coarse sampling loses information, and finite-step dynamics contaminate the noise estimate in equation 11.

5.3 RATE-STABLE FITS IN OCEAN DRIFTERS

The drifters provide a contrasting case: fitted velocity damping implies an integral time near 28 hours, well separated from hourly sampling. Point ratios 0.837 and 0.869 pass the audit, although TWOSTAGE predicts better on both coordinates (NLL differences $+0.0648$ and $+0.0352$; Figure 4). Appendix C.5 gives spatial coverage, prediction intervals, and a descriptive OU diffusivity conversion for comparison with Zhurbas & Oh (2004).

The corresponding TWOSTAGE audit ratios are 1.005 and 0.934, also inside the same interval and closer to one. Both estimators thus show the same cross-dataset stability pattern.

Full-refit resampling of the 32 training buoy IDs gives 95% intervals $[0.769, 0.974]$ and $[0.865, 1.096]$. The first interval crosses the lower bound, making its point-estimate acceptance less secure than the second coordinate's. Appendix D.2 reports full resampling results.

6 RELATED WORK

Dictionary methods learn deterministic dynamics, stochastic equations, and weak or Koopman-generator moments (Brunton et al., 2016; Boninsegna et al., 2018; Messenger & Bortz, 2021; Klus et al., 2020). Kramers–Moyal and likelihood approaches connect sampled trajectories to drift and diffusion (Kramers, 1940; Moyal, 1949; Gardiner, 2009; Aït-Sahalia, 2002), with measurement noise requiring correction (Böttcher et al., 2006). SFI and ULI provide related force/diffusion estimators (Frishman & Ronceray, 2020; Brückner et al., 2020); we benchmark their official implementations under matched dictionaries. Posterior second moments are standard in EM and state-space inference (Dempster et al., 1977; Särkkä & Solin, 2019). Our contribution is the explicit RTS endpoint correction for diffusion regression after reconstruction and its residual-error analysis; exact-discrete CAR(2) likelihood supplies a linear reference (Van Loan, 1978).

Integrated-diffusion estimators study exact position samples under long-span asymptotics with fixed (Ditlevsen & Sørensen, 2004) or vanishing (Gloter, 2006) sampling intervals. Latent-path methods provide alternative inference routes, including noisy indirect observations (Ferretti et al., 2020; Sutter et al., 2016). A matched numerical comparison with these approaches under our dictionaries and localization-noise protocol remains open. Online drift learning from error-free positions with supplied diffusion (Albrecht & Reich, 2026) addresses a different observation setting.

Latent force models (Álvarez et al., 2009), neural SDEs (Li et al., 2020; Kidger et al., 2021; Jia & Benson, 2019), and differentiable particle filters (Corenflos et al., 2021) learn hidden dynamics through alternative inference objectives. Our neural controls distinguish reconstruction quality, diffusion recovery, and rate stability.

7 DISCUSSION AND CONCLUSION

Posterior-mean reconstruction removes uncertainty needed to estimate stochastic forcing. Restoring the endpoint variance lowers diffusion error in all 72 frozen confirmation units while prediction scores barely change. Cross-rate re-identification extends evaluation to real trajectories: the contrasting cell and drifter results separate predictive fit from sampling-rate stability. Together, recovery tests and rate audits expose errors that prediction alone can miss.

Confirmation covers one-dimensional systems with iid Gaussian localization noise, a fixed sampling grid, supplied dictionaries, and offline smoothing. Relating surrogate-posterior and omitted-moment errors to parameter recovery remains open, as do general multidimensional state-dependent diffusion and robustness across sample size, observation density, noise family, and quantization. The drifter OU conversion is approximate, and its rate audit concerns an already interpolated product rather than raw measurements. These limits motivate reporting prediction, recovery, and rate stability separately.

AI USE STATEMENT

Generative AI tools were used to translate and restructure a pre-existing research draft, improve English exposition, organize the manuscript in the supplied LaTeX template, and assist with experiment, analysis, and plotting code. AI assistance also supported checks of mathematical derivations and proofs, literature positioning, interpretation of results, and feedback on experimental methodology. They were not used to fabricate experimental observations or numerical results. Reported values were computed by the documented simulations or analyses of released data and traced to saved local outputs; figures, equations, citations, and prose were checked against code and source materials. The authors take responsibility for the final content.

ETHICS STATEMENT

This work performs no new human-subject or animal experiment. The cell-collider trajectories and NOAA drifter trajectories were publicly released by their original data providers; our analysis uses only trajectory coordinates and associated metadata. The original cell study and the archived NOAA dataset record are cited for provenance.

REPRODUCIBILITY STATEMENT

Anonymous code, experiment inputs, and reference results are available at `https://anonymous.4open.science/r/OMWG-C-85D3/`. The supplementary code repository contains the estimator and baseline implementations, experiment configurations and seed ranges, numerical tests, reference result JSON files, and the real-trajectory inputs needed by the reported fits. Appendix B specifies the algorithm, Appendix C lists protocols and tables, and Appendix E maps reported results to executable entry points. The repository README gives full-run commands; a short verification script reruns one confirmation unit and checks its errors against the saved result.

REFERENCES

Yacine Aït-Sahalia. Maximum likelihood estimation of discretely sampled diffusions: A closed-form approximation approach. *Econometrica*, 70(1):223–262, 2002. doi: 10.1111/1468-0262.00274.

Jan Albrecht and Sebastian Reich. Parameter estimation for partially observed second-order diffusion processes. In *Stochastic Transport in Upper Ocean Dynamics IV*, volume 15 of *Mathematics of Planet Earth*, pp. 1–13. Springer Nature Switzerland, 2026. doi: 10.1007/978-3-032-12749-5_1.

Mauricio Álvarez, David Luengo, and Neil D. Lawrence. Latent force models. In *Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics*, volume 5 of *Proceedings of Machine Learning Research*, pp. 9–16, 2009. URL `https://proceedings.mlr.press/v5/alvarez09a.html`.

Lorenzo Boninsegna, Feliks Nüske, and Cecilia Clementi. Sparse learning of stochastic dynamical equations. *The Journal of Chemical Physics*, 148(24):241723, 2018. doi: 10.1063/1.5018409.

Frank Böttcher, Joachim Peinke, David Kleinhans, Rudolf Friedrich, Pedro G. Lind, and Maria Haase. Reconstruction of complex dynamical systems affected by strong measurement noise. *Physical Review Letters*, 97(9):090603, 2006. doi: 10.1103/PhysRevLett.97.090603.

David B. Brückner, Pierre Ronceray, and Chase P. Broedersz. Inferring the dynamics of underdamped stochastic systems. *Physical Review Letters*, 125(5):058103, 2020. doi: 10.1103/PhysRevLett.125.058103.

David B. Brückner, Nicolas Arlt, Alexandra Fink, Pierre Ronceray, Joachim O. Rädler, and Chase P. Broedersz. Learning the dynamics of cell–cell interactions in confined cell migration. *Proceedings of the National Academy of Sciences*, 118(7):e2016602118, 2021. doi: 10.1073/pnas.2016602118.

Steven L. Brunton, Joshua L. Proctor, and J. Nathan Kutz. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the National Academy of Sciences*, 113(15):3932–3937, 2016. doi: 10.1073/pnas.1517384113.

Adrien Corenflos, James Thornton, George Deligiannidis, and Arnaud Doucet. Differentiable particle filtering via entropy-regularized optimal transport. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 2100–2111, 2021. URL https://proceedings.mlr.press/v139/corenflos21a.html.

Arthur P. Dempster, Nan M. Laird, and Donald B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B*, 39(1):1–22, 1977. doi: 10.1111/j.2517-6161.1977.tb01600.x.

Susanne Ditlevsen and Michael Sørensen. Inference for observations of integrated diffusion processes. *Scandinavian Journal of Statistics*, 31(3):417–429, 2004. doi: 10.1111/j.1467-9469.2004.02_023.x.

Shane Elipot, Rick Lumpkin, Renellys C. Perez, Jonathan M. Lilly, Jeffrey J. Early, and Adam M. Sykulski. A global surface drifter data set at hourly resolution. *Journal of Geophysical Research: Oceans*, 121(5):2937–2966, 2016. doi: 10.1002/2016JC011716.

Shane Elipot, Adam Sykulski, Rick Lumpkin, Luca Centurioni, and Mayra Pazos. Hourly location, current velocity, and temperature collected from global drifter program drifters world-wide. NOAA National Centers for Environmental Information, Dataset, 2022. Subset accessed for 2016–2018.

Federica Ferretti, Victor Chardès, Thierry Mora, Aleksandra M. Walczak, and Irene Giardina. Building general Langevin models from discrete datasets. *Physical Review X*, 10(3):031018, 2020. doi: 10.1103/PhysRevX.10.031018.

