WHERE SHOULD OBSERVATIONS ENTER? DUAL-PATH INTEGRATION FOR CONTINUING SENSING WITH RECURRENT GAUSSIAN STATES

Anonymous authors
Paper under double-blind review

ABSTRACT

Reconstructing dense unsteady flow fields from sparse near-body measurements is an underdetermined inverse problem, as most of the field is never directly observed. A common latent-state design for sparse flow reconstruction compresses the available measurements into a single latent state and decodes every requested coordinate from that state. Although efficient for temporal propagation, this design couples state updating with query readout: when the latent state is the decoder's sole interface to the observations, every query is decoded from the same fused summary. This raises a central design question: should observations enter the recurrent state, the query readout, or both? GAUSSASSIM addresses this question with two observation paths built around a recurrent anisotropic Gaussian field state. The state path assimilates observations into the Gaussian state for current decoding and subsequent propagation, while the query path uses time-indexed observation tokens to correct each readout without altering that state. On the ACDM *Inc* and periodic FlowBench benchmarks, GAUSSASSIM outperforms the compared learned baselines in mean reconstruction error. Across four evaluation settings, ten-seed factorial comparisons show that each observation path further reduces mean error when the other is active. All eight conditional gains have pointwise 95% bootstrap intervals above zero. Equal-token controls on NURBS show that distinct past observations contribute beyond repeated current frames, while spatial diagnostics associate larger query corrections with higher state-only error. Together, these results demonstrate complementary benefits from recurrent state assimilation and direct query-conditioned observation access in continuing-sensing flow reconstruction.

1 INTRODUCTION

Unsteady flows are rarely measured throughout their domain. In many diagnostic, monitoring, and control settings, a small set of probes near a solid body records local quantities such as velocity or pressure, while the surrounding field remains unobserved (Willcox, 2006; Manohar et al., 2018; Fukami et al., 2021). Reconstructing the full velocity–pressure field from these measurements is therefore an underdetermined inverse problem (Everson & Sirovich, 1995; Santos et al., 2023; Williams et al., 2024). We study continuing sensing: the same probe locations and measurement channels provide a new sparse snapshot at each target time, and the model must infer the full field from this evolving observation stream.

Each incoming snapshot can inform both the evolving field estimate and its reconstruction at a particular coordinate. Methods such as Senseiver, FLRONet, and STRIDE fuse measurements into a compact representation before decoding the requested field (Santos et al., 2023; Dang & Nguyen, 2025; Tong & Chen, 2026). In a recurrent reconstructor, such a representation carries information across time, but it need not be the best interface for every current query. If it is the decoder's sole observation interface, measurements reach each query only through their fused representation, coupling state estimation with coordinate-specific use at readout.

Prediction–correction methods use measurements to update the state estimate (Singh et al., 2024; Si & Chen, 2025; Chen et al., 2025), while sensor-conditioned decoders map encoded measurements

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.

to arbitrary coordinates (Santos et al., 2023; Tong & Chen, 2026). Continuous-PDE separates latent evolution from continuous queries starting from partial initial conditions (Janny et al., 2024). Continuing sensing adds a further choice: an arriving observation can influence subsequent recurrence, remain accessible to the current query, or serve both roles.

GAUSSASSIM provides both capabilities around a recurrent anisotropic Gaussian field (Figure 1). The state path assimilates each snapshot into coordinate-bearing primitives that support continuous field decoding and subsequent propagation. The query path retains a separate bank of observation tokens and uses each requested coordinate to retrieve a correction to the base prediction. This correction is not written back into recurrence. Gaussian primitives supply the spatial representation; the two interfaces determine how observations modify and supplement it.

Do these capabilities remain useful together? Ten-seed factorial comparisons on Reynolds-held-out ACDM *Inc* (Kohl et al., 2026) and three periodic FlowBench families (Tali et al., 2025) show that either path further lowers mean error when the other is active (Section 6). Matched readout controls on ACDM *Inc* and NURBS further show that query-conditioned observation aggregation outperforms a shared summary even when both decoders retain coordinate dependence. The evidence thus addresses both whether to retain the observation interface and how queries should use it; temporal and spatial analyses characterize its behavior. The contributions are threefold:

- We introduce a dual-path observation integration framework for continuing sensing that separates observation-driven state updates from direct query-time observation access.
- We propose GAUSSASSIM, a recurrent Gaussian-field method that implements this framework through repeated state assimilation and coordinate-conditioned correction from time-indexed observation tokens, without writing the correction back to the state.
- We conduct extensive experiments on ACDM *Inc* and periodic FlowBench, outperforming the compared learned baselines. Relative to the state-only model, mean reconstruction error decreases by 8.9% on ACDM *Inc* and 22.6% averaged across the three FlowBench families. Controlled studies further demonstrate the complementary benefits of the two observation paths and the advantage of query-conditioned aggregation.

2 RELATED WORK

Temporal sparse-field reconstruction and forecasting. ConvLSTM (Shi et al., 2015) and dynamic mode decomposition (Schmid, 2010) model temporal evolution, while neural operators (Lu et al., 2021; Li et al., 2021) learn maps between function spaces. For partial observations, delay coordinates (Takens, 1981) motivate finite sensor histories; SHRED (Williams et al., 2024; Ye et al., 2025) maps such histories to a field representation. FLRONet (Dang & Nguyen, 2025), FieldFormer (Bhardwaj et al., 2026), and STRIDE (Tong & Chen, 2026) combine temporal encoding with field or coordinate-based decoding; implicit neural fields (Luo et al., 2024) and stochastic NODE–DMD (Kim & Dean, 2026) address continuous reconstruction and uncertain dynamics. Continuous-PDE (Janny et al., 2024) evolves latent anchors from partial initial conditions and queries them through a continuous observer. Unlike open-loop forecasting, continuing sensing retains target-time measurements. GAUSSASSIM updates a recurrent field state from observation histories while retaining observation tokens for coordinate-specific reconstruction.

Recursive filtering and learned data assimilation. Kalman filtering (Kalman, 1960) and ensemble filtering (Evensen, 1994) combine model evolution with observation-based state updates. NODA (Singh et al., 2024) learns recursive prediction and correction operators for semilinear PDEs. Latent-EnSF (Si & Chen, 2025) aligns encoded states and sparse observations, LD-EnSF (Xiao et al., 2026) also learns latent evolution, and FlowDAS (Chen et al., 2025) uses stochastic interpolants for observation-conditioned state estimation. These methods use measurements for state estimation and subsequent evolution. GAUSSASSIM preserves this role and adds an observation-token correction to the current readout without altering the propagated state.

Query-conditioned field representations. Senseiver (Santos et al., 2023) reconstructs a field by querying a sensor-derived latent array, while STRIDE (Tong & Chen, 2026) conditions a continuous field decoder on a temporally encoded state. Both support coordinate-dependent outputs through a shared observation representation.

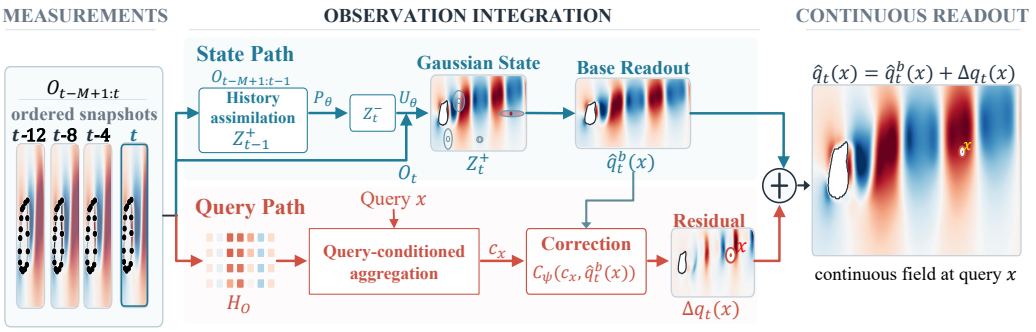

Figure 1: **Dual-path observation integration in GAUSSASSIM.** $O_{t-M+1:t}$ contains $M = 4$ snapshots of $q = (u, v, p)$ at $S = 16$ near-body probes. **State path (blue):** history assimilation summarizes initialization and recurrent updates over $O_{t-M+1:t-1}$ into Z^+_{t-1}. P_θ predicts Z^-_t; U_θ assimilates O_t to obtain Z^+_t, decoded as $\hat{q}^b_t(x)$. **Query path (red):** x aggregates time-indexed tokens H_O from the full window into a query-specific feature c_x; $C_\psi(c_x, \hat{q}^b_t(x))$ yields $\Delta q_t(x)$, added to the base readout without changing the state. Snapshot labels index stored frames (stride four); window and state indices count sampled observations. Black dots mark probes and ellipses Gaussian supports. Fields show v-channel quantities; input backgrounds provide context only.

Query-conditioned aggregation additionally lets each coordinate select features from encoded observations, as in attentive neural processes (ANPs) (Kim et al., 2019). ANPs combine this deterministic path with a global latent variable that models uncertainty over functions. In GAUSSASSIM, the shared representation is an observation-updated Gaussian field state that supports current decoding and is propagated to subsequent observation steps. A separate query path aggregates time-indexed observation tokens for each coordinate and corrects its output without updating that state.

3 PROBLEM FORMULATION

Let $\Omega_g \subset \mathbb{R}^2$ be the fluid domain associated with body geometry g, and let $q_t = (u_t, v_t, p_t) : \Omega_g \to \mathbb{R}^3$ denote the flow state at time t. For S sparse sensors at coordinates $X_s = \{x_s\}_{s=1}^S$, we observe

$$Y_{t-M+1:t} = \{y_{\tau,s} = q_\tau(x_s)\}_{\tau=t-M+1}^t. \tag{1}$$

Sensors remain fixed within each trajectory and sample the first layer of valid fluid cells adjacent to the body, giving *surface-adjacent* or *near-body* measurements.

The task is to estimate the target-time field at any query $x \in \Omega_g$:

$$\hat{q}_t(x) = F_\theta(x, X_s, Y_{t-M+1:t}, g, c). \tag{2}$$

The observation window includes measurements at t. Target-time observations define a continuing-sensing reconstruction task.

For variable geometries, g contains a body mask, signed distance, fluid fraction, and signed-distance gradients in the sensor/query coordinate frame. The primary tasks use no condition c; geometry-OOD supplies all methods with the known $c_{Re} = (\log Re - \log 100)/(\log 1000 - \log 100)$.

Each channel is normalized using training-trajectory statistics only: $\tilde{q}_c = (q_c - m_c)/s_c$. We evaluate reconstruction using

$$\mathrm{nrel}L_2(\hat{q}, q) = \frac{\left\| \mathrm{vec}(\tilde{\hat{q}} - \tilde{q}) \right\|_2}{\max\{\|\mathrm{vec}(\tilde{q})\|_2, \epsilon\}}. \tag{3}$$

With $\epsilon = 10^{-8}$, this training-normalized joint relative error combines all channels and valid fluid cells in one joint norm. We also report per-channel relative errors and raw-unit RMSE.

4 METHOD: DUAL-PATH OBSERVATION INTEGRATION

4.1 PROPAGATE, ASSIMILATE, DECODE, CORRECT

The state path propagates a shared Gaussian state and assimilates incoming observations into it; the query path supplements its readout with coordinate-specific observation correction (Figure 1). Below, measurements and fields use Section 3's channel normalization with tildes omitted; inverse normalization recovers physical outputs. Let $O_\tau = \{(x_s, y_{\tau s})\}_{s=1}^{S}$ denote one sparse snapshot. For variable geometry, a convolutional encoder maps the geometry grid to a global code $g_{\text{global}} = G_g(g)$; we write $\zeta = [g_{\text{global}}, c]$, omitting unavailable components. The four operations are

$$Z_\tau^- = P_\theta(Z_{\tau-1}^+, g, \zeta), \qquad Z_\tau^+ = U_\theta(Z_\tau^-, O_\tau, g, \zeta), \tag{4}$$

$$\hat{q}_t^b(x) = D_\theta(Z_t^+, x, g, \zeta), \qquad \hat{q}_t(x) = \hat{q}_t^b(x) + \Delta q_t(x). \tag{5}$$

The *state path* updates a shared posterior for current decoding and subsequent propagation. The *query path* computes $\Delta q_t(x)$ by attending to full-window observation tokens for each coordinate, without writing the correction back to that state. Recurrence spans one observation window ($M = 4$ in the primary experiments); overlapping windows initialize independently.