Anna Frishman and Pierre Ronceray. Learning force fields from stochastic trajectories. *Physical Review X*, 10(2):021009, 2020. doi: 10.1103/PhysRevX.10.021009.

Crispin Gardiner. *Stochastic Methods: A Handbook for the Natural and Social Sciences*. Springer, 4 edition, 2009. URL https://link.springer.com/book/9783540707127.

Arnaud Gloter. Parameter estimation for a discretely observed integrated diffusion process. *Scandinavian Journal of Statistics*, 33(1):83–104, 2006. doi: 10.1111/j.1467-9469.2006.00465.x.

Junteng Jia and Austin R. Benson. Neural jump stochastic differential equations. In *Advances in Neural Information Processing Systems*, volume 32, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/59b1deff341edb0b76ace57820cef237-Abstract.html.

Rudolf E. Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82(1):35–45, 1960. doi: 10.1115/1.3662552.

Patrick Kidger, James Foster, Xuechen Li, and Terry J. Lyons. Neural SDEs as infinite-dimensional GANs. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 5453–5463, 2021. URL https://proceedings.mlr.press/v139/kidger21b.html.

Stefan Klus, Feliks Nüske, Sebastian Peitz, Jan-Hendrik Niemann, Cecilia Clementi, and Christof Schütte. Data-driven approximation of the Koopman generator: Model reduction, system identification, and control. *Physica D: Nonlinear Phenomena*, 406:132416, 2020. doi: 10.1016/j.physd.2020.132416.

Hendrik A. Kramers. Brownian motion in a field of force and the diffusion model of chemical reactions. *Physica*, 7(4):284–304, 1940. doi: 10.1016/S0031-8914(40)90098-2.

Xuechen Li, Ting-Kam Leonard Wong, Ricky T. Q. Chen, and David Duvenaud. Scalable gradients for stochastic differential equations. In *Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pp. 3870–3882, 2020. URL https://proceedings.mlr.press/v108/li20i.html.

Daniel A. Messenger and David M. Bortz. Weak SINDy: Galerkin-based data-driven model selection. *Multiscale Modeling & Simulation*, 19(3):1474–1497, 2021. doi: 10.1137/20M1343166.

J. E. Moyal. Stochastic processes and statistical physics. *Journal of the Royal Statistical Society: Series B (Methodological)*, 11(2):150–210, 1949. doi: 10.1111/j.2517-6161.1949.tb00030.x.

Natural Earth. 1:110m physical vectors: Land. Public-domain vector map data, 2026. URL https://www.naturalearthdata.com/downloads/110m-physical-vectors/. Accessed 2026-09-17.

Herbert E. Rauch, F. Tung, and Charlotte T. Striebel. Maximum likelihood estimates of linear dynamic systems. *AIAA Journal*, 3(8):1445–1450, 1965. doi: 10.2514/3.3166.

John Rice. Bandwidth choice for nonparametric regression. *The Annals of Statistics*, 12(4):1215–1230, 1984. doi: 10.1214/aos/1176346788.

Pierre Ronceray. Stochastic force inference (SFI). Zenodo software archive, 2026. URL https://doi.org/10.5281/zenodo.21532843. Version v2.0.3.

Simo Särkkä and Arno Solin. *Applied Stochastic Differential Equations*. Cambridge University Press, 2019. doi: 10.1017/9781108186735.

Tobias Sutter, Arnab Ganguly, and Heinz Koeppl. A variational approach to path estimation and parameter inference of hidden diffusion processes. *Journal of Machine Learning Research*, 17 (190):1–37, 2016. URL https://jmlr.org/papers/v17/16-075.html.

Charles F. Van Loan. Computing integrals involving the matrix exponential. *IEEE Transactions on Automatic Control*, 23(3):395–404, 1978. doi: 10.1109/TAC.1978.1101743.

Victor Zhurbas and Im Sang Oh. Drifter-derived maps of lateral diffusivity in the pacific and atlantic oceans in relation to surface circulation patterns. *Journal of Geophysical Research: Oceans*, 109 (C5):C05015, 2004. doi: 10.1029/2003JC002241.

APPENDIX

A PROOFS AND ADDITIONAL POSTERIOR-CORRECTION DETAILS

A.1 FULL-STATE MOMENTS AND THE IMPLEMENTED APPROXIMATIONS

Let $\mathcal{F}_t$ be the complete-state filtration, let Z_t be an $\mathcal{F}_t$-measurable instrument, and write $A_h = \int_t^{t+h} a(X_s)\,ds$. At the true coefficients and with sufficient integrability, the martingale and Itô-isometry conditions imply

$$\mathbb{E}[Z_t R_h] = 0, \qquad \mathbb{E}[Z_t(R_h^2 - A_h)] = 0. \tag{16}$$

Writing $p_*(X \mid Y)$ for the true state posterior, the observed-data version averages the entire products, $\mathbb{E}_Y\mathbb{E}_{p_*}[Z_t R_h \mid Y] = 0$ and $\mathbb{E}_Y\mathbb{E}_{p_*}[Z_t(R_h^2 - A_h) \mid Y] = 0$. Replacing these joint expectations by products of smoothed features and residual moments introduces additional covariance and nonlinear-feature errors. An RTS estimate at time t depends on observations after t; therefore $Z_t = \Phi(\mu_t)$ or $\Psi(\mu_t)$ is not in general an adapted instrument for the original innovation. Non-overlap within one lag does not remove this dependence, and blocks at different lags can overlap.

The smoother defines a surrogate posterior p_s, initially under a constant-velocity model and subsequently under a fitted affine model. The conditional identity in Proposition 1 holds under either law when its means and variances are evaluated consistently. The implemented correction approximates a moment under p_s; even completing that moment does not establish that $\mathbb{E}_{p_s}[R_h^2 \mid Y] = \mathbb{E}_{p_*}[R_h^2 \mid Y]$. Thus surrogate-posterior error, residual-moment truncation, feature and instrument substitution, finite-grid quadrature, and sampling error are distinct sources of estimation error.

The diffusion dictionary also contains a plug-in approximation. For the Gaussian marginal used here,

$$\mathbb{E}_{p_s}[Q^2 \mid Y] = \mu_Q^2 + P^{qq}, \qquad \mathbb{E}_{p_s}[Q^3 \mid Y] = \mu_Q^3 + 3\mu_Q P^{qq}. \tag{17}$$

The implementation uses μ_Q^2 and μ_Q^3. Correcting only the response does not correct the diffusion regressors or the joint instrumental products.

A.2 ENDPOINT DOMINANCE AND AFFINE FINITE-GRID FORMULA

Definition 1 (Endpoint-dominance diagnostic)**.** When $\mathrm{Var}(D \mid Y) > 0$, define

$$u_h = \sqrt{\frac{\mathrm{Var}(B \mid Y)}{\mathrm{Var}(D \mid Y)}}, \qquad \eta_h = \frac{|2\widetilde{R}_h\delta_h + \delta_h^2|}{\mathrm{Var}(D \mid Y)}. \tag{18}$$

Corollary 1 (Sufficient endpoint dominance)**.** *Under Proposition 1,*

$$\frac{|\mathbb{E}[R^2 \mid Y] - \widetilde{M}_h|}{\mathrm{Var}(D \mid Y)} \le \eta_h + u_h^2 + 2u_h. \tag{19}$$

Consequently, $\eta_h + u_h^2 + 2u_h < 1$ is sufficient for the restored endpoint variance to exceed the worst-case absolute omitted block.

The bound is sufficient but not necessary because covariance terms can cancel. Across 32,640 linear development blocks, the absolute omitted affine term divided by endpoint variance has median 0.022 and 90th percentile 0.032. On the 32,640 affine development blocks, u_h has median 0.027 and 90th percentile 0.041. All quantities concern the same specified posterior.

Proposition 2 (Affine-Gaussian finite-grid exactness)**.** *Let $b(x) = c_0 + c^\mathsf{T}x$ and $B_h = c_0 h + \sum_{j=0}^{m-1} c^\mathsf{T} X_{t_j}\Delta t_j$, using the same quadrature in R and $\widetilde{R}_h$. If the stacked grid state is conditionally Gaussian, $P_{ij} = \mathrm{Cov}(X_{t_i}, X_{t_j} \mid Y)$, and w_i are the residual weights, then $\delta_h = 0$, $R \mid Y$ is Gaussian, and*

$$\mathrm{Var}(R \mid Y) = \sum_{i,j} w_i^\mathsf{T} P_{ij} w_j, \qquad \mathbb{E}[R^2 \mid Y] = \widetilde{R}_h^2 + \sum_{i,j} w_i^\mathsf{T} P_{ij} w_j. \tag{20}$$

The confirmed estimator retains only the endpoint-velocity variance. Adopting the complete expression would change the estimator; the diagnostics below do not make that replacement.

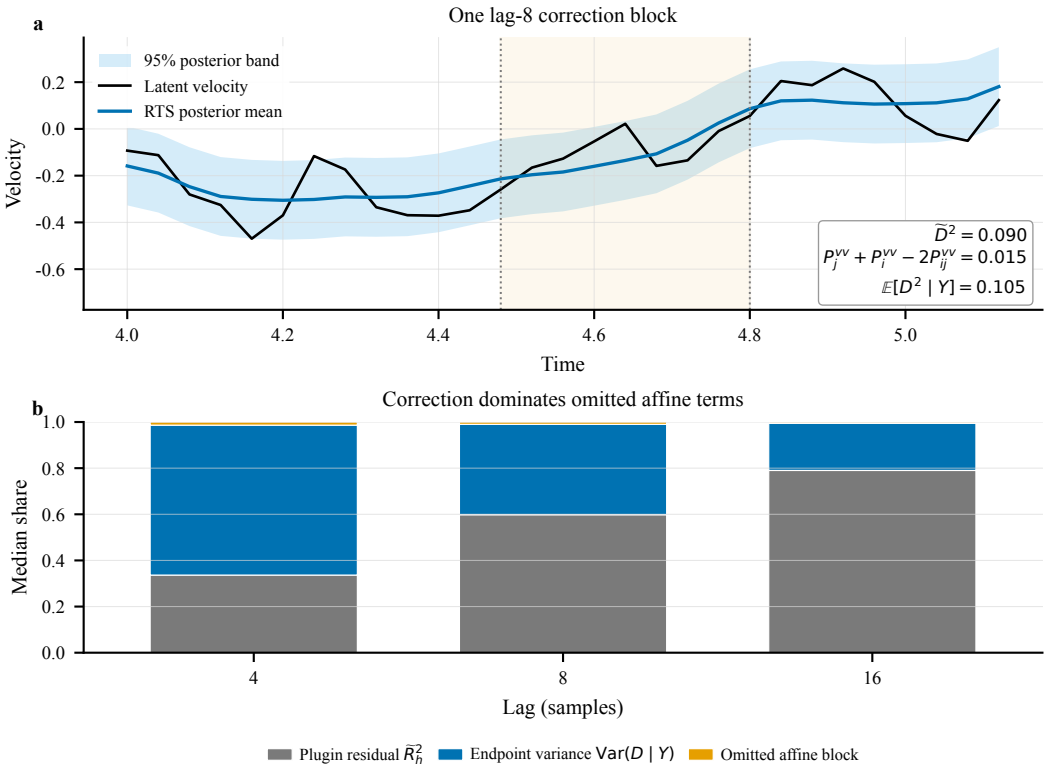


Figure 5: Endpoint correction and its scale. **a**, The shaded span marks one lag-8 block from linear development seed 128 at 2% localization noise. The posterior mean contracts a realized latent path; the endpoint identity restores $P_j^{vv} + P_i^{vv} - 2P_{ij}^{vv}$. **b**, Median block-wise shares at lags 4, 8, and 16 across 32,640 affine linear-system development blocks. Each block is normalized by $\widetilde{R}_h^2 + \mathrm{Var}(D \mid Y) + |\mathrm{Var}(B \mid Y) - 2\mathrm{Cov}(D, B \mid Y)|$; orange is the absolute omitted affine block.

A.3 Proof of the conditional residual identity

Proof of Proposition 1. Conditional total variance gives $\mathbb{E}[R^2 \mid Y] = \mathbb{E}[R \mid Y]^2 + \mathrm{Var}(R \mid Y)$. Bilinearity applied to $R = D - B$ gives $\mathrm{Var}(R \mid Y) = \mathrm{Var}(D \mid Y) + \mathrm{Var}(B \mid Y) - 2\mathrm{Cov}(D, B \mid Y)$. Substituting $\mathbb{E}[R \mid Y] = \widetilde{R}_h + \delta_h$ and subtracting Equation 5 proves the result. □

A.4 Proof of sufficient endpoint dominance

Proof of Corollary 1. Conditional Cauchy–Schwarz gives

$$|\mathrm{Cov}(D, B \mid Y)| \le \sqrt{\mathrm{Var}(D \mid Y)\mathrm{Var}(B \mid Y)}. \tag{21}$$

Apply the triangle inequality to Equation 8 and the displayed covariance bound. Dividing by $\mathrm{Var}(D \mid Y)$ turns $\mathrm{Var}(B \mid Y)/\mathrm{Var}(D \mid Y)$ into u_h^2 and the covariance bound into $2u_h$; Definition 1 supplies η_h. This proves Equation 19 and the strict inequality conclusion. □

A.5 Proof of affine-Gaussian finite-grid exactness

Proof of Proposition 2. Affine drift and fixed quadrature make R an affine functional $r_0 + \sum_i w_i^\top X_{t_i}$ of the stacked conditional Gaussian state. It is therefore conditionally Gaussian. Linearity of conditional expectation and use of the same quadrature on μ_s give $\delta_h = 0$. Expanding the variance of the linear functional yields $\sum_{i,j} w_i^\top P_{ij} w_j$; adding the squared conditional mean proves both identities in Equation 20. □

Equation 17 follows by writing $Q = \mu_Q + Z$ with centered Gaussian Z. Odd centered moments vanish and $\mathbb{E}[Z^2 \mid Y] = \mathrm{Var}(Q \mid Y)$.

A.6 Cubic residual diagnostic under the surrogate posterior

For $b(q, v) = c_0 + c_1 q + c_2 v + c_3 q^3$, let $\alpha_i = c_1 + 3c_3((\mu_i^q)^2 + P_{ii}^{qq})$. Gaussian moment identities give

$$\begin{aligned}\mathrm{Cov}_{p_s}(b_i, b_j \mid Y) &= \alpha_i \alpha_j P_{ij}^{qq} + \alpha_i c_2 P_{ij}^{qv} + c_2 \alpha_j P_{ij}^{vq} + c_2^2 P_{ij}^{vv} \\ &\quad + 18 c_3^2 \mu_i^q \mu_j^q (P_{ij}^{qq})^2 + 6 c_3^2 (P_{ij}^{qq})^3, \qquad (22)\\ \mathrm{Cov}_{p_s}(D, b_i \mid Y) &= \alpha_i \mathrm{Cov}_{p_s}(D, Q_i \mid Y) + c_2 \mathrm{Cov}_{p_s}(D, V_i \mid Y), \qquad (23)\\ \delta_h &= -\sum_i 3 c_3 \mu_i^q P_{ii}^{qq} \Delta t_i. \qquad (24)\end{aligned}$$

Summing with the quadrature weights supplies $\mathrm{Var}(B \mid Y)$ and $\mathrm{Cov}(D, B \mid Y)$, hence the complete discrepancy in Equation 8. This calculation includes the nonlinear conditional mean and moments through order six. It was checked against Monte Carlo draws from a correlated Gaussian path and against the affine limit.

We apply it to every fitted trajectory for Duffing development seeds 128–131, both noise levels, and lags 4/8/16, under the final surrogate Gaussian smoother. Table 1 reports $|\mathbb{E}_{p_s}[R^2 \mid Y] - \widetilde{M}_h| / \mathrm{Var}_{p_s}(D \mid Y)$. All 32,640 evaluated blocks have a ratio below one. These are dependent blocks, not independent experimental repetitions. The result supports endpoint dominance within this surrogate and grid. Appendix A.1 distinguishes this truncation error from the other sources of estimation error.

Table 1: Absolute omitted cubic correction relative to endpoint variance. Each noise level includes four seeds and 40 fit trajectories per seed.