Both paths share snapshot tokens. In the geometry's normalized coordinate frame, an MLP E_s embeds raw coordinates, measurements, and local geometry; sensor self-attention A_s yields

$$(e_{\tau s})_{s=1}^{S} = A_s\big((E_s([x_s, y_{\tau s}, g(x_s)]))_{s=1}^{S}\big). \tag{6}$$

The state update consumes snapshots in temporal order, whereas the query route retains one token per sensor and lag and augments it with a learned lag embedding.

4.2 RECURRENT GAUSSIAN STATE

Building on localized basis functions and Gaussian field representations (Broomhead & Lowe, 1988; Kerbl et al., 2023; Xing et al., 2025), we organize the recurrent state as K anisotropic Gaussian primitives for assimilation and continuous decoding:

$$Z_\tau = \{(\mu_{\tau k}, \sigma_{\tau k}, a_{\tau k}, w_{\tau k}, h_{\tau k})\}_{k=1}^{K}, \tag{7}$$

where μ_k and σ_k encode center and axis-aligned support, $a_k \in \mathbb{R}^3$ is a normalized UVP amplitude, w_k is a mixture weight, and h_k is a learned feature. Mean pooling of the first snapshot's self-attended sensor tokens, followed by a context MLP and primitive head, initializes Z_{t-M+1}^+. At each later lag, primitive self-attention predicts a prior Z_τ^-; its features then cross-attend to the current sensor tokens to produce the posterior Z_τ^+. Both operations update the same packed primitive attributes through bounded residual maps, keeping centers, scales, and weights in valid ranges while allowing amplitudes and latent features to evolve. Exact maps and update bounds are given in Appendix B.

At query x, primitive support and normalized mixture weights are

$$\phi_k(x) = \exp\Big[-\tfrac{1}{2}\,\|(x - \mu_{tk}) \oslash \sigma_{tk}\|_2^2\Big], \qquad \alpha_k(x) = \frac{w_{tk}\phi_k(x)}{\sum_j w_{tj}\phi_j(x) + \epsilon}. \tag{8}$$

Their weighted amplitudes and features form a localized field scaffold,

$$q_G(x) = \sum_k \alpha_k(x) a_{tk}, \qquad h_G(x) = \sum_k \alpha_k(x) h_{tk}, \qquad m_G(x) = \sum_k w_{tk}\phi_k(x). \tag{9}$$

A continuous decoder uses Fourier query features $\gamma(x)$ (Tancik et al., 2020) to predict a residual around this scaffold:

$$\hat{q}_t^b(x) = q_G(x) + D_b(h_G(x), q_G(x), \log(m_G(x) + \epsilon), \gamma(x), g(x), \zeta). \tag{10}$$

The residual decoder D_b uses the Gaussian state and query features; the query correction below additionally reads observation tokens directly. We use $\epsilon = 10^{-8}$ in the mixture normalization and log mass for numerical stability.

4.3 QUERY-CONDITIONED OBSERVATION CORRECTION

The second interface preserves one token per sensor and lag. With normalized relative offset $\delta_\tau = (\tau - t)/\max\{M-1, 1\}$,

$$m_{\tau s} = P_m(e_{\tau s}) + P_t(\delta_\tau), \qquad H_O = \mathrm{Enc}_O(\{m_{\tau s}\}_{\tau,s}), \tag{11}$$

where Enc_O applies token layer normalization and lightweight self-attention (Vaswani et al., 2017). At each coordinate, cross-attention retrieves an observation feature for additive correction:

$$r_x = P_q(\gamma(x), g(x), \zeta), \tag{12}$$

$$c_x = \mathrm{LN}(r_x + \mathrm{CrossAttn}(r_x, H_O, H_O)), \tag{13}$$

$$\Delta q_t(x) = C_\psi(c_x, \hat{q}_t^b(x)), \qquad \hat{q}_t(x) = \hat{q}_t^b(x) + \Delta q_t(x). \tag{14}$$

Thus H_O preserves sensor–lag entries outside the Gaussian state, and the query determines which encoded evidence contributes to its correction.

Staged optimization. Let ℓ denote mean-squared error over sampled fluid queries and the three normalized channels. Base training minimizes

$$\mathcal{L}_{\mathrm{base}} = \ell(\hat{q}_t^b, q_t) + \lambda_G \ell(q_G, q_t),$$

optionally preceded by scaffold warm-up using $\ell(q_G, q_t)$ alone. We then freeze the validation-selected Base, including the shared sensor encoder, and train the query module to minimize $\mathcal{L}_{\mathrm{query}} = \ell(\hat{q}_t^b + \Delta q_t, q_t)$. Zero initialization of the correction output starts this stage exactly at the Base prediction, so query-specific corrections are learned against a fixed state readout. Schedules and λ_G are specified in Appendix B.

At inference, the model is called on batches of query coordinates; each call computes Z_t^+ and H_O from the M4 snapshots and shares them across the queries in that batch.

5 EXPERIMENTS

5.1 EVALUATION PROTOCOL

We organize the evaluation around three questions about how continuing observations are used. **Q1:** Does dual-path integration improve reconstruction across held-out intervals and trajectories? **Q2:** Does each observation path add value with the other active, and does query-conditioned aggregation improve the readout? **Q3:** Do past observations improve reconstruction beyond current-frame information and token count?

Tasks. We study two complementary continuing-sensing settings. The first studies periodic continuing-sensing reconstruction on the NURBS, Harmonics, and Skeleton families of Flow-Bench (Tali et al., 2025). Dense UVP fields are reconstructed at 512×128 from geometry-specific S16/M4 observations. A fixed pretraining screen on ground-truth v over frames $[60, 240)$ retains trajectories with best nonzero-lag correlation at least 0.8 and one-cycle discrepancy at most 0.4; this dense screen is never a model input. Contiguous 84/8/8% windows then train, validate, and test each retained geometry–Re trajectory with target-time sparse observations; the test intervals contain 1,816 NURBS, 1,944 Harmonics, and 1,064 Skeleton fields per seed.

The second uses the public ACDM *Inc* subset (Kohl et al., 2026) (denoted 128_{inc}): 91 PhiFlow trajectories (Holl & Thuerey, 2024) on a fixed cylinder geometry, spanning $Re = 100, 110, \ldots, 1000$ with 1,300 UVP snapshots each. Complete trajectories follow a fixed Reynolds-stratified 70/10/11 train/validation/test split. The sensing protocol supplies $S = 16$ fixed near-body sensors over $M = 4$ snapshots for dense (u, v, p) reconstruction on a 128×64 grid. The main evaluator scores all 6,039 eligible post-startup test fields per seed, measuring cross-trajectory reconstruction at nine interior and two endpoint Reynolds cases. Appendix A details data and sensor protocols.

Comparisons and evaluation. External comparisons include ConvLSTM (Shi et al., 2015), FieldFormer (Bhardwaj et al., 2026), S-NOT (Liu et al., 2026), FLRONet (Dang & Nguyen, 2025), Senseiver (Santos et al., 2023), and PySHRED (Ye et al., 2025) where reported. The cylinder PySHRED and Senseiver adapt released code; the remaining comparisons use local task adaptations, including architecture-inspired controls (Appendix C). We construct three controls with Gaussian-field decoding and no query correction: *Current-frame encoding* uses only O_t; *Joint-window encoding* processes $O_{t-M+1:t}$ jointly without recurrence; *Recurrent assimilation* propagates and updates a state across the window. Comparisons share available observations, normalization, splits, scored fields, and 45 training epochs, using three seeds unless noted. Field-output models train with channel-balanced MSE on 4,096 fluid queries; PySHRED uses MSE on 128 standardized POD coefficients for training and checkpoint selection. All models are scored by the same joint and channelwise normalized field errors.

5.2 Q1: Does dual-path integration improve reconstruction?

The dual path improves all three periodic families. Adding query correction to the recurrent Gaussian state lowers error from 0.1043 to 0.0777 on NURBS, from 0.1202 to 0.0929 on Harmonics, and from 0.1201 to 0.0961 on Skeleton (Table 1). The three-family average falls from 0.1149 to 0.0889, lower than those of the other compared learned methods. Each model is shared across its family's geometries. All three families are trained separately, replicating the design rather than transferring weights; Q2 tests the two observation factors in each family and on ACDM *Inc*.

Table 1: Periodic FlowBench (S16/M4): frame-equal normalized relative error on complete test intervals, mean $\pm$ sample SD over three seeds. Models are shared within families; Gaussian controls omit query correction. Lower is better. †Architecture-inspired local controls (Appendix C).

Method	NURBS	Harmonics	Skeleton	Average
Compared baselines				
ConvLSTM (Shi et al., 2015)	0.1823 ± 0.0057	0.1885 ± 0.0072	0.2224 ± 0.0070	0.1977
Senseiver (Santos et al., 2023)	0.1262 ± 0.0061	0.1536 ± 0.0221	0.1530 ± 0.0030	0.1442
FLRONet† (Dang & Nguyen, 2025)	0.1122 ± 0.0028	0.1226 ± 0.0031	0.1283 ± 0.0071	0.1210
FieldFormer† (Bhardwaj et al., 2026)	0.1071 ± 0.0043	0.1199 ± 0.0094	0.1293 ± 0.0103	0.1188
S-NOT† (Liu et al., 2026)	0.1093 ± 0.0016	0.1147 ± 0.0066	0.1199 ± 0.0069	0.1146
Gaussian controls (this work)				
Current-frame encoding	0.1353 ± 0.0031	0.1431 ± 0.0033	0.1595 ± 0.0048	0.1460
Joint-window encoding	0.1329 ± 0.0064	0.1289 ± 0.0096	0.1353 ± 0.0030	0.1324
Recurrent assimilation	0.1043 ± 0.0041	0.1202 ± 0.0027	0.1201 ± 0.0009	0.1149
GaussAssim	$\mathbf{0.0777 \pm 0.0010}$	$\mathbf{0.0929 \pm 0.0050}$	$\mathbf{0.0961 \pm 0.0112}$	**0.0889**

Figure 2 shows coherent wake phase and vortex spacing across the fixed cylinder and three shape families; reconstruction errors concentrate around thin shear layers and small wake features.

Accuracy extends to Reynolds-held-out trajectories. Table 2 reports complete-interval 128_{inc} results on Reynolds-held-out trajectories. GaussAssim has the lowest learned-model mean, 0.0195 ± 0.0026, versus 0.0214 ± 0.0015 for recurrent assimilation and 0.0256 ± 0.0047 for joint-window encoding. Errors decrease on all three channels relative to both controls; the largest reduction relative to joint-window encoding is on cross-flow velocity v. Paired crossed-bootstrap intervals support lower error than joint-window encoding and S-NOT; differences from current-frame encoding and recurrent assimilation are not resolved. Appendix D reports raw-unit, regional, differential-field, corruption, and efficiency diagnostics.

The two benchmarks therefore test different forms of reuse: a family model across held-out temporal intervals, and a fixed-geometry model across unseen Reynolds trajectories.

Extended settings. Self-advected restricted-ALE with query correction reaches a slightly lower periodic mean (0.0878 versus 0.0889), with a resolved gain only on Harmonics and higher cylinder error. Appendix E reports this propagation extension and pressure-only sensing; Appendix F examines geometry-OOD as a stress test.