Noise	Lag	Blocks	Median	90th percentile
2%	4	9440	0.027	0.049
2%	8	4640	0.035	0.067
2%	16	2240	0.037	0.064
5%	4	9440	0.031	0.064
5%	8	4640	0.046	0.102
5%	16	2240	0.059	0.126

A.7 Proof of finite-step chord-statistic bias

Proof of Lemma 1. Set the three observation times to $0, \Delta, 2\Delta$. Under the constant-coefficient model,

$$Q_s = Q_0 + V_0 s + \tfrac{1}{2} b s^2 + \sqrt{a} \int_0^s (s - u)\, dW_u.$$

The initial position and velocity cancel from the latent chord residual,

$$Q_\Delta - \tfrac{1}{2}(Q_0 + Q_{2\Delta}) = -\frac{b\Delta^2}{2} - \frac{\sqrt{a}}{2} \left\{ \int_0^\Delta u\, dW_u + \int_\Delta^{2\Delta} (2\Delta - u)\, dW_u \right\}. \qquad (25)$$

The two integrals have disjoint supports, hence zero covariance, and each has variance $\Delta^3/3$. The latent residual therefore has mean $-b\Delta^2/2$ and variance $a\Delta^3/6$. The independent localization term $\varepsilon_\Delta - (\varepsilon_0 + \varepsilon_{2\Delta})/2$ has variance $3r/2$. Thus

$$\mathbb{E}[e^2] = \frac{3r}{2} + \frac{a\Delta^3}{6} + \frac{b^2\Delta^4}{4}. \qquad (26)$$

Division by the equal-spacing denominator $3/2$ in Equation 11 proves Equation 12. Averaging overlapping chords changes their sampling variance but not this expectation. □

Table 2: Median $\widehat{r}/r$ in the frozen confirmation. Sampling variability and finite-step contamination both affect these estimates.

System	2% noise	5% noise
Linear	1.008	0.997
Duffing	1.000	0.996
State-dependent	1.008	0.997
Pooled	1.001	

B ALGORITHM

Algorithm 1 specifies the implemented one-dimensional confirmation path, called with `OMWGConfig(epochs=0)`. The raw-noise fallback is retained for reproducibility; it triggered in 0/72 confirmation units. No learned dictionary update is executed.

Algorithm 1 OMWG-C: endpoint-corrected diffusion with frozen drift

Require: Position trajectories $\mathcal{D}$ on a shared grid; $\Phi = [1, q, v, q^3]$, $\Psi = [1, q^2]$
Require: Lags $\mathcal{L} = (4, 8, 16)$, refits $K = 5$, ridge $\lambda = 10^{-8}$
Ensure: Drift $\widehat{\beta}$, diffusion $\widehat{\gamma}$, observation variance $\widehat{r}$, fallback flag
Split $\mathcal{D}$ by trajectory into fit and validation sets (40/8 in confirmation)
Estimate $\widehat{r}$ on fit positions using Equation 11
Select $s \in \{0.01, 0.03, 0.1, 0.3, 1, 3, 10\}$ by minimum filter NLL on the first 12 fit paths
Run constant-velocity RTS on all fit paths; retain (μ, P, J)
Fit baseline $(\widehat{\beta}, \widehat{\gamma}_{\text{base}})$ with lag-8 weak-IV moments; freeze $\widehat{\beta}$
if $\sqrt{\widehat{r}/\mathrm{Var}(Y_{\text{fit}})} > 0.10$ **then**
    return $(\widehat{\beta}, \widehat{\gamma}_{\text{base}}, \widehat{r}, \text{true})$
end if
$\widehat{\gamma} \leftarrow$ CORRECTEDFIT$(\mu, P, J, \widehat{\beta}, \mathcal{L})$
for $k = 1, \ldots, K$ **do**
    $s \leftarrow \max\{\widehat{\gamma}_0, 10^{-3}\}$
    Run linear RTS with drift $\widehat{\beta}_q q + \widehat{\beta}_v v$ and constant intensity s
    $\widehat{\gamma} \leftarrow$ CORRECTEDFIT$(\mu, P, J, \widehat{\beta}, \mathcal{L})$
end for
return $(\widehat{\beta}, \widehat{\gamma}, \widehat{r}, \text{false})$
function CORRECTEDFIT$(\mu, P, J, \beta, \mathcal{L})$
    Initialize empty row arrays Z, X, y
    for each path and each lag $\ell \in \mathcal{L}$ **do**
        for $i = 0, \ell, 2\ell, \ldots$ with $j = i + \ell < n$ **do**
            $h \leftarrow t_j - t_i$; $C_{ij} \leftarrow (J_i \cdots J_{j-1})P_j$
            $u \leftarrow P_i^{vv} + P_j^{vv} - 2C_{ij}^{vv}$
            $R \leftarrow \mu_j^v - \mu_i^v - \sum_{k=i}^{j-1} \Phi(\mu_k)\beta\, \Delta t_k$
            Append $\Psi(\mu_i)$ to Z, $h^{-1} \sum_{k=i}^{j-1} \Psi(\mu_k)\Delta t_k$ to X
            Append $(R^2 + u)/h$ to y
        end for
    end for
    $S \leftarrow \mathrm{diag}_d\big(\max\{\mathrm{mean}(X_{:,d}^2), 10^{-12}\}\big)$
    return $(Z^\top X + \lambda S)^{-1} Z^\top y$ ▷ implemented by a linear solve
end function

The full drift dictionary enters every residual, although refit smoothing uses only its linear terms. Specifically, its transition is $F_k = \begin{bmatrix} 1 & \Delta t_k \\ \widehat{\beta}_q \Delta t_k & 1 + \widehat{\beta}_v \Delta t_k \end{bmatrix}$, with process covariance from Equation 13 at the current s. RTS is exact for this discrete surrogate; the transition and covariance approximate the continuous-time linear model. The CAR(2) comparator instead uses exact continuous-to-discrete transitions and covariances. Validation does not select the confirmation iterate. Predictions use the diffusion floor in Equation 2. For real data, use the variants in Section 5: cells select add-back/refit candidates by validation NLL with baseline fallback; GDP uses two-dimensional add-back only.

Recalibrate and refit after stride-2 subsampling, replacing lags $(4, 8, 16)$ by $(2, 4, 8)$, and report Equation 15. Reconstruction-null diagnostics additionally simulate from the fitted null and repeat the same observation and reconstruction pipeline before comparison.

C Full Protocols and Additional Results

C.1 Synthetic data and one-shot gates

Every synthetic trajectory is generated at step 0.002, preceded by 1000 warm-up steps, and observed every 20 fine steps. Training-side data contain 48 trajectories; the last 8 are validation trajectories and the first 40 determine every calibration and fit. Independent fine paths generate the test occupancy distribution. Confirmation seeds 144–155 were evaluated once after the method and five gates were frozen. The gates assess median improvement, no systematic degradation, a per-condition floor, absolute diffusion dominance, and drift identity. All five passed. The bootstrap interval was computed after confirmation by resampling the 12 seeds as clusters, each contributing all six conditions.

The official linear ULI comparison was added after the method audit and is therefore development evidence, not part of the one-shot confirmation. It uses SFI 2.0.3 on seeds 128–139, receives the same 40 noisy fit trajectories and supplied dictionaries as the other position-only methods, and estimates localization noise internally. We convert ULI's convention $dV = F\,dt + \sqrt{2D}\,dW$ to our $a = 2D$ convention before scoring. All 72 fits are finite.

Table 3: Median observation NLL per predicted position on the corrected external-baseline grid. ULI uses its own train-position noise estimate. Lower is better.

Condition	GRU	ULI EKF	TwoStage EKF	OMWG-C EKF	CAR(2)
Linear 2%	-0.610	-2.447	-2.470	-2.472	-2.472
Linear 5%	-0.462	-0.390	-1.709	-1.713	-1.714
Duffing 2%	-0.998	-2.110	-2.256	-2.257	-2.237
Duffing 5%	-0.864	0.407	-1.462	-1.465	-1.446
State-dep. 2%	-0.312	-2.797	-2.818	-2.818	-2.819
State-dep. 5%	-0.170	-0.373	-2.057	-2.058	-2.059
Pooled	-0.550	-1.634	-2.181	-2.183	-2.173

C.2 Detailed recovery and baseline comparisons

The CAR(2) comparison shows where the nonlinear dictionary helps. On the linear oscillator, correctly specified CAR(2) has system-level median joint error 0.047 versus 0.079 for OMWG-C and wins 21/24 units. On Duffing, OMWG-C wins 24/24 units, with medians 0.088 versus 0.678 (Figure 6b). The methods are close on state-dependent diffusion, where CAR(2)'s occupancy-weighted constant approximation remains competitive. The correction is most useful here when the nonlinear dictionary captures dynamics outside the affine likelihood model.

These panels retain the condition-level joint-error view of confirmation, the comparison with affine CAR(2), and direct Duffing function recovery. They complement the main-text diffusion-error and mechanism figures.

ULI's pooled median joint error is 0.489 versus 0.093 for OMWG-C, which wins all 72 paired development units. Its internal localization-variance estimate is accurate in the aggregate (median $\widehat{r}/r = 1.003$). This comparison evaluates the specified closed-form ULI path, whose finite-difference corrections differ from our posterior-covariance correction after smoothing.

C.3 Neural development controls

Supervised neural reconstruction control. This control asks whether plug-in attenuation is specific to the RTS architecture. A 25,986-parameter bidirectional GRU with a linear state head was trained on ground-truth (q, v) targets from the 40 fit trajectories and early-stopped on eight validation trajectories. Its deterministic reconstructions were passed to the same lag-8 weak-IV backend, without an endpoint correction. Table 6 reports four-seed medians. Across all 24 units, the neural, RTS, and true-state median diffusion ratios were 0.716, 0.813, and 1.018; median fit-set velocity RMSE was

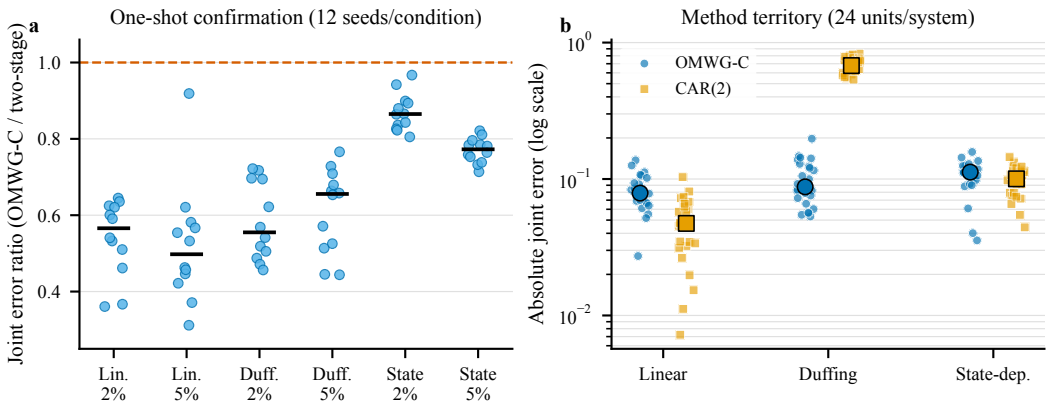


Figure 6: Synthetic evidence. **a**, Joint-error ratios for all 72 one-shot confirmation units; horizontal bars are condition medians and the dashed line denotes no improvement. **b**, Per-unit absolute joint errors for OMWG-C and exact-discrete CAR(2) on the post-audit development grid; large symbols are medians. CAR(2) benefits from correct specification on the linear oscillator, whereas its affine drift is misspecified for Duffing.

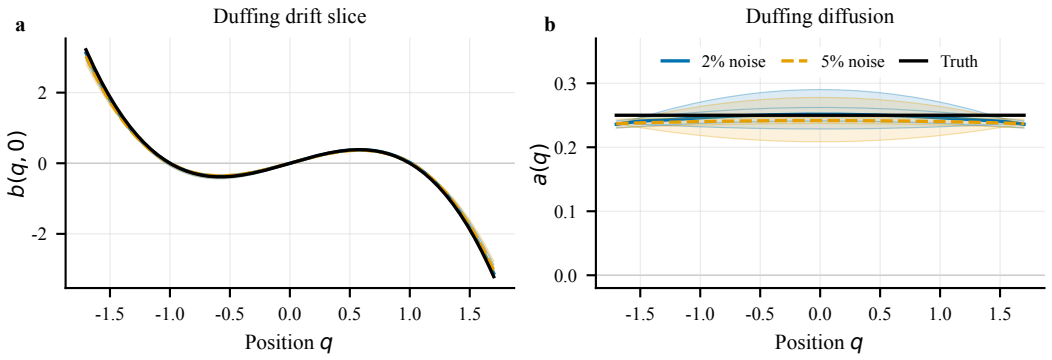


Figure 7: Recovered Duffing functions on the default-lag development grid. **a**, Drift slice $\widehat{b}(q, v = 0)$ against $b(q, 0) = q - q^3$. **b**, $\widehat{a}(q)$ against $a(q) = 0.25$. Thick colored curves are seed medians, faint curves are the four seeds, and shaded regions span their minima to maxima at each q; truth is black.

0.0977 for the neural reconstructor and 0.0920 for RTS. Every neural ratio was below one (range 0.561–0.883), and the ratio decreased from 2% to 5% noise for every matched system–seed pair. The two state-dependent seed-130 fits stopped earlier than the remainder (epochs 603 and 559); excluding them does not change the all-below-one conclusion.

Latent neural-SDE audit control. A separate control uses MLP drift and positive diffusion functions, a GRU initial-state encoder, and a differentiable extended Kalman filter. It minimizes teacher-forced one-step prediction NLL with four Euler–Maruyama substeps per observation interval; observation noise is fixed at the same train-only estimate $\widehat{r}$. Every rate is retrained independently. Table 7 separates the 16-unit primary grid from ten targeted probes. All runs pass the fixed audit interval, with individual ratios 0.926–1.036. Native-rate condition-median NLL differs from the corresponding EKF reference by at most 0.045. Two coarse-rate runs, both on seed 131, have substantially worse validation NLL than their condition peers and are retained. On the state-dependent probes, some learned σ_θ curves have 20–24% maximum relative error near an occupancy edge even though their occupancy averages pass the audit.

Table 4: Median absolute joint error on the corrected external-baseline development grid (12 seeds per condition). ULI is a post-audit development comparator. Bold marks the best position-only method in each row.

Condition	Diff. KM	GP+IV	ULI	TwoStage	OMWG-C	CAR(2)
Linear 2%	4.349	0.159	0.187	0.132	0.068	**0.046**
Linear 5%	20.425	0.270	0.610	0.190	0.085	**0.055**
Duffing 2%	9.733	0.221	0.494	0.144	**0.074**	0.590
Duffing 5%	34.094	0.341	1.042	0.226	**0.126**	0.752
State-dep. 2%	4.334	0.171	0.178	0.122	0.107	**0.095**
State-dep. 5%	20.228	0.279	0.600	0.154	0.114	**0.105**

Table 5: Matched component ablation on 72 development units. Drift is fixed. The ratio is the median of per-unit diffusion-error ratios relative to TwoStage. The final column counts strict improvements over TwoStage.

Diffusion estimator	Median e_a	Error ratio	Wins
Single lag, plug-in (TwoStage)	0.241	1.000	–
Multi lag, plug-in	0.353	1.402	0/72
Single lag, endpoint correction	0.105	0.386	62/72
Multi lag, endpoint correction	0.122	0.454	51/72
Multi lag, plug-in + five refits	0.489	2.051	0/72
Multi lag, correction + five refits (OMWG-C)	0.094	0.356	69/72

Table 6: Neural reconstruction control (development seeds 128–131). Entries are median occupancy-averaged estimated-to-true diffusion ratios; values below one indicate attenuation. The neural and RTS columns use the same uncorrected lag-8 plug-in backend.

Condition	Neural plug-in	RTS plug-in	True-state reference
Linear 2%	0.824	0.822	1.024
Linear 5%	0.734	0.806	1.024
Duffing 2%	0.708	0.799	1.012
Duffing 5%	0.596	0.624	1.012
State-dep. 2%	0.798	0.902	1.017
State-dep. 5%	0.685	0.809	1.017

Table 7: Latent neural-SDE audit. NLL is the native-rate condition median; "ref." is the corresponding EKF value in Table 3. The final column is native-rate occupancy-averaged diffusion divided by truth. Primary rows use four seeds; the stride-4 probe uses two.

Condition	Rates	Neural NLL	Ref. NLL	ρ_{rate}	$a_{\mathrm{eff}}/a_{\mathrm{true}}$
Linear 2%	1/2	-2.427	-2.472	0.949	1.019
Linear 5%	1/2	-1.707	-1.713	0.973	1.020
Duffing 2%	1/2	-2.253	-2.257	0.962	1.024
Duffing 5%	1/2	-1.473	-1.465	0.997	1.021
State-dep. 2%	1/2	-2.794	-2.818	0.940	1.018
State-dep. 5%	1/2	-2.062	-2.058	0.975	1.013
Duffing 5% probe	1/4	-1.478	-1.465	0.980	0.965

C.4 Development-only transfer and two-dimensional results

At unseen horizons 0.02, 0.06, and 0.08, transition-covariance error ratios are 0.717, 0.690, and 0.697. The shortest-horizon mean-error ratio is 1.149 and fails the pre-set 1.10 gate; we retain that failure. Long-run position total variation is nearly neutral (ratio 1.021), while velocity-autocorrelation and first-passage errors improve modestly (0.953 and 0.938). No simulated trajectory explodes.