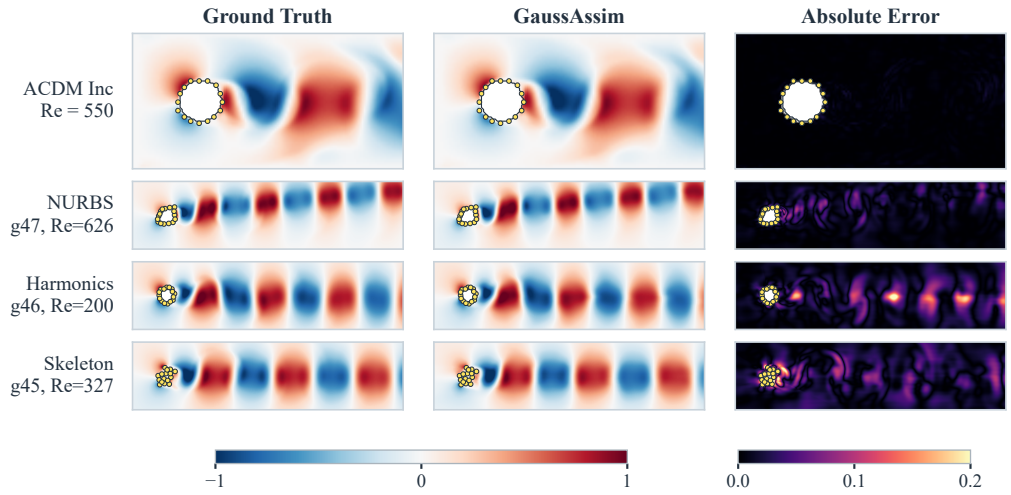

Figure 2: **Cross-dataset reconstruction.** Transverse velocity and error, normalized by the case's 99.5th percentile of $|v|$, with shared color scales and undistorted grid aspect ratios. Middle cylinder and median FlowBench test cases/frames were fixed before inspecting predictions. Yellow markers show the 16 sensors.

Table 2: Learning-based cross-trajectory reconstruction on ACDM *Inc* with S16/M4 observations. Training-normalized relative error over all 6,039 eligible test fields per seed; mean ± sample SD across seeds 123/456/789. Lower is better. †Architecture-inspired local controls (Appendix C).

Method	Joint	u	v	p	Params.
Compared baselines					
PySHRED (Ye et al., 2025)	0.0563 ± 0.0026	0.0511	0.0805	0.0395	0.46M
Senseiver (Santos et al., 2023)	0.0547 ± 0.0004	0.0590	0.0708	0.0205	1.34M
S-NOT† (Liu et al., 2026)	0.0254 ± 0.0015	0.0265	0.0312	0.0206	0.32M
FieldFormer† (Bhardwaj et al., 2026)	0.0227 ± 0.0004	0.0228	0.0273	0.0200	0.35M
Gaussian controls (this work)					
Current-frame encoding	0.0201 ± 0.0040	0.0197	0.0262	0.0167	0.92M
Joint-window encoding	0.0256 ± 0.0047	0.0251	0.0337	0.0215	1.03M
Recurrent assimilation	0.0214 ± 0.0015	0.0217	0.0262	0.0182	1.06M
GAUSSASSIM	**0.0195 ± 0.0026**	0.0200	0.0247	0.0164	1.28M

6 MECHANISM: HOW DO THE TWO PATHS CONTRIBUTE?

6.1 Q2: ARE BOTH PATHS USEFUL UNDER MATCHED INTERVENTIONS?

Across four settings, each path lowers mean error with the other active (Table 3, ten seeds). All cells initialize from the first observation; the factors enable assimilation at the remaining three lags and query correction. For each state-factor setting, query-on/off share a validation-selected Base30: query-on freezes it and trains correction for 15 epochs; query-off continues Base training for 15 epochs. Epochs, inputs, splits, and evaluation are matched, but parameter updates differ.

The complete model has the lowest case-equal mean in all four settings, and all eight conditional intervals in Figure 3 exclude zero. Both effects improve in 10/10 seeds per FlowBench family; ACDM *Inc* has 10/10 for query correction and 7/10 for assimilation, whose interval has a lower bound close to zero. Q1 uses three-seed, frame-equal comparisons; this factorial uses ten separately trained seeds and equal Re/geometry weights. Intervals resample seeds and cases after within-case frame averaging (Appendix A).

Table 3: **Two observation factors across four settings.** All cells use first-snapshot initialization; a enables repeated assimilation and b query correction. Values are case-equal mean ± sample SD over ten seeds (1–10). Cases/frames per model are 11/6,039 (ACDM *Inc*), 84/1,816 (NURBS), 83/1,944 (Harmonics), and 53/1,064 (Skeleton), covering each complete test interval.

a	b	ACDM *Inc*	NURBS	Harmonics	Skeleton
0	0	0.0251 ± 0.0028	0.1228 ± 0.0076	0.1336 ± 0.0059	0.1480 ± 0.0050
0	1	0.0216 ± 0.0025	0.0879 ± 0.0025	0.1007 ± 0.0045	0.1063 ± 0.0039
1	0	0.0241 ± 0.0021	0.0989 ± 0.0043	0.1158 ± 0.0076	0.1177 ± 0.0071
1	1	**0.0200 ± 0.0018**	**0.0779 ± 0.0045**	**0.0878 ± 0.0034**	**0.0895 ± 0.0049**



Figure 3: **Each path retains value with the other active.** Points and bars show case-equal error reductions and crossed seed×case 95% bootstrap intervals; positive values favor retaining the path. Counts indicate improving seeds. Intervals are pointwise, without multiple-comparison adjustment.

6.2 Q3: Do past observations add useful information?

Distinct past snapshots improve NURBS reconstruction beyond repeating the current measurement. All four three-seed controls train both paths with the inputs in Table 4. Repeated-current preserves M4's token count; past-only shifts the four-snapshot window back one step and predicts after one unobserved propagation step. Current-only and repeated-current have higher error than ordered M4, with paired intervals above zero. Past-only achieves 0.0739 mean error, demonstrating one-step periodic reconstruction from past measurements; its difference from M4 remains unresolved. On 128_{inc}, a three-seed, 220-field-per-seed control changes only query history while keeping M4 state inputs; M4, M1, and repeated-current show no resolved difference (Appendix D.1).

Table 4: NURBS temporal-information controls over complete intervals: geometry-equal mean ± seed standard deviation (three seeds, 84 geometries). Both paths train with the listed window. Δ is row minus ordered M4 under paired crossed seed×geometry resampling.

Input	Observations	Joint nrelL_2	Δ [95% interval]
Current only	O_t	0.0914 ± 0.0039	0.0153 [0.0117, 0.0196]
Repeated current	$O_t \times 4$	0.0844 ± 0.0033	0.0083 [0.0039, 0.0135]
Past only	$O_{t-16}, O_{t-12}, O_{t-8}, O_{t-4}$	**0.0739 ± 0.0024**	-0.0023 [-0.0052, 0.0015]
Ordered M4	$O_{t-12}, O_{t-8}, O_{t-4}, O_t$	0.0761 ± 0.0013	reference

6.3 Spatial and temporal roles

On sampled test windows, disabling repeated assimilation at inference increases joint error more at later M4 steps, with query correction off (Appendix D.5). Spatial state benefit is the reduction in pointwise normalized-channel error norm from initialization-only to repeated assimilation with query correction off; query benefit compares uncorrected and corrected outputs under repeated assimilation. Larger corrections coincide with higher Base error, with correlations of 0.748 on ACDM *Inc* and 0.738 on NURBS. The two paths' high-benefit regions overlap partially: their top-20% sets have Jaccard overlaps of 0.228 and 0.181, respectively, above the 0.111 independent-set reference.

6.4 QUERY-CONDITIONED OBSERVATION AGGREGATION

Does query-conditioned aggregation improve on a shared summary? Table 5 compares three readouts on a frozen 30-epoch Base (B0). N uses query features and B0 without tokens. S uses a learned coordinate-independent query to supply the same observation summary to every coordinate; Q aggregates tokens separately for each coordinate. Both S and Q retain coordinate-dependent decoding. S/Q share token-encoder and coordinate-decoder architectures, paired initializations, and training rules; effective parameter counts differ by less than 0.05%. N's trainable parameter count differs from Q's by less than 0.2%; all heads train for 15 epochs with the Base in evaluation mode.

Table 5: **Query-conditioned aggregation improves a fixed state readout.** Case-equal $\mathrm{nrel}L_2$ (mean $\pm$ sample SD, three seeds); 1,816 NURBS and 6,039 ACDM test fields per seed. The two prespecified $S-Q$ comparisons use paired seed$\times$case bootstrap intervals at 97.5% marginal coverage (Bonferroni adjustment). B0 is uncorrected, not the continued state-only model in Table 3.

Readout / contrast	NURBS	ACDM *Inc*
B0: frozen Base	0.11142 ± 0.00609	0.02957 ± 0.00427
N: no observation tokens	0.10090 ± 0.00675	0.02196 ± 0.00252
S: shared observation summary	0.08479 ± 0.00829	0.02210 ± 0.00220
Q: query-conditioned aggregation	$\mathbf{0.07940 \pm 0.00621}$	$\mathbf{0.02045 \pm 0.00226}$
$S-Q$	0.00539	0.00165
Adjusted interval	$[0.00297, 0.00803]$	$[0.00105, 0.00228]$

Q lowers S error by 6.36% on NURBS and 7.48% on ACDM *Inc*, with positive adjusted intervals and gains in 3/3 seeds per setting. Across-seed mean gains cover 82/84 geometries and 11/11 Reynolds cases. After averaging the three seeds within each case, median $S-Q$ gains are 0.00537 on NURBS and 0.00159 on ACDM *Inc* (Appendix D.3, Table 18). Positive case medians and broad improvement coverage jointly support gains across the evaluated cases. Q also outperforms N and B0; its advantage over S supports query-conditioned aggregation beyond coordinate-dependent decoding.

For the ten-seed factorial full-model checkpoints (Table 3), comparing corrected and uncorrected outputs holds the Gaussian state, inputs, and weights fixed. Correction lowers case-equal mean error in 10/10 seeds in all four settings (Appendix D.3, Table 19). After averaging the ten seeds within each case, gains are positive for every evaluated Reynolds case and geometry across all four settings.

7 DISCUSSION

The conditional gains show that, under the evaluated training protocols, observations remain useful at readout after state assimilation, while direct query access does not eliminate the benefit of repeated assimilation (Appendix D.2). The matched S/Q comparison identifies a further distinction between coordinate-dependent decoding and coordinate-dependent aggregation. A shared summary can produce spatially varying outputs, yet allowing each coordinate to aggregate its own observation feature yields lower error.

8 CONCLUSION

We introduced GAUSSASSIM for unsteady flow reconstruction from sparse near-body measurements under continuing sensing. Its dual-path design combines recurrent Gaussian state assimilation with query-conditioned observation access, separating updates carried into subsequent steps from corrections confined to the current readout. Experiments on ACDM *Inc* and periodic FlowBench demonstrate competitive reconstruction accuracy, with each observation path providing further gains when the other is active. Matched readout controls further show that query-dependent observation aggregation improves on a shared summary that already supports coordinate-dependent decoding. Together, these findings highlight the value of designing both where observations enter a recurrent reconstructor and how each query accesses them.

AI USE STATEMENT

Generative AI assisted coding, experimental design and analysis, and manuscript editing. The authors reviewed the outputs and take full responsibility for this work.

REPRODUCIBILITY STATEMENT

Appendices A–F describe data processing, experimental protocols, and implementation details. Code and configurations will be publicly released upon acceptance.

REFERENCES

Ankit Bhardwaj, Ananth Balashankar, and Lakshmi Subramanian. FieldFormer: Locality-aware transformers for spatio-temporal modeling on sparse sensor networks. *Transactions on Machine Learning Research*, 2026. ISSN 2835-8856. URL https://openreview.net/forum?id=we4FYGOE2y.

D. S. Broomhead and David Lowe. Multivariable functional interpolation and adaptive networks. *Complex Systems*, 2(3):321–355, 1988. URL https://www.complex-systems.com/abstracts/v02_i03_a05/.

Siyi Chen, Yixuan Jia, Qing Qu, He Sun, and Jeffrey A. Fessler. FlowDAS: A stochastic interpolant-based framework for data assimilation. In *Advances in Neural Information Processing Systems*, volume 38, pp. 103061–103102, 2025. doi: 10.52202/085713-3445. URL https://papers.nips.cc/paper_files/paper/2025/hash/951690f8dae19c76394404540b7ca734-Abstract-Conference.html.

Hiep Vo Dang and Phong C. H. Nguyen. Deep operator learning for high-fidelity fluid flow field reconstruction from sparse sensor measurements. *Journal of Computing and Information Science in Engineering*, 26(1):011007, 11 2025. ISSN 1530-9827. doi: 10.1115/1.4070332.