The two-dimensional pilot uses state (Q_1, Q_2, V_1, V_2), drift features $[1, q_1, q_2, v_1, v_2, q_1^3, q_2^3]$, and diffusion features $[1, q_1^2, q_2^2]$. The add-back-only variant improves 30/32 constant-diffusion units, with pooled median diffusion-error ratios of 0.656 for the linear system and 0.635 for Duffing. It degrades all 16 non-polynomial state-dependent units (ratios 1.03–2.19). The state-aware refinement D divided by confirmed variant C has condition medians 1.218, 1.219, 0.991, 0.873, 0.770, and 0.796

Table 8: Transfer ratios, OMWG-C divided by TWOSTAGE. Values below one favor OMWG-C.

Category	Metric	Median ratio
Transition $h = 0.02$	mean / covariance error	1.149 / 0.717
Transition $h = 0.06$	mean / covariance error	0.973 / 0.690
Transition $h = 0.08$	mean / covariance error	0.931 / 0.697
Long time	stationary-position TV	1.021
Long time	velocity-ACF RMSE	0.953
Long time	first-passage CDF distance	0.938

for linear 2%, linear 5%, Duffing 2%, Duffing 5%, state-dependent 2%, and state-dependent 5%, respectively. Because 25/72 units degrade by more than 10%, variant D is rejected.

Table 9: Lag sensitivity on 24 development units (seeds 128–131). Ratios compare OMWG-C with TWOSTAGE; values below one favor OMWG-C.

Lag set	Median joint ratio	Median diffusion ratio	Joint wins
$(2, 4, 8)$	0.617	0.380	21/24
$(4, 8, 16)$	0.629	0.398	22/24
$(8, 16, 32)$	0.698	0.487	23/24

The improvement is stable to halving or doubling all diffusion lags, although longer lags weaken its magnitude. The five fixed refits also approach an empirical fixed point. Median relative changes in predicted diffusion after iterations 1–5 are 0.058, 0.037, 0.016, 0.006, and 0.003 (90th percentiles 0.159, 0.086, 0.046, 0.026, and 0.015). On the same 24 units, median wall-clock time is 0.70 seconds for TWOSTAGE and 1.95 seconds for the complete OMWG-C fit on an Apple-arm64 CPU under Python 3.12.7. These timing and sensitivity results are engineering diagnostics.

C.5 REAL-DATA COVERAGE AND RECONSTRUCTION

Prediction intervals and drifter scale. The cell cohorts first appear favorable under prediction. Native-rate test NLL differences (endpoint-corrected fit minus TWOSTAGE) are -0.0111 with 95% pair bootstrap interval $[-0.0150, -0.0075]$ for sd01, -0.1434 $[-0.2016, -0.0898]$ for sd02, and $+0.0061$ $[-0.0043, 0.0181]$ for the untouched sd03 replication. The predictive claim therefore holds in two of three cohorts.

The GDP admission code splits segments at timestamp gaps exceeding two hours; it does not threshold the product's source-interpolation gap field, which reaches 12 hours in retained segments. Fits use at most the first 720 hourly observations per segment. The drifters provide a contrasting case, with a fitted dynamical correlation time well separated from sampling. Figure 8 shows the spatial coverage and reconstructed circulation. The fitted velocity damping corresponds to an integral time near 28 hours, compared with one-hour sampling. Point ratios are 0.837 and 0.869, both inside the fixed interval. For scale only, an Ornstein–Uhlenbeck (OU) conversion gives spatial diffusivities of 7.4–$7.6 \times 10^3\,\mathrm{m^2 s^{-1}}$. Comparison with drifter-derived lateral diffusivity maps such as Zhurbas & Oh (2004) requires matched regions and estimation conventions. TWOSTAGE has better NLL on both coordinates. The corrected-fit minus TWOSTAGE differences are 0.0648 and 0.0352, with segment-bootstrap intervals $[0.0482, 0.0851]$ and $[0.0302, 0.0400]$, respectively. Across the two datasets, prediction favors the corrected fit in two cell cohorts where the audit fails. It favors TWOSTAGE on the drifters, where the point estimates pass.

Figure 8 retains the full admitted drifter coverage and the descriptive reconstruction map. Figure 9 shows the held-out cell trajectory used to illustrate the observation-to-velocity reconstruction before auditing.

C.6 REAL-DATA STRESS TESTS AND NULL CALIBRATION

An empirical audit of the released coordinates finds an approximately $0.648\,\mu\mathrm{m}$ quantization grid. This estimate comes from the coordinates rather than a microscope specification in the original

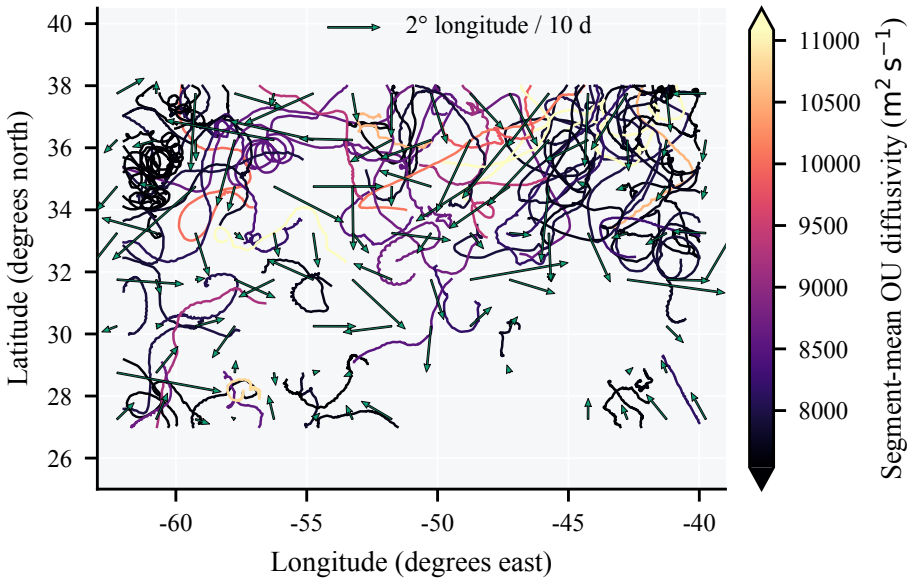


Figure 8: Spatial coverage and reconstructed circulation. All 94 admitted drogued GDP segments are colored by segment-mean OU diffusivity from the single frozen stride-1 global model. Dark or gray-looking tracks belong to the same color scale, not a separate or excluded group. Colors saturate below the 5th and above the 95th segment percentiles, indicated by the colorbar tips; no segments are discarded. Green arrows average segment-wise time means of RTS-reconstructed velocities within 1.5° cells, using segments with at least 24 hourly observations in that cell. Arrow length is the 10-day displacement; the key denotes 2° of longitude over 10 days. This field is a descriptive summary of reconstructed data, not an output of the fitted drift, which carries no position dependence. Segment means vary because the fitted diffusion is state-dependent; the value quoted in the text is the occupancy-averaged conversion. Coastlines use Natural Earth 1:110m land boundaries (Natural Earth, 2026).

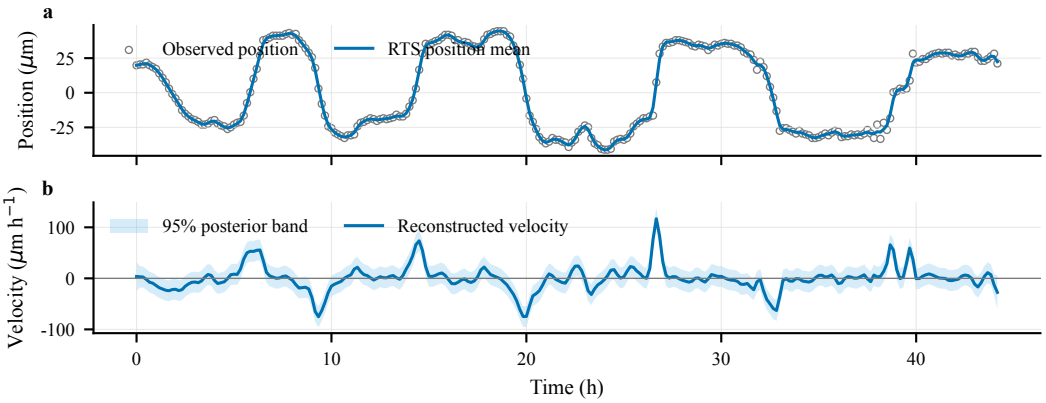


Figure 9: One held-out sd01 cell trajectory before physical auditing. **a**, Raw positions and the RTS posterior position mean. **b**, Reconstructed velocity mean with a pointwise 95% posterior band. The first cell in the first deterministic test pair is shown without outcome-based selection.

study. Early diagnostics found scale-dependent noise estimates, lag-one residual correlation, and partner-conditioned residual structure. Matched reconstruction nulls changed their interpretation. Pure iid or quantized observation models reproduce the apparent power-law noise statistic. Null residual correlations (0.68–0.79) exceed the real correlations (0.638/0.672). The null also reproduces

Table 10: Stress tests for the cell cross-rate collapse. The first three white-noise controls do not reproduce the observed 0.41–0.57 range; colored forcing can produce comparable attenuation but remains difficult to identify.