Geir Evensen. Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte Carlo methods to forecast error statistics. *Journal of Geophysical Research: Oceans*, 99(C5):10143–10162, 1994. doi: 10.1029/94JC00572. URL https://agupubs.onlinelibrary.wiley.com/doi/10.1029/94JC00572.

R. Everson and L. Sirovich. Karhunen–Loève procedure for gappy data. *Journal of the Optical Society of America A*, 12(8):1657–1664, 1995. doi: 10.1364/JOSAA.12.001657. URL https://opg.optica.org/josaa/abstract.cfm?uri=josaa-12-8-1657.

Kai Fukami, Romit Maulik, Nesar Ramachandra, Koji Fukagata, and Kunihiko Taira. Global field reconstruction from sparse sensors with voronoi tessellation-assisted deep learning. *Nature Machine Intelligence*, 3(11):945–951, 2021.

Philipp Holl and Nils Thuerey. Φ_{Flow}: Differentiable simulations for PyTorch, TensorFlow and Jax. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 18515–18546, 2024. URL https://proceedings.mlr.press/v235/holl24a.html.

Steeven Janny, Madiha Nadri, Julie Digne, and Christian Wolf. Space and time continuous physics simulation from partial observations. In *International Conference on Learning Representations*, pp. 21423–21446, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/5ce3a49415f78db65a714b4f05c62f4e-Abstract-Conference.html.

R. 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. URL https://doi.org/10.1115/1.3662552.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkuehler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4):1–14, 2023. doi: 10.1145/3592433.

Hyunjik Kim, Andriy Mnih, Jonathan Schwarz, Marta Garnelo, S. M. Ali Eslami, Dan Rosenbaum, Oriol Vinyals, and Yee Whye Teh. Attentive neural processes. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=SkE6PjC9KX.

Yujin Kim and Sarah Dean. Sparse-to-field reconstruction via stochastic neural dynamic mode decomposition. In *Proceedings of the 8th Annual Learning for Dynamics and Control Conference*, volume 331 of *Proceedings of Machine Learning Research*, pp. 1134–1156. PMLR, 2026. URL https://proceedings.mlr.press/v331/kim26b.html.

Georg Kohl, Li-Wei Chen, and Nils Thuerey. Benchmarking autoregressive conditional diffusion models for turbulent flow simulation. *Neural Networks*, 199:108641, 2026. doi: 10.1016/j.neunet.2026.108641. URL https://www.sciencedirect.com/science/article/pii/S0893608026001036.

Zongyi Li, Nikola Borislavov Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=c8P9NQVtmnO.

Qibang Liu, Weiheng Zhong, Seid Koric, and Hadi Meidani. Sequential neural operator transformer for high-fidelity surrogates of time-dependent non-linear partial differential equations. *Engineering Applications of Artificial Intelligence*, 172:114428, 2026. doi: 10.1016/j.engappai.2026.114428. URL https://www.sciencedirect.com/science/article/pii/S0952197626007098.

Lu Lu, Pengzhan Jin, Guofei Pang, Zhongqiang Zhang, and George Em Karniadakis. Learning nonlinear operators via deeponet based on the universal approximation theorem of operators. *Nature machine intelligence*, 3(3):218–229, 2021.

Xihaier Luo, Wei Xu, Balasubramanya Nadiga, Yihui Ren, and Shinjae Yoo. Continuous field reconstruction from sparse observations with implicit neural networks. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=kuTZMZdCPZ.

Krithika Manohar, Bingni W Brunton, J Nathan Kutz, and Steven L Brunton. Data-driven sparse sensor placement for reconstruction: Demonstrating the benefits of exploiting known patterns. *IEEE Control Systems Magazine*, 38(3):63–86, 2018.

Javier E Santos, Zachary R Fox, Arvind Mohan, Daniel O'Malley, Hari Viswanathan, and Nicholas Lubbers. Development of the senseiver for efficient field reconstruction from sparse observations. *Nature Machine Intelligence*, 5(11):1317–1325, 2023.

Peter J. Schmid. Dynamic mode decomposition of numerical and experimental data. *Journal of Fluid Mechanics*, 656:5–28, 2010. doi: 10.1017/S0022112010001217. URL https://www.cambridge.org/core/journals/journal-of-fluid-mechanics/article/abs/dynamic-mode-decomposition-of-numerical-and-experimental-data/AA4C763B525515AD4521A6CC5E10DBD4.

Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. In *Advances in Neural Information Processing Systems*, volume 28, 2015. URL https://papers.nips.cc/paper_files/paper/2015/hash/07563a3fe3bbe7e3ba84431ad9d055af-Abstract.html.

Phillip Si and Peng Chen. Latent-EnSF: A latent ensemble score filter for high-dimensional data assimilation with sparse observation data. In *International Conference on Learning Representations*, pp. 94879–94895, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/ecc276bd799a37e9b66508ed8f609d3d-Paper-Conference.pdf.

Ashutosh Singh, Ricardo Borsoi, Deniz Erdogmus, and Tales Imbiriba. Learning semi-linear neural operators: A unified recursive framework for prediction and data assimilation. In *International Conference on Learning Representations*, pp. 44092–44109, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/c031a1c2cae74a3d53a7f1e19638afe0-Abstract-Conference.html.

Floris Takens. Detecting strange attractors in turbulence. In *Dynamical Systems and Turbulence, Warwick 1980*, volume 898 of *Lecture Notes in Mathematics*, pp. 366–381. Springer, 1981. doi: 10.1007/BFb0091924. URL https://link.springer.com/chapter/10.1007/BFb0091924.

Ronak Tali, Ali Rabeh, Cheng-Hau Yang, Mehdi Shadkhah, Samundra Karki, Abhisek Upadhyaya, Suriya Dhakshinamoorthy, Marjan Saadati, Soumik Sarkar, Adarsh Krishnamurthy, Chinmay Hegde, Aditya Balu, and Baskar Ganapathysubramanian. Flowbench: A large scale benchmark for flow simulation over complex geometries. *Journal of Data-centric Machine Learning Research*, 2025. URL https://openreview.net/forum?id=0xrOmmB2KQ.

Matthew Tancik, Pratul Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. *Advances in neural information processing systems*, 33:7537–7547, 2020.

Yanjie Tong and Peng Chen. From sparse sensors to continuous fields: STRIDE for spatiotemporal reconstruction. *arXiv preprint arXiv:2602.04201*, 2026. URL https://arxiv.org/abs/2602.04201.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://papers.nips.cc/paper_files/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

K. Willcox. Unsteady flow sensing and estimation via the gappy proper orthogonal decomposition. *Computers & Fluids*, 35(2):208–226, 2006. doi: 10.1016/j.compfluid.2004.11.006. URL https://www.sciencedirect.com/science/article/pii/S0045793005000113.

Jan P. Williams, Olivia Zahn, and J. Nathan Kutz. Sensing with shallow recurrent decoder networks. *Proceedings of the Royal Society A*, 480(2298):20240054, 2024. doi: 10.1098/rspa.2024.0054. URL https://doi.org/10.1098/rspa.2024.0054.

Pengpeng Xiao, Phillip Si, and Peng Chen. LD-EnSF: Synergizing latent dynamics with ensemble score filters for fast data assimilation with sparse observations. In *International Conference on Learning Representations*, pp. 152461–152491, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/f7883c88d25e8cd0f05f06c92375febc-Paper-Conference.pdf.

Jingrui Xing, Bin Wang, Mengyu Chu, and Baoquan Chen. Gaussian Fluids: A grid-free fluid solver based on Gaussian Spatial Representation. In *ACM SIGGRAPH 2025 Conference Papers*, SIGGRAPH '25, New York, NY, USA, 2025. Association for Computing Machinery. doi: 10.1145/3721238.3730620. URL https://doi.org/10.1145/3721238.3730620.

David Ye, Jan Williams, Mars Gao, Stefano Riva, Matteo Tomasetto, David Zoro, and J. Nathan Kutz. PySHRED: A python package for shallow recurrent decoding for sparse sensing, model reduction and scientific discovery. *arXiv preprint arXiv:2507.20954*, 2025. URL https://arxiv.org/abs/2507.20954.

APPENDIX

A SUPPLEMENTARY OVERVIEW AND EXPERIMENTAL PROTOCOLS

Task and notation. For a fluid domain Ω_g and field $q_t = (u_t, v_t, p_t) : \Omega_g \to \mathbb{R}^3$, each reconstruction uses four ordered snapshots from 16 fixed, surface-adjacent sensors to estimate $q_t(x)$ at dense coordinates $x \in \Omega_g$. The cylinder window uses consecutive stored snapshots $(O_{t-3}, O_{t-2}, O_{t-1}, O_t)$; the FlowBench periodic protocol uses a four-frame lag stride, $(O_{t-12}, O_{t-8}, O_{t-4}, O_t)$. The window includes target-time measurements, so the primary task is continuing-sensing reconstruction rather than open-loop forecasting. Sensors, queries, geometry fields, and Gaussian centers share one normalized coordinate frame.

Method and controlled factors. GAUSSASSIM initializes 64 anisotropic Gaussian primitives from the first snapshot, alternates learned propagation with sensor-to-state assimilation over the remaining three lags, and decodes the final posterior into a base field. A separate query path reads all time-indexed observation tokens and adds a coordinate-specific residual that is not written back to recurrence. The matched 2×2 study varies initialization-only versus repeated assimilation, and no query correction versus query correction; all cells initialize from the first observation.

Metrics and statistical unit. After train-only channel normalization, the headline error is $\mathrm{nrel}L_2 = \|\mathrm{vec}(\tilde{\hat{q}} - \tilde{q})\|_2 / \max\{\|\mathrm{vec}(\tilde{q})\|_2, 10^{-8}\}$ over all UVP channels and valid fluid cells. Lower is better. Frames are first aggregated within each model-seed–test-case cell; crossed intervals then resample model seeds and test cases, grouped by Reynolds trajectory or geometry. FlowBench test intervals share geometry identities with training; the cylinder test trajectories have held-out Reynolds numbers.

A.1 DATASET SUMMARY

Table 6: Dataset and protocol summary. "Rows" denotes independent FlowBench geometry–Re trajectories.

Setting	Grid	Channels	Geom.	Rows	Split unit
128_{inc}	128×64	u, v, p	1	91	trajectory (70/10/11)
Periodic NURBS	512×128	u, v, p	84	227	time within row
Periodic Harmonics	512×128	u, v, p	83	243	time within row
Periodic Skeleton	512×128	u, v, p	53	133	time within row

A.2 CYLINDER-DATA PROVENANCE AND SPLIT

128_{inc} is the public Inc subset released with ACDM (Kohl et al., 2026), distributed through the mediaTUM record 10.14459/2024mp1734798. The archived metadata identifies PhiFlow 2.0.3 (Holl & Thuerey, 2024), a physical domain $[0, 4] \times [0, 2]$, a diameter-0.6 cylinder centered at $(1, 1)$, inflow speed 0.5, viscosity $\nu = 0.3/Re$, and source time step $\Delta t = 0.05$. Released 256×128 simulations are downsampled to 128×64.

We remove the first 200 stored frames, use frame stride two, and compute each channel's mean and standard deviation from valid fluid cells in the 70 training trajectories only. The 16 sensor coordinates are fixed surface-adjacent first-fluid-layer points and are identical across Reynolds trajectories because the geometry is fixed. The test set contains $Re = 100, 190, \ldots, 910, 1000$: nine cases lie inside the sampled range and the two endpoints test the range boundaries. The external comparison and ten-seed factorial both score every eligible post-startup test field (6,039 fields per model seed). The architectural and query-history controls in Appendix D.1 compare models on 20 fields per trajectory (220 fields per seed); the matched readout study in Appendix D.3 uses all 6,039 fields.

A.3 FLOWBENCH PROCESSING AND PERIODIC COHORT CONSTRUCTION

We use the public 1024×256 FlowBench release (Tali et al., 2025) and deterministically average each 2×2 block to 512×128 for flow and geometry arrays. The body mask, signed distance,

fluid fraction, and signed-distance gradients share one normalized coordinate frame with sensors and queries. For each geometry, candidate probes are fluid-cell centers in the first 8-neighbor layer adjacent to the solid mask; deterministic farthest-point sampling selects 16 approximately uniform candidates. The resulting geometry-specific coordinates are fixed across Reynolds numbers, frames, epochs, and splits and are normalized by the same grid extents as query coordinates.

Periodicity is measured on a spatially subsampled v field over frames $[60, 240)$. After subtracting the temporal mean, we retain a row when its best nonzero-lag normalized correlation is at least 0.8 and its one-cycle normalized discrepancy is at most 0.4. The resulting fixed criterion defines the cohort used throughout the periodic benchmark. All retained rows use contiguous 84/8/8% train/validation/test windows, target-time sparse measurements remain available, and the full-interval evaluator scores every eligible test frame. Dense fields supply supervised reconstruction targets, the deterministic cohort screen, and evaluation; they are never learned-model inputs.

A.4 Statistical aggregation

For the primary factorial and temporal-information tests, frames are first aggregated within each model-seed–test-case cell. Let e_{absg} denote this error for state factor $a \in \{0, 1\}$, query factor $b \in \{0, 1\}$, model seed s, and test case g. Tables 3 and 4 weight model seeds and test cases (Reynolds trajectories or geometries) equally, whereas Tables 1 and 2 use the primary frame-equal performance metric. For the ten-seed NURBS dual-path model, the same predictions give 0.07790 under geometry-equal weighting and 0.07947 under frame-equal weighting (Table 16). The external comparisons and temporal-information controls use separately trained seeds 123/456/789; their NURBS M4 reference gives 0.07612 geometry-equal and 0.07774 frame-equal error. Suppressing the seed/case indices, the factorial contrasts are

$$\Delta_{\mathrm{S}} = \tfrac{1}{2}[(e_{00} - e_{10}) + (e_{01} - e_{11})], \qquad \Delta_{\mathrm{Q}} = \tfrac{1}{2}[(e_{00} - e_{01}) + (e_{10} - e_{11})], \tag{15}$$

$$\Delta_{\mathrm{S}|Q} = e_{01} - e_{11}, \qquad \Delta_{\mathrm{Q}|S} = e_{10} - e_{11}, \tag{16}$$

$$I = e_{10} + e_{01} - e_{00} - e_{11}. \tag{17}$$

Positive Δ denotes error reduction and positive I denotes super-additivity. Point estimates average each contrast over all seed/case pairs; table standard deviations are sample standard deviations of the case-equal seed means. For each of 100,000 bootstrap draws, model-seed indices and test-case indices are independently sampled with replacement; the same sampled indices are applied to all four model cells before computing the paired estimand. The reported 95% interval is the 2.5–97.5 percentile range, computed separately for each contrast without multiple-comparison adjustment. The ten-seed study resamples ten model seeds and respectively 11, 84, 83, or 53 cases; bootstrap RNG seeds are 20260909–20260912 in cylinder/NURBS/Harmonics/Skeleton order, with batches of 2,000 draws. We report each seed's effect and the number of seeds with positive effects. The resulting empirical intervals quantify variation across retraining seeds and the finite test-case groups represented by each benchmark. Primary analyses use crossed estimates; group-only intervals are reported for secondary analyses.

B Implementation Details

B.1 State and query paths

The recurrent Gaussian models use 64 primitives with sensor and primitive widths of 96, one four-head primitive self-attention layer, one four-head sensor cross-attention layer, and three-layer MLPs for propagation and assimilation. The continuous decoder uses six-band Fourier coordinate features and a four-layer primitive-plus-residual readout (hidden width 192 on 128_{inc} and 160 on FlowBench). For numerical stability, Gaussian support $\phi_k(x)$ is evaluated using $(x - \mu_{tk}) \oslash (\sigma_{tk} + 10^{-6})$, with the offset added to each scale component. The query path has width 96, one four-head self-attention layer, four-head query cross-attention, and a zero-initialized additive output projection. The state is trained for 30 epochs; the selected seed-matched checkpoint is then frozen while the query path is trained for 15 epochs. Matched state-only and external learned controls use 45 total epochs. AdamW uses weight decay 10^{-4} and gradient-norm clipping at 1.0; 128_{inc} uses batch size 16 and learning rate 3×10^{-4} (query-stage rate 5×10^{-4}), while FlowBench uses batch size 4 and learning rate 5×10^{-4}. The cylinder model uses $[\sigma_{\min}, \sigma_{\max}] = [0.015, 0.28]$ and $\rho_\mu = 0.04$; FlowBench uses $[0.01, 0.35]$ and

$\rho_\mu = 0.03$. FlowBench encodes SDF, fluid fraction, and SDF gradients with a width-24 convolutional geometry encoder and a 64-dimensional global code; the fixed-cylinder task omits this branch. The snapshot encoder concatenates raw coordinates in $[0,1]^2$, normalized UVP measurements, and local geometry features, applies a three-layer MLP, and then one four-head sensor self-attention layer. It retains one token per sensor; mean pooling followed by a context MLP and primitive head initializes the Gaussian state from the first snapshot. Fourier features encode query coordinates in the field decoder and query-correction branch, not coordinates in the snapshot encoder. In the primary experiments, the global geometry code, where available, conditions the state and readout modules; Reynolds-number conditioning is used only in the geometry-OOD setting. In the primary target-time windows, the query branch embeds relative offsets $\delta_\tau = (\tau - t)/\max\{M-1, 1\}$ in sampled-observation indices, giving $[-1, -2/3, -1/3, 0]$ for M4 and zero for M1. The projected sensor tokens and lag embeddings are added, layer-normalized, and passed through token self-attention before query cross-attention.

Training objectives. All losses use channel-normalized fields, with ℓ denoting MSE over sampled fluid queries and channels. On 128_{inc}, the first five Base epochs train the scaffold alone using $\ell(q_G, q_t)$, with residual decoding disabled; epochs 6–30 minimize $\ell(\hat{q}_t^b, q_t) + 0.1\ell(q_G, q_t)$. On periodic FlowBench, recurrent Base training enables residual decoding from the first epoch and uses $\lambda_G = 0$, minimizing $\ell(\hat{q}_t^b, q_t)$. After selecting the Base checkpoint by validation, the query stage freezes the Base and shared sensor encoder and minimizes $\ell(\hat{q}_t^b + \Delta q_t, q_t)$ for 15 epochs. The scaffold term retained in the cylinder training implementation is constant with respect to query parameters and does not affect this optimization. State-only continuation instead updates Base parameters for 15 more epochs using the corresponding Base objective.

Table 7: **Architecture and training configuration of the recurrent Gaussian models.** State/query controls use these settings with the modifications specified in their experimental protocols.

Component	Configuration
Observation window	16 surface-adjacent sensors, four ordered snapshots, normalized UVP values and coordinates
Recurrent state	64 anisotropic Gaussian primitives; 96-dimensional sensor and primitive features
Continuous readout	Fourier coordinate features with Gaussian primitive scaffold plus learned residual
Query correction	Width 96; one four-head token self-attention layer; four-head query cross-attention; additive zero-initialized output
Spatial supervision	4,096 sampled fluid coordinates per training field; channel-balanced UVP loss
Optimization	AdamW, weight decay 10^{-4}, gradient-norm clipping at 1.0; batch/LR: $16/3 \times 10^{-4}$ on 128_{inc}, $4/5 \times 10^{-4}$ on FlowBench
Staged schedule	30 state-model epochs, then 15 epochs for the correction path with the selected state checkpoint frozen

B.2 Window-level computation

Table 8 makes the information flow explicit for one M4 reconstruction window. The query correction is computed only after the final posterior state has been formed and is never written back to recurrence. Each window starts from initialization, including overlapping windows.

Writing one packed primitive as $s_k = [\mu_k, \sigma_k, a_k, w_k, h_k]$, both propagation and assimilation apply the following bounded attribute map to their predicted residuals:

$$\mu_k' = \text{clip}_{[0,1]^2}(\mu_k + \rho_\mu \tanh \delta_{\mu,k}), \qquad \sigma_k' = \text{clip}_{[\sigma_{\min}, \sigma_{\max}]^2}\left(\sigma_k \odot e^{0.08 \tanh \delta_{\sigma,k}}\right), \tag{18}$$

$$a_k' = a_k + 0.15 \tanh \delta_{a,k}, \qquad w_k' = \text{clip}_{[10^{-4},1]}(w_k + 0.08 \tanh \delta_{w,k}), \tag{19}$$

$$h_k' = \text{LN}(h_k + 0.20 \tanh \delta_{h,k}). \tag{20}$$

Here ρ_μ is the configured maximum center step and can be attenuated for sensor assimilation. Propagation and assimilation predict separate residuals from the packed state, global conditioning, and, when enabled, geometry sampled at the current centers.

Table 8: **Window-level GaussAssim computation.** The state path changes the posterior used by later lags; the query path changes only the requested field values.

Step	Operation
1	Encode the ordered observations $O_{t-M+1:t}$ into per-sensor, per-lag tokens and retain the complete token bank H_O.
2	Initialize the Gaussian posterior Z^+_{t-M+1} from the first sparse snapshot.
3	For each later lag τ, propagate $Z^+_{\tau-1}$ to a prior Z^-_τ and assimilate O_τ to obtain Z^+_τ.
4	Decode the final posterior Z^+_t at all requested coordinates to produce the base field $\hat{q}^b_t(x)$.
5	Retrieve a coordinate-specific residual from H_O and return $\hat{q}_t(x) = \hat{q}^b_t(x) + \Delta q_t(x)$.

For the ten-seed factorial, the first sparse snapshot initializes all cells. The state factor controls assimilation after propagation at the remaining lags; the query factor enables direct token correction at readout. For each seed, cells 00/01 share one validation-selected Base30 checkpoint, and cells 10/11 share another. Query-on freezes the Base and optimizes correction; query-off starts from those same Base weights and optimizes the Base, each for 15 further epochs. Checkpoint selection uses validation only within the second stage; all 160 training runs finish before full-interval test evaluation. Cylinder continuation uses learning rate 3×10^{-4} and correction 5×10^{-4}; both FlowBench branches use 5×10^{-4}. Query-off starts with a fresh AdamW optimizer; cylinder query-on also resets its optimizer at the stage boundary, while FlowBench query-on retains its optimizer and freezes Base parameters. Thus 45 training epochs are matched, not computational cost, wall time, or parameter-update trajectories.

C Baseline Implementations

Gaussian controls introduced in this work. *Current-frame encoding* uses only the target-time sensor snapshot: a pointwise MLP and sensor self-attention produce a pooled context, from which a primitive head generates a Gaussian field for coordinate-conditioned decoding. *Joint-window encoding* processes all sensor–lag tokens, including coordinates, measurements, and lag offsets, with an MLP and two Transformer layers; mean pooling and a context MLP supply its primitive head. Both use the Gaussian field decoder and the configured geometry and global conditioning features. Neither propagates a Gaussian state between observations; *Recurrent assimilation* instead alternates propagation and observation updates within the window, without query correction. All three decode from their Gaussian field without the direct observation-token readout used by GaussAssim. These comparisons distinguish current-frame encoding, joint-history encoding, and recurrent assimilation, with differences in both temporal input use and architecture.

Implementation provenance. PySHRED and Senseiver in the cylinder comparison use released implementations with task adaptation to the fixed S16/M4 UVP protocol. PySHRED uses the released LSTM, MLP, and SHRED classes (Ye et al., 2025): each lag supplies 48 sensor values, the decoder predicts 128 POD coefficients, and the inverse POD map produces the dense field. The POD basis, sensor scaling, and coefficient scaling are fitted on training trajectories only. Both training and validation use mean squared error over the 128 standardized coefficients; the checkpoint with the lowest validation coefficient MSE is retained. After inverse scaling and POD reconstruction, the same field-space test metrics as for the other methods are computed. Senseiver retains its pinned official Encoder and Decoder (Santos et al., 2023); the adapter concatenates UVP values with Fourier features of sensor coordinates and lag, flattens the 64 sensor–lag tokens, and queries the decoder at target-time coordinates. It adds no geometry or Reynolds-number conditioning on the fixed cylinder.