Factor	Cross-rate result	Verdict
Coarse sampling + 5% iid noise	median 0.981; 1/8 below 0.8	insufficient
Omitted symmetric partner coupling	median 0.998; 0/8 below 0.8	insufficient
Static drift/diffusion heterogeneity	medians 0.950–0.969	insufficient
Fast colored forcing, $\tau = \Delta t$	range 0.48–0.75	reproduces mechanism
Correct colored likelihood at stride 2	condition medians 0.549–0.833	recovery fails
Joint two-rate colored fit	32/32 outside recovery band	recovery fails

partner-conditioned curve shapes with correlations 0.97–1.00. Only the short-lag residual second moment remains qualitatively different from the null.

D Additional Matched Evidence and Audit Uncertainty

D.1 Matched component ablation

All six variants use the same 40 fit trajectories, train-only noise calibration, initial constant-velocity smoother, and frozen lag-8 drift. Single-lag diffusion uses lag 8; multi-lag diffusion uses 4/8/16. Both iterative variants perform five affine-smoother refits. One omits the endpoint correction at every update; the other includes it. There is no validation or truth-based selection among these variants. The single-lag plug-in fit reproduces the frozen TwoStage diffusion coefficients numerically. This new 72-unit grid uses development seeds 128–139, not the confirmation seeds. Historical early ablations used a different training-side split and are not pooled into this table.

Table 11: Condition-wise median diffusion-error ratios to TwoStage. Columns follow the six variants in Table 5.

Condition	Single PI	Multi PI	Single corr.	Multi corr.	PI + refits	OMWG-C
Linear 2%	1.000	1.506	0.394	0.451	1.684	0.303
Linear 5%	1.000	1.337	0.355	0.366	1.557	0.258
Duffing 2%	1.000	1.502	0.155	0.186	1.981	0.267
Duffing 5%	1.000	1.296	0.140	0.178	1.772	0.300
State-dep. 2%	1.000	1.401	0.747	1.115	2.288	0.755
State-dep. 5%	1.000	1.403	0.806	1.097	2.219	0.586

Table 12: Descriptive paired uncertainty and bias direction. Intervals resample the 12 seeds as clusters 10,000 times, retaining all six conditions per seed. The last column is the median occupancy-average diffusion divided by truth.

Variant	Error ratio	95% interval	$a_{\mathrm{eff}}/a_{\mathrm{true}}$
Single plug-in	1.000	[1.000, 1.000]	0.773
Multi plug-in	1.402	[1.345, 1.521]	0.652
Single corrected	0.386	[0.344, 0.415]	1.037
Multi corrected	0.454	[0.426, 0.479]	1.079
Multi plug-in + refits	2.051	[1.911, 2.194]	0.531
OMWG-C	0.356	[0.301, 0.448]	0.933

The endpoint term is the decisive component on this grid. Multi-lag pooling without it increases attenuation, and uncorrected refits amplify that effect. Single-lag correction is competitive and has lower pooled error than multi-lag correction without refits. Adding scales therefore does not improve every version of the estimator.

D.2 Uncertainty of the cross-rate audit

We supplement the historical point-estimate decisions with 400 paired cluster bootstrap refits per dataset (seed 20260919). For cells, we resample whole pairs independently within the frozen fit and validation partitions. We use the same sampled pairs at both rates. Every draw recalibrates observation noise, selects process intensity, reconstructs states, refits drift and diffusion, and repeats validation-based selection among add-back, five refits, and the baseline fallback. Test pairs remain untouched. The historical cell pipeline chooses an iterate by validation, whereas the synthetic confirmation uses five fixed refits; this distinction is retained.

For GDP, the 56 fit segments belong to 32 distinct buoys. We resample buoy IDs and keep all fit segments of each chosen buoy together. At both rates, we repeat noise calibration, intensity selection, vector drift fitting, and the frozen two-dimensional add-back fit. The 94 admitted segments come from 56 buoys overall. Treating all segments as independent would ignore repeated observations of the same buoy. Training buoy IDs do not overlap the validation or test IDs. One buoy appears in both validation and test partitions, but GDP uses neither validation-based fitting nor selection. The earlier NLL intervals resample test segments and should be read as segment-level descriptive intervals, not buoy-cluster intervals.

For each dataset, both rates are evaluated on the same fixed native-rate training-position grid, so the ratio compares fitted functions under a common occupancy measure. Unresampled refits reproduce the frozen coefficients numerically. This change in evaluation grid alters no reported point ratio at three decimals. The intervals are conditional on the observed reference grid and model family. The original band [0.8, 1.25] remains fixed.

Table 13: Full-refit audit uncertainty. In-band fraction is the fraction of bootstrap estimates within the fixed band; it is not a posterior probability that the model is physically correct.

Dataset	Fit clusters	Point ratio	95% interval	In-band fraction
sd01	54	0.568	[0.450, 0.649]	0.0%
sd02	93	0.412	[0.325, 0.484]	0.0%
sd03	51	0.511	[0.381, 0.580]	0.0%
GDP coord. 0	32	0.837	[0.769, 0.974]	90.8%
GDP coord. 1	32	0.869	[0.865, 1.096]	99.8%

There were 0 numerical failures among 1,600 bootstrap draws. All draw records and cell iterate-selection counts are retained in the exported statistics.

D.3 What passing the audit does and does not detect

Cross-rate stability cannot detect every bias. For example, multiplying both fitted diffusion functions by the same positive constant leaves ρ_{rate} unchanged under a common occupancy measure. Passing is therefore evidence of rate stability, not an absolute calibration test.

We additionally refit TwoStage and OMWG-C at stride 2 for all 72 development units, subsampling the identical noisy realizations and halving the lags to preserve physical durations. Both rates are evaluated on the same independent test-state occupancy, used only for diagnosis. Table 14 cross-tabulates the fixed audit decision with native-rate diffusion relative-L_2 error. The 10% and 25% error cutoffs are descriptive bins, not retrospectively selected acceptance gates or population error rates.

D.4 Additional matched nonlinear comparators

This predeclared small development grid uses all three systems, both noise levels, and seeds 128–129 (12 units). Function errors use independent test states generated with seeds offset by 100, the same occupancy-weighted relative-L_2 definitions, and joint error $\sqrt{e_b e_a}$.

SFI parametric path. In addition to the closed-form ULI estimator, we run SFI 2.0.3's `infer_force` and `infer_diffusion` with the same force and diffusion dictionaries and 40 fit trajectories. We retain the official defaults of RK4 with one substep, five outer force iterations, automatic errors-in-variables handling, and a 100-iteration diffusion optimization budget. Localization

Table 14: Known-error diagnostic for the fixed rate audit. Rates are re-identified independently; high error can coexist with rate stability.

Method	Native diffusion error	Units	Pass	Reject
TwoStage	$e_a \leq 0.10$	4	3	1
TwoStage	$0.10 < e_a \leq 0.25$	36	36	0
TwoStage	$e_a > 0.25$	32	31	1
OMWG-C	$e_a \leq 0.10$	40	40	0
OMWG-C	$0.10 < e_a \leq 0.25$	32	32	0
OMWG-C	$e_a > 0.25$	0	0	0

noise is estimated internally. We convert the diffusion convention from D to $a = 2D$. No truth-based hyperparameter tuning or post-hoc removal of terminal estimates is performed. All 12 runs yield finite fits. The force solver reports convergence in 11/12 runs; the remaining terminal estimate is retained. The pooled median joint errors are 0.052 for SFI parametric and 0.081 for OMWG-C, with SFI lower in 10/12 paired units.

Neural-SDE function recovery. The earlier audit artifacts contain summaries but no saved model weights. We therefore train a separate native-rate development run using the original configuration. It uses 64-unit drift/diffusion MLPs, a 32-unit GRU encoder, 300 epochs, four EKF substeps, and fixed train-estimated observation noise. We retain the original 500-epoch retry rule for unstable fits. We save all weights and evaluate the drift and squared diffusion amplitude directly on the same independent test states. Ground-truth states never enter training or model selection. Function recovery and the historical 26-unit cross-rate audit use separate runs. The new run uses 0 retries and retains 0 fits flagged unstable after the budget.

Table 15: Additional matched development comparators. Each condition contains two seeds; medians are descriptive. All three methods are scored on exactly the same test states.