Cylinder configurations and selection. The cylinder adapters use fixed configurations across seeds 123/456/789. Senseiver uses 256 latent slots of width 152; PySHRED uses a two-layer LSTM of width 128 and an MLP with hidden widths 350/400. Both train for 45 epochs with batch size 16 and AdamW weight decay 10^{-4}; the learning rates are 3×10^{-4} for Senseiver and 10^{-3} for PySHRED. Checkpoint selection uses 32 validation samples from each of the ten validation trajectories. Senseiver minimizes the trajectory-equal mean of sampled-query normalized relative error, selecting epochs 35/42/43 with validation errors 0.05411/0.05372/0.05395. PySHRED minimizes standardized-coefficient

MSE, selecting epochs 40/38/38 with values 0.06391/0.06096/0.06339. These validation values use different objectives; the external comparison uses the common field-space test metric after restoring each selected checkpoint. The architectures and training settings are fixed before their formal runs, and held-out test results are not used for configuration or checkpoint selection.

Sparse-input task adaptations. The FieldFormer, S-NOT, and FLRONet comparisons use architecture-inspired local controls for sparse-observation reconstruction. These use sensor-token self-attention with query cross-attention for FieldFormer (Bhardwaj et al., 2026), recurrent encoding of pooled sensor features for S-NOT (Liu et al., 2026), and an observation branch with a coordinate trunk for FLRONet (Dang & Nguyen, 2025). The FlowBench Senseiver (Santos et al., 2023) uses learned latent slots, sensor-to-latent cross-attention, latent self-attention, and query-to-latent decoding; the cylinder comparison instead uses the released implementation described above. All four use S16/M4 observations and the common data splits and field-space evaluation defined in Appendix A. For variable geometries, these controls receive the shared local geometry features and global geometry code through their input and query interfaces.

Raster and recurrent controls. VoronoiCNN (Fukami et al., 2021) fills a raster from the nearest current-frame sensor, appends sensor distance and coordinate channels, and predicts a field with a convolutional network. ConvLSTM (Shi et al., 2015) instead scatters each snapshot to a value raster with an occupancy channel, updates a recurrent grid state, and uses a convolutional readout. Both append geometry grids where configured and bilinearly sample their predicted rasters at query coordinates. They are protocol-matched local implementations. STRIDE (Tong & Chen, 2026) is discussed as related work but is not evaluated.

D Additional Dual-Path Evidence

D.1 Architectural and temporal controls

The supplementary architectural and query-history controls below use the 220-field cylinder evaluation and three seeds (123/456/789). Base continuation compares further state optimization with query-path training from the same seed-matched, validation-selected Base after the 30-epoch first stage. It updates all Base parameters for 15 epochs with a fresh AdamW optimizer at the Base learning rate, selecting the checkpoint by validation error within this stage. This control shares its architecture with the recurrent-assimilation model in Table 22, which is trained continuously for 45 epochs. The query-MLP control adds 218,595 parameters, six fewer than the query path, and receives only the query coordinate and Base prediction. Pooled history receives the same M4 observations, averages sensor features, and uses a GRU to form one global summary; it has 1.18M parameters, 0.10M fewer than the full query path. It changes both the encoder and readout, whereas the matched shared-summary experiment in Appendix D.3 retains Q's token-encoding architecture. Joint-window + query (Window+Q in Table 11) uses the same correction module and 30+15 staged schedule on the joint-window encoding backbone.

Table 9: Matched 128_{inc} controls under a shared S16/M4 split, query budget, and seeds. Δ is control minus dual path after held-out-Re aggregation. Query MLP is parameter-matched; †pooled history has 1.18M parameters versus 1.28M for the full model.

Control	Joint $\mathrm{nrel}L_2$	Params.	Δ [95% interval]	Dual path wins
Base continuation	0.0231 ± 0.0032	1.06M	0.0029 [0.0015, 0.0038]	10/11
Base + query MLP (no history)	0.0220 ± 0.0021	1.28M	0.0018 [0.0011, 0.0027]	11/11
Base + pooled M4 tokens†	0.0218 ± 0.0021	1.18M	0.0016 [0.0013, 0.0018]	11/11
Joint-window + query	0.0238 ± 0.0019	1.25M	0.0036 [0.0016, 0.0070]	11/11
Dual path, M1 query input	0.0204 ± 0.0023	1.28M	0.0002 [-0.0001, 0.0004]	8/11
Dual path, wrong-lag query input	0.0204 ± 0.0025	1.28M	0.0002 [0.0001, 0.0003]	10/11
Dual path, repeated-current input	0.0203 ± 0.0022	1.28M	0.0001 [-0.0001, 0.0002]	9/11
GaussAssim	$\mathbf{0.0202 \pm 0.0026}$	1.28M	reference	–

On 128_{inc}, M1, wrong-lag, and repeated-current controls change only the direct query-path input while keeping the M4 state path unchanged. Wrong-lag keeps lag positions fixed while assigning each

past sensor-value frame to an incorrect lag, leaving the current frame untouched. Repeated-current preserves 64 tokens but copies the current 16-sensor snapshot into all four lag positions. Tables 10 and 11 report channel and held-out-Re decompositions.

Table 10: Per-channel errors for matched 128_{inc} controls. Joint error is a concatenated normalized norm.

Control	Joint	u	v	p
Base continuation	0.0231	0.0229	0.0295	0.0195
Base + query MLP (no history)	0.0220	0.0220	0.0282	0.0188
Base + pooled M4 tokens	0.0218	0.0224	0.0273	0.0183
Joint-window + query	0.0238	0.0239	0.0309	0.0201
Dual path, M1 query input	0.0204	0.0208	0.0257	0.0172
Dual path, wrong-lag query input	0.0204	0.0209	0.0257	0.0172
Dual path, repeated-current input	0.0203	0.0208	0.0256	0.0170
GAUSSASSIM	0.0202	0.0208	0.0255	0.0169

Table 11: Per-trajectory error for ACDM *Inc* controls over 20 fields and three seeds. Index i maps to $Re = 100 + 10i$.

Index	Re	Cont.	Q-MLP	Pool	Window+Q	M1	Wrong lag	Repeat	Dual
0	100	0.0253	0.0262	0.0239	0.0237	0.0226	0.0217	0.0223	**0.0217**
9	190	0.0210	0.0189	0.0189	0.0200	0.0176	0.0173	0.0173	**0.0172**
18	280	0.0189	0.0159	0.0166	0.0192	0.0151	0.0152	0.0150	**0.0148**
27	370	0.0171	0.0139	0.0151	0.0170	0.0138	0.0139	0.0135	**0.0135**
36	460	0.0170	0.0136	0.0147	0.0140	0.0134	0.0135	0.0133	**0.0131**
45	550	0.0157	0.0139	0.0146	0.0137	0.0132	0.0133	0.0131	**0.0129**
54	640	0.0154	0.0129	0.0139	0.0138	0.0127	0.0124	0.0126	**0.0123**
63	730	0.0160	0.0139	0.0147	0.0147	0.0131	0.0130	0.0132	**0.0129**
72	820	0.0206	0.0194	0.0180	0.0179	0.0158	0.0164	**0.0158**	0.0161
81	910	0.0199	0.0187	0.0176	0.0177	**0.0160**	0.0161	0.0161	0.0161
90	1000	**0.0675**	0.0749	0.0716	0.0905	0.0706	0.0716	0.0707	0.0714

D.2 TEN-SEED FACTORIAL RESULTS

The two factors are repeated assimilation (a) and query correction (b); every cell initializes from the first snapshot. The four settings use seeds 1–10 and the complete test intervals listed in Table 3. Tables below give all seed-level errors, conditional effects, average main effects, and interactions under equal-case weighting, followed by both aggregation schemes for the same predictions. The positive count n_+ counts training seeds with positive mean effects; it is not an additional independence assumption or significance test. On ACDM *Inc*, the interval for the state effect includes zero without query correction, whereas the interval for its conditional effect with query correction excludes zero. Interaction intervals are negative for NURBS and Skeleton and cross zero for ACDM *Inc* and Harmonics; conditional value does not imply super-additivity.

Table 12: **ACDM *Inc*: ten-seed factorial results.** Errors and effects weight cases equally. Positive effects indicate error reduction; positive I indicates super-additivity. Intervals are pointwise crossed seed×case 95% bootstrap intervals, without multiplicity adjustment.

Seed	E_{00}	E_{01}	E_{10}	E_{11}	$\Delta_{\mathrm{S}\mid Q}$	$\Delta_{\mathrm{Q}\mid S}$	Δ_{S}	Δ_{Q}	I
1	0.0243	0.0199	0.0243	0.0200	-0.0002	0.0043	-0.0001	0.0044	-0.0002
2	0.0236	0.0199	0.0235	0.0230	-0.0031	0.0005	-0.0015	0.0021	-0.0032
3	0.0211	0.0199	0.0221	0.0186	0.0013	0.0034	0.0001	0.0023	0.0023
4	0.0261	0.0247	0.0235	0.0197	0.0050	0.0037	0.0038	0.0026	0.0023
5	0.0258	0.0215	0.0221	0.0183	0.0032	0.0038	0.0035	0.0040	-0.0005
6	0.0265	0.0212	0.0265	0.0185	0.0026	0.0079	0.0013	0.0066	0.0026
7	0.0286	0.0255	0.0266	0.0231	0.0024	0.0035	0.0022	0.0033	0.0005
8	0.0225	0.0188	0.0213	0.0179	0.0009	0.0034	0.0010	0.0035	-0.0003
9	0.0231	0.0199	0.0239	0.0203	-0.0004	0.0036	-0.0006	0.0034	0.0004
10	0.0299	0.0247	0.0272	0.0208	0.0039	0.0064	0.0033	0.0058	0.0012

Contrast	Mean	95% interval	n_+
$\Delta_{\mathrm{S}\mid\neg Q} = E_{00} - E_{10}$	0.00105	[-0.00026, 0.00272]	8
$\Delta_{\mathrm{S}\mid Q} = E_{01} - E_{11}$	0.00156	[0.00010, 0.00452]	7
$\Delta_{\mathrm{Q}\mid\neg S} = E_{00} - E_{01}$	0.00355	[0.00045, 0.00533]	10
$\Delta_{\mathrm{Q}\mid S} = E_{10} - E_{11}$	0.00406	[0.00184, 0.00537]	10
Average state effect Δ_{S}	0.00131	[0.00007, 0.00353]	7
Average query effect Δ_{Q}	0.00381	[0.00130, 0.00523]	10
Interaction I	0.00051	[-0.00070, 0.00253]	6

Table 13: **NURBS: ten-seed factorial results.** Errors and effects weight cases equally. Positive effects indicate error reduction; positive I indicates super-additivity. Intervals are pointwise crossed seed×case 95% bootstrap intervals, without multiplicity adjustment.

Seed	E_{00}	E_{01}	E_{10}	E_{11}	$\Delta_{\mathrm{S}\mid Q}$	$\Delta_{\mathrm{Q}\mid S}$	Δ_{S}	Δ_{Q}	I
1	0.1277	0.0854	0.0956	0.0747	0.0107	0.0209	0.0214	0.0316	-0.0214
2	0.1186	0.0912	0.1007	0.0895	0.0017	0.0112	0.0098	0.0193	-0.0162
3	0.1253	0.0856	0.0950	0.0757	0.0099	0.0194	0.0201	0.0295	-0.0203
4	0.1339	0.0900	0.1032	0.0757	0.0143	0.0275	0.0225	0.0357	-0.0163
5	0.1276	0.0893	0.0964	0.0791	0.0103	0.0173	0.0208	0.0278	-0.0209
6	0.1254	0.0903	0.0937	0.0752	0.0150	0.0184	0.0234	0.0268	-0.0167
7	0.1250	0.0893	0.0994	0.0766	0.0127	0.0228	0.0191	0.0292	-0.0129
8	0.1074	0.0853	0.0963	0.0772	0.0081	0.0191	0.0096	0.0206	-0.0030
9	0.1230	0.0884	0.1070	0.0806	0.0078	0.0264	0.0119	0.0305	-0.0082
10	0.1138	0.0840	0.1021	0.0748	0.0092	0.0273	0.0105	0.0285	-0.0025

Contrast	Mean	95% interval	n_+
$\Delta_{\mathrm{S}\mid\neg Q} = E_{00} - E_{10}$	0.02382	[0.01685, 0.03115]	10
$\Delta_{\mathrm{S}\mid Q} = E_{01} - E_{11}$	0.00997	[0.00715, 0.01263]	10
$\Delta_{\mathrm{Q}\mid\neg S} = E_{00} - E_{01}$	0.03488	[0.02932, 0.04080]	10
$\Delta_{\mathrm{Q}\mid S} = E_{10} - E_{11}$	0.02104	[0.01703, 0.02504]	10
Average state effect Δ_{S}	0.01690	[0.01255, 0.02140]	10
Average query effect Δ_{Q}	0.02796	[0.02412, 0.03175]	10
Interaction I	-0.01384	[-0.02025, -0.00779]	0

Table 14: **Harmonics: ten-seed factorial results.** Errors and effects weight cases equally. Positive effects indicate error reduction; positive I indicates super-additivity. Intervals are pointwise crossed seed×case 95% bootstrap intervals, without multiplicity adjustment.

Seed	E_{00}	E_{01}	E_{10}	E_{11}	$\Delta_{\mathrm{S}\mid Q}$	$\Delta_{\mathrm{Q}\mid S}$	Δ_{S}	Δ_{Q}	I
1	0.1289	0.0949	0.1255	0.0909	0.0040	0.0346	0.0037	0.0344	0.0006
2	0.1413	0.1056	0.1101	0.0861	0.0196	0.0240	0.0253	0.0298	-0.0116
3	0.1318	0.1008	0.1156	0.0879	0.0129	0.0277	0.0146	0.0294	-0.0033
4	0.1380	0.1036	0.1142	0.0859	0.0177	0.0284	0.0208	0.0314	-0.0060
5	0.1304	0.0997	0.1278	0.0930	0.0067	0.0348	0.0046	0.0327	0.0041
6	0.1318	0.0962	0.1087	0.0842	0.0120	0.0245	0.0175	0.0300	-0.0111
7	0.1267	0.0990	0.1206	0.0908	0.0082	0.0299	0.0071	0.0288	0.0021
8	0.1313	0.0993	0.1087	0.0819	0.0174	0.0268	0.0200	0.0294	-0.0052
9	0.1451	0.1098	0.1204	0.0900	0.0198	0.0304	0.0222	0.0328	-0.0049
10	0.1311	0.0984	0.1060	0.0873	0.0110	0.0187	0.0181	0.0257	-0.0140

Contrast	Mean	95% interval	n_+
$\Delta_{\mathrm{S}\mid\neg Q} = E_{00} - E_{10}$	0.01786	[0.01052, 0.02504]	10
$\Delta_{\mathrm{S}\mid Q} = E_{01} - E_{11}$	0.01293	[0.00928, 0.01674]	10
$\Delta_{\mathrm{Q}\mid\neg S} = E_{00} - E_{01}$	0.03291	[0.02902, 0.03691]	10
$\Delta_{\mathrm{Q}\mid S} = E_{10} - E_{11}$	0.02797	[0.02370, 0.03221]	10
Average state effect Δ_{S}	0.01540	[0.01021, 0.02047]	10
Average query effect Δ_{Q}	0.03044	[0.02737, 0.03358]	10
Interaction I	-0.00493	[-0.01035, 0.00034]	3

Table 15: **Skeleton: ten-seed factorial results.** Errors and effects weight cases equally. Positive effects indicate error reduction; positive I indicates super-additivity. Intervals are pointwise crossed seed×case 95% bootstrap intervals, without multiplicity adjustment.

Seed	E_{00}	E_{01}	E_{10}	E_{11}	$\Delta_{\mathrm{S}\mid Q}$	$\Delta_{\mathrm{Q}\mid S}$	Δ_{S}	Δ_{Q}	I
1	0.1392	0.1037	0.1306	0.0995	0.0042	0.0310	0.0064	0.0332	-0.0044
2	0.1483	0.1040	0.1115	0.0862	0.0178	0.0253	0.0273	0.0348	-0.0190
3	0.1504	0.1051	0.1209	0.0929	0.0123	0.0281	0.0209	0.0367	-0.0172
4	0.1477	0.1140	0.1217	0.0890	0.0251	0.0328	0.0255	0.0332	-0.0009
5	0.1554	0.1050	0.1085	0.0874	0.0176	0.0211	0.0323	0.0357	-0.0294
6	0.1499	0.1006	0.1217	0.0888	0.0118	0.0329	0.0200	0.0411	-0.0164
7	0.1429	0.1110	0.1210	0.0945	0.0165	0.0265	0.0192	0.0291	-0.0053
8	0.1534	0.1052	0.1193	0.0857	0.0195	0.0336	0.0268	0.0409	-0.0146
9	0.1492	0.1054	0.1092	0.0831	0.0223	0.0261	0.0312	0.0350	-0.0177
10	0.1433	0.1090	0.1122	0.0877	0.0213	0.0245	0.0262	0.0293	-0.0098

Contrast	Mean	95% interval	n_+
$\Delta_{\mathrm{S}\mid\neg Q} = E_{00} - E_{10}$	0.03031	[0.02121, 0.03999]	10
$\Delta_{\mathrm{S}\mid Q} = E_{01} - E_{11}$	0.01684	[0.01212, 0.02114]	10
$\Delta_{\mathrm{Q}\mid\neg S} = E_{00} - E_{01}$	0.04164	[0.03466, 0.04916]	10
$\Delta_{\mathrm{Q}\mid S} = E_{10} - E_{11}$	0.02817	[0.02360, 0.03339]	10
Average state effect Δ_{S}	0.02358	[0.01732, 0.02959]	10
Average query effect Δ_{Q}	0.03491	[0.03043, 0.03963]	10
Interaction I	-0.01347	[-0.02200, -0.00554]	0

Table 16: **Aggregation in the ten-seed study.** Both columns summarize the same predictions, using either equal frame or equal case weights. SD is computed across seed means. External comparisons and temporal controls are separate three-seed experiments.

Setting	Cell (a, b)	Frame-equal mean $\pm$ SD	Case-equal mean $\pm$ SD
ACDM *Inc*	(0,0)	0.0251 ± 0.0028	0.0251 ± 0.0028
	(0,1)	0.0216 ± 0.0025	0.0216 ± 0.0025
	(1,0)	0.0241 ± 0.0021	0.0241 ± 0.0021
	(1,1)	0.0200 ± 0.0018	0.0200 ± 0.0018
NURBS	(0,0)	0.1242 ± 0.0060	0.1228 ± 0.0076
	(0,1)	0.0892 ± 0.0020	0.0879 ± 0.0025
	(1,0)	0.1026 ± 0.0047	0.0989 ± 0.0043
	(1,1)	0.0795 ± 0.0046	0.0779 ± 0.0045
Harmonics	(0,0)	0.1374 ± 0.0055	0.1336 ± 0.0059
	(0,1)	0.1021 ± 0.0048	0.1007 ± 0.0045
	(1,0)	0.1185 ± 0.0074	0.1158 ± 0.0076
	(1,1)	0.0884 ± 0.0033	0.0878 ± 0.0034
Skeleton	(0,0)	0.1472 ± 0.0041	0.1480 ± 0.0050
	(0,1)	0.1066 ± 0.0038	0.1063 ± 0.0039
	(1,0)	0.1188 ± 0.0065	0.1177 ± 0.0071
	(1,1)	0.0895 ± 0.0041	0.0895 ± 0.0049

D.3 Matched readout attribution

Question and common prediction. This experiment tests whether the query coordinate should influence observation aggregation as well as final decoding. Each model reconstructs the normalized velocity–pressure field $q_t = (u_t, v_t, p_t)$ from $S = 16$ near-body probes over $M = 4$ snapshots, including the target time. The NURBS setting uses fixed geometry-specific probes and time-held-out intervals of screened periodic FlowBench trajectories; ACDM *Inc* uses fixed cylinder probes and 11 held-out Reynolds trajectories. For each training seed, a 64-primitive recurrent Gaussian model assimilates the window and supplies the same frozen state and uncorrected prediction $b(x)$ to all readout alternatives. B0 denotes this prediction alone, before any correction. Its parameters and buffers remain fixed, and the Base stays in evaluation mode while each readout is trained.

Three readout alternatives. Let $H \in \mathbb{R}^{64\times 96}$ be the encoded sensor–lag tokens, formed from normalized measurements, sensor coordinates, lag embeddings, and available geometry information using one self-attention layer. Write $r_x = P_q(\gamma(x), g(x), \zeta)$ for the width-96 query feature, where γ is the six-band Fourier coordinate encoding, $g(x)$ contains local geometry features, and ζ contains global geometry or other configured conditioning. The fixed-cylinder model omits geometry features. With attention A, layer normalization L, and residual decoder C, the observation-based readouts are

$$\hat{q}^{\mathrm{Q}}(x) = b(x) + C_{\mathrm{Q}}(L_{\mathrm{Q}}(r_x + A_{\mathrm{Q}}(r_x, H, H)), b(x)), \tag{21}$$

$$\hat{q}^{\mathrm{S}}(x) = b(x) + C_{\mathrm{S}}(L_{\mathrm{S}}(r_x + A_{\mathrm{S}}(a, H, H)), b(x)). \tag{22}$$

Here $a \in \mathbb{R}^{96}$ is one learned query vector shared by all requested coordinates. S therefore forms a single observation summary per window, then combines it with each coordinate's r_x and Base prediction. Both Q and S can represent spatially varying corrections; only Q's aggregation weights depend on the requested coordinate. Their token encoders, attention projections, query projections, normalization, and residual decoders have identical architectures and corresponding initial values, but are trained separately. S adds only the 96 parameters in a. Control N instead uses

$$\hat{q}^{\mathrm{N}}(x) = b(x) + C_{\mathrm{N}}(\gamma(x), g(x), \zeta, b(x)), \tag{23}$$

using a five-layer MLP with four hidden layers, width 245 on NURBS and 252 on ACDM *Inc*. It accesses observations only through $b(x)$, not through tokens or a separate summary. All three heads use an additive, zero-initialized output, so training starts from B0.

Capacity and optimization. Effective trainable head sizes for N/S/Q are 205,558/206,019/205,923 on NURBS and 199,587/199,491/199,395 on ACDM *Inc*; unused output branches are excluded. The

common frozen Bases contain 1,362,307 and 1,062,035 parameters, respectively. Seeds 1, 2, and 3 each start from the corresponding validation-selected 30-epoch Base and freshly initialize all heads, including Q. Each head trains for 15 epochs with a fresh AdamW optimizer, learning rate 5×10^{-4}, weight decay 10^{-4}, gradient clipping at 1.0, and 4,096 fluid queries per training example. Batch sizes are 4 on NURBS and 16 on ACDM *Inc*. Training batches and query samples are paired by seed and epoch across heads. The loss is MSE over training-normalized UVP; the cylinder primitive regularizer is constant with respect to the readout parameters. The checkpoint minimizing validation error is selected, retaining the earliest in a tie. Validation uses two fields per NURBS trajectory and 32 per cylinder trajectory; test evaluation follows completion of all 18 readout trainings.

Evaluation and inference. All alternatives score the same 1,816 NURBS fields (84 geometries) or 6,039 cylinder fields (11 Reynolds cases) per seed. For each field, error is the joint normalized relative norm $\|\operatorname{vec}(\tilde{\hat{q}} - \tilde{q})\|_2 / \max\{\|\operatorname{vec}(\tilde{q})\|_2, 10^{-8}\}$ over valid fluid cells and all three channels, using training-only normalization. Let e_{hsg} be the mean field error for head h, seed s, and case g. The reported error is $(3G)^{-1} \sum_{s,g} e_{hsg}$, with sample SD across the three case-equal seed means. For $h \in \{\mathrm{S}, \mathrm{N}, \mathrm{B0}\}$, the paired gain is $\Delta_{h-\mathrm{Q}} = (3G)^{-1} \sum_{s,g} (e_{hsg} - e_{\mathrm{Q}sg})$. This three-seed comparison uses the 30-epoch B0 and separately trained N/S/Q heads; the external comparisons, temporal controls, and ten-seed factorial each use their respective training and evaluation protocols.