Condition	Method	e_b	e_a	Joint
Linear 2%	OMWG-C	0.057	0.060	0.058
Linear 2%	Neural SDE	0.072	0.060	0.066
Linear 2%	SFI parametric	0.061	0.022	0.036
Linear 5%	OMWG-C	0.097	0.081	0.089
Linear 5%	Neural SDE	0.075	0.084	0.079
Linear 5%	SFI parametric	0.078	0.057	0.066
Duffing 2%	OMWG-C	0.073	0.063	0.068
Duffing 2%	Neural SDE	0.170	0.071	0.109
Duffing 2%	SFI parametric	0.059	0.021	0.035
Duffing 5%	OMWG-C	0.122	0.099	0.109
Duffing 5%	Neural SDE	0.176	0.103	0.133
Duffing 5%	SFI parametric	0.072	0.067	0.067
State-dep. 2%	OMWG-C	0.058	0.114	0.081
State-dep. 2%	Neural SDE	0.077	0.076	0.076
State-dep. 2%	SFI parametric	0.076	0.036	0.052
State-dep. 5%	OMWG-C	0.068	0.144	0.098
State-dep. 5%	Neural SDE	0.078	0.105	0.090
State-dep. 5%	SFI parametric	0.094	0.084	0.088

D.5 Full-grid SFI comparison and computational cost

After the initial 12-unit comparison, we extended SFI parametric to all 72 development units (seeds 128–139), retaining the initial fits and unchanged default settings. This extension was fixed before the additional 60 fits. It is development evidence, not a new confirmation. Drift, diffusion, and joint error are reported separately in Table 16; neither speed nor accuracy alone defines dominance.

All 72 SFI fits are finite; 5 force solves do not report convergence and their terminal estimates remain included. OMWG-C wins 30/72 in drift, 23/72 in diffusion, 21/72 in joint error.

Table 16: Complete 72-unit matched development comparison with 12 seeds per condition. Both methods use identical noisy fit trajectories, dictionaries, and independent test states.

Condition	Method	e_b	e_a	Joint
Linear 2%	OMWG-C	0.094	0.063	0.068
Linear 2%	SFI parametric	0.087	0.050	0.062
Linear 5%	OMWG-C	0.119	0.083	0.085
Linear 5%	SFI parametric	0.100	0.065	0.081
Duffing 2%	OMWG-C	0.089	0.068	0.074
Duffing 2%	SFI parametric	0.059	0.048	0.057
Duffing 5%	OMWG-C	0.136	0.117	0.126
Duffing 5%	SFI parametric	0.092	0.108	0.087
State-dep. 2%	OMWG-C	0.085	0.130	0.107
State-dep. 2%	SFI parametric	0.092	0.054	0.076
State-dep. 5%	OMWG-C	0.104	0.155	0.114
State-dep. 5%	SFI parametric	0.109	0.105	0.091

Timing and memory protocol. We benchmark the original 12 matched units in serial fresh processes on an Apple M5 Pro (18 CPU cores, 64 GiB RAM), using the same Python environment and float64 arithmetic. SFI's parametric solver already computes internally in float64. Its cost run also keeps inputs and returned coefficients in float64. The accuracy runs use the default session precision for inputs and returned coefficients. Each fit receives 40 trajectories of 240 positions. Data adaptation, train-only noise calibration, process-intensity selection where applicable, and all estimator iterations are timed. Imports, simulation, garbage collection before each fit, and test scoring are excluded. Host conversion synchronizes JAX results. Each fresh process performs three fits to identical data, initializing a new estimator each time. We report the first fit separately from the median of the next two. Any JIT work repeated by the public API remains included. Persistent JAX disk caching is disabled. Method order rotates between units, and no other project experiments run concurrently. We limit BLAS to one thread and also request JAX CPU threading limits. Internal runtime pools can still use multiple workers. CPU times and actual thread counts are retained in the exported records.

Peak resident memory is the operating system's maximum recorded process memory. It includes Python, method libraries, data preparation, and compilation caches, so it does not measure the estimator's minimal working memory. We also record pre-fit RSS to expose runtime overhead. The confirmed OMWG-C module imports PyTorch even though its frozen fit uses NumPy, and this overhead is retained. Table 17 reports medians across these matched units and includes the single-lag ablation.

Table 17: Precision and implementation cost on the matched 12-unit subset. Error columns are the original matched accuracy results; first/repeat wall time and repeat CPU time are seconds. Peak RSS is MiB and includes all three fits and runtime libraries.

Method	Joint	e_a	First	Repeat	CPU	Peak
TwoStage	0.128	0.235	0.63	0.63	0.63	239
Single corrected	0.078	0.102	0.77	0.77	0.77	240
OMWG-C	0.081	0.082	1.59	1.57	1.57	242
SFI parametric	0.052	0.052	3.90	1.77	7.51	663

Pre-fit RSS medians are 237 MiB (TwoStage), 237 MiB (Single corrected), 237 MiB (OMWG-C), 264 MiB (SFI parametric). Maximum observed process thread counts are 1 (TwoStage), 1 (Single corrected), 1 (OMWG-C), 92 (SFI parametric); allocated runtime threads are not necessarily simultaneously active. CPU time sums consumed processor time across threads, whereas wall time measures elapsed fit latency. The comparison retains native runtime parallelism on the same host.

Operation counts and scope. Let N be the number of observations, d the latent state dimension, and p the largest dictionary size. Let L, G, and K count the lags, process-intensity candidates, and diffusion refits, respectively. For dense state matrices, OMWG-C's filtering/smoothing and lagged-covariance products cost $O((G + (K + 1)(L + 1))Nd^3)$. Constructing moments and solving the small IV systems adds $O((K + 1)(NLp^2 + p^3))$. This is an upper bound on operation count.

Intensity selection uses at most 12 fit trajectories. Stored posterior means, covariances, and gains require $O(Nd^2)$ space, and materialized moment rows add $O(NLp + p^2)$. We do not store a dense covariance over all time pairs. Here $d = 2$, $p = 4$, $L = 3$, $G = 7$, and $K = 5$ are fixed. No particle population, neural training, or autodifferentiation is used in the confirmed estimator.

SFI's parametric path performs RK4 shooting, errors-in-variables force updates, and exact banded residual whitening, followed by diffusion likelihood optimization on cached force tensors. Its force API uses up to five outer iterations (inner budget 80), and diffusion optimization has a 100-iteration budget. With fixed state size, basis size, band width, and iteration budgets, its data passes also scale linearly in N. Automatic differentiation, cached tensors, compilation, and solver iterations affect constants and memory. OMWG-C combines an explicit, reusable posterior endpoint correction with small moment systems. The benchmark quantifies the resource savings of this implementation on the matched grid.

E Reproducibility Map

Run the following commands from the code repository root.

```
python -m pip install -r requirements.txt
python scripts/verify_results.py
python -m unittest discover -s tests -v
python scripts/check_fit.py
```

The first verification script checks input and result hashes and recomputes headline statistics. The final command simulates and fits one confirmation unit; `--system duffing` and `--system state_dependent` select the other systems. Full experiment commands and optional comparator dependencies are listed in the repository README.

- estimator and two-stage baseline: `position_only/omwg.py`, with `position_only/position_proxy.py` and `position_only/oracle.py`;
- confirmation, matched ablation, and position-only baselines: `experiments/run_confirmation.py`, `experiments/run_ablation.py`, and `experiments/run_baselines.py`;
- official ULI and SFI comparisons: `experiments/run_uli.py` and `experiments/run_sfi.py`; cost: `experiments/run_cost.py`;
- neural controls: `experiments/run_neural_reconstructor.py`, `experiments/run_neural_sde.py`, and `experiments/run_neural_sde_recovery.py`;
- cell and drifter studies: `experiments/run_cells.py`, `experiments/run_cell_replication.py`, and `experiments/run_drifters.py`;
- full-refit audit intervals and synthetic audit calibration: `experiments/run_audit_bootstrap.py` and `experiments/run_synthetic_audit.py`.

Reference outputs are under `results/`, with checksums in `results/manifest.json`. The confirmation is `results/confirmation/results.json`; component comparisons and audit intervals are under `results/comparisons/`. For baselines and drifters, `results/baselines/results_corrected.json` and `results/drifters/results_corrected.json` are authoritative. The confirmation uses `OMWGConfig(epochs=0)`: its train-only raw-noise gate triggers in 0/72 units, and the optional neural training loop is not run. The real-data inputs are the three released cell-coordinate files under `data/external/cell_collider/` and the 94 admitted drifter segments in `data/gdp/segments_480h.npz`. Their provenance and preprocessing are specified in the README and `experiments/prepare_drifters.py`.