For each of 100,000 crossed bootstrap draws, three seed indices and G case indices are independently sampled with replacement and shared across all heads. The two prespecified primary comparisons are $\Delta_{\mathrm{S}-\mathrm{Q}}$ on NURBS and ACDM *Inc*. Their 1.25th–98.75th percentile intervals apply the Bonferroni adjustment for these two tests (97.5% marginal intervals for a nominal 95% family-level criterion). Secondary $\mathrm{N}-\mathrm{Q}$ and $\mathrm{B0}-\mathrm{Q}$ contrasts use unadjusted 2.5th–97.5th percentile intervals. The bootstrap uses batches of 2,000 and RNG seeds 2026091902/2026091901 for NURBS/ACDM. Positive-case counts and case medians use $\bar{\Delta}_g = 3^{-1} \sum_s (e_{hsg} - e_{\mathrm{Q}sg})$; they describe coverage across cases, not independent training repetitions or per-case significance.

Results. On NURBS, S and Q have mean errors 0.08479 ± 0.00829 and 0.07940 ± 0.00621; on ACDM *Inc*, they give 0.02210 ± 0.00220 and 0.02045 ± 0.00226 (sample SD over three seeds). Q improves on S in all three seeds of each setting, reducing case-equal mean error by 6.36% on NURBS and 7.48% on ACDM *Inc*. The adjusted $\mathrm{S} - \mathrm{Q}$ intervals are $[0.00297, 0.00803]$ and $[0.00105, 0.00228]$, respectively. Tables 17–18 give all seed results and paired contrasts; Figure 4 includes every case, including the two NURBS cases with negative mean gains. These results favor query-conditioned aggregation over the tested shared-summary readout while preserving both alternatives' coordinate-dependent output capability.

Post-hoc sensitivity to training-seed uncertainty. As a sensitivity analysis, we hold the evaluated cases fixed and form one paired mean gain $d_s = G^{-1} \sum_g (e_{\mathrm{S}sg} - e_{\mathrm{Q}sg})$ per training seed. The three NURBS gains are 0.003203, 0.007701, and 0.005278; the ACDM *Inc* gains are 0.001992, 0.001458, and 0.001510. Assuming normally distributed seed-level gains, a two-sided 97.5% Student-t interval is $\bar{d} \pm t_{0.9875,2} s_d / \sqrt{3}$, where s_d is their sample standard deviation. It gives $[-0.00267, 0.01346]$ on NURBS and $[0.00060, 0.00271]$ on ACDM *Inc*. This assesses retraining uncertainty for the observed cases, whereas the prespecified crossed bootstrap also resamples cases; the normality assumption is difficult to assess with only three seeds. All observed seed gains remain positive in both settings, while the NURBS inference about mean gain across retraining is sensitive to the interval construction. The prespecified crossed-bootstrap comparisons remain the primary analysis.

Correction of the same state prediction across four settings. A separate evaluation of the ten-seed dual-path checkpoints compares each output with its own uncorrected Gaussian prediction, keeping the state, inputs, and weights fixed. Table 19 uses complete test intervals, equal-case weighting, and the same crossed bootstrap, with ten resampled seeds and unadjusted 95% intervals. RNG seeds are 2026091950–2026091953 in ACDM/NURBS/Harmonics/Skeleton order. Every training seed improves in each setting. This measures correction of an existing prediction; the separately trained factorial cells measure the value of retaining either interface under the staged training scheme.

Table 17: **Readout errors for every training seed.** Frames are averaged within each case, then cases are weighted equally. Positive $S - Q$ favors query-conditioned aggregation.

Setting	Seed	B0	N	S	Q	$S - Q$
NURBS	1	0.11708	0.10617	0.07883	0.07563	0.00320
NURBS	2	0.10497	0.09328	0.09426	0.08656	0.00770
NURBS	3	0.11222	0.10324	0.08128	0.07600	0.00528
ACDM *Inc*	1	0.03315	0.02209	0.02196	0.01997	0.00199
ACDM *Inc*	2	0.03073	0.02441	0.02437	0.02291	0.00146
ACDM *Inc*	3	0.02484	0.01937	0.01998	0.01847	0.00151

Table 18: **Paired readout contrasts and case coverage.** All intervals here are pointwise 95%; adjusted intervals for the two primary $S - Q$ tests appear in Table 5. Case medians and positive counts use each case's three-seed mean, not per-case significance tests. All six contrasts improve in 3/3 training seeds.

Setting	Contrast	Mean	95% interval	Case median	Cases > 0
NURBS	$S - Q$	0.00539	$[0.00325, 0.00767]$	0.00537	82/84
NURBS	$N - Q$	0.02150	$[0.00706, 0.03170]$	0.01967	84/84
NURBS	$B0 - Q$	0.03203	$[0.01898, 0.04270]$	0.02920	84/84
ACDM *Inc*	$S - Q$	0.00165	$[0.00111, 0.00216]$	0.00159	11/11
ACDM *Inc*	$N - Q$	0.00151	$[0.00048, 0.00269]$	0.00078	9/11
ACDM *Inc*	$B0 - Q$	0.00912	$[0.00613, 0.01299]$	0.00960	11/11

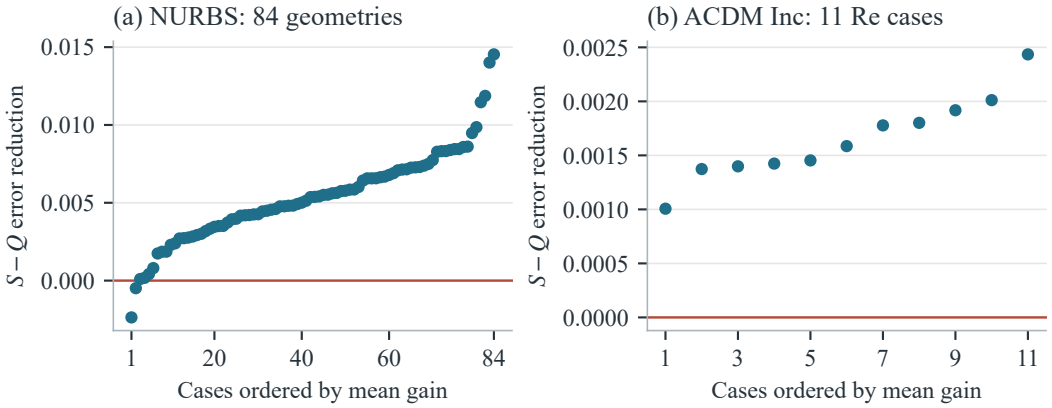

Figure 4: **Complete case-level coverage of the shared-summary comparison.** Each point is one geometry or Reynolds case, averaged across three training seeds and ordered by its $S - Q$ gain. Positive values favor Q; 82/84 NURBS geometries and 11/11 cylinder cases improve. Points are descriptive case means, not separate significance tests.

Table 19: **Correction at a fixed checkpoint.** Within each of the ten-seed full models, the Gaussian state and its uncorrected prediction are held fixed; the gain is error before minus after correction. These are separate readout evaluations of the factorial full models, not retrained state-only comparisons.

Setting	Mean gain	Pointwise 95% interval	Seeds > 0	Cases > 0
ACDM *Inc*	0.00880	$[0.00714, 0.01028]$	10/10	11/11
NURBS	0.04023	$[0.03318, 0.04685]$	10/10	84/84
Harmonics	0.04384	$[0.03984, 0.04799]$	10/10	83/83
Skeleton	0.05046	$[0.04488, 0.05666]$	10/10	53/53

D.4 PHYSICAL AND ROBUSTNESS DIAGNOSTICS

Physical diagnostics use all 6,039 eligible cylinder test fields per seed; robustness tests use a 220-field subset, with 20 fields per test trajectory. For the fixed cylinder, the near-body region contains valid cells within two equivalent cylinder radii of the body center; the wake region starts two radii downstream and extends three radii on either side of the centerline. Velocity derivatives use centered differences with the physical dx, dy recorded by PhiFlow and are evaluated only where the center and four stencil neighbors are fluid. Boundary-normal error uses first-fluid-layer cells with a four-neighbor solid contact and the radial outward normal of the circular cylinder. All regional and differential diagnostics compare predictions with ground truth after reversing channel normalization. Near-body and wake errors concatenate the physical-unit UVP values within the respective region before taking the relative norm. Writing $\mathbf{u} = (u, v)$, the two reported RMSE diagnostics are

$$E_{\mathrm{div}} = \sqrt{\langle (\nabla\cdot\hat{\mathbf{u}} - \nabla\cdot\mathbf{u})^2 \rangle_{\mathrm{interior}}}, \qquad E_{\mathrm{normal}} = \sqrt{\left\langle \left((\hat{\mathbf{u}} - \mathbf{u})\cdot n\right)^2 \right\rangle_{\mathrm{boundary}}}. \tag{24}$$

The brackets denote the mean over the indicated cells, not a residual against zero. Vorticity error is $\|\hat{\omega} - \omega\|_2 / \max(\|\omega\|_2, 10^{-12})$, with $\omega = \partial_x v - \partial_y u$ on the same interior stencil. These diagnostics are computed per frame and then averaged over test frames and training seeds.

For missing-sensor tests, one seeded sensor-index subset is removed from every M4 lag without zero imputation. Noise is independent Gaussian perturbation $\eta \sim \mathcal{N}(0, \alpha^2)$ added to each normalized sensor value, with $\alpha \in \{0.02, 0.05\}$; coordinates are unchanged. Each training seed uses a fixed validation-selected checkpoint across sensor conditions; corruption results average seeds 20260827/20260828/20260829, with identical perturbations across methods for each case and frame. Table 22 reports the mean and sample standard deviation across training seeds 123/456/789 after that corruption average; the clean condition is evaluated once per training seed.

Table 20: Practical comparison with the recurrent-assimilation control. Errors use the 128_{inc} evaluator unless noted. Detailed model-level results follow below.

Evidence and protocol	GAUSSASSIM	Recurrent assimilation
Raw RMSE $(u/v/p)$, complete interval	$(5.8, 4.6, 0.0781)\times10^{-3}$	$(5.3, 4.5, 0.0788)\times10^{-3}$
Additive noise, $0.02\sigma/0.05\sigma$	$0.0340/0.0666$	$0.0336/0.0661$
Missing sensors, $25\%/50\%$	$0.2132/0.5134$	$0.2339/0.4233$
Full-field latency, RTX 5060 Ti	18.72 ms, 40 MiB	16.32 ms, 45 MiB

Table 21: Physical-unit diagnostics on the complete ACDM *Inc* interval, averaged over three training seeds. Near-body and wake norms use denormalized UVP. Divergence and boundary-normal RMSE compare prediction with GT, not with zero; derivatives use physical grid spacing and valid fluid stencils.

(a) Field and regional errors

Gaussian model	RMSE-u	RMSE-v	RMSE-p	Near-body rel.	Wake rel.
Current-frame encoding	0.0058	0.0051	8.37e-05	**0.0082**	0.0128
Joint-window encoding	0.0085	0.0070	1.10e-04	0.0102	0.0161
Recurrent assimilation	**0.0053**	**0.0045**	7.88e-05	0.0101	0.0133
GAUSSASSIM	0.0058	0.0046	**7.81e-05**	0.0095	**0.0119**

(b) Differential and boundary errors

Gaussian model	Div. RMSE	Vort. rel.	Boundary u_n RMSE
Current-frame encoding	**0.0283**	**0.0360**	**0.0029**
Joint-window encoding	0.0372	0.0449	0.0040
Recurrent assimilation	0.0397	0.0429	0.0044
GAUSSASSIM	0.0465	0.0476	0.0046

Table 22: Zero-shot robustness on the 220-field 128_{inc} subset. Gaussian noise uses normalized units; sensor removal is shared across M4 lags. Each training seed first averages three corruption seeds, then values report mean $\pm$ sample standard deviation across three training seeds (clean uses one evaluation per seed).

Sensor condition	Current-frame encoding	Joint-window encoding	Recurrent assimilation	GAUSSASSIM
Clean	0.0210 ± 0.0041	0.0268 ± 0.0052	0.0220 ± 0.0015	0.0202 ± 0.0026
Noise 0.02σ	0.0397 ± 0.0076	0.0422 ± 0.0117	0.0336 ± 0.0041	0.0340 ± 0.0060
Noise 0.05σ	0.0809 ± 0.0146	0.0804 ± 0.0235	0.0661 ± 0.0124	0.0666 ± 0.0127
25% missing	0.2885 ± 0.1544	0.2213 ± 0.0460	0.2339 ± 0.1416	0.2132 ± 0.0444
50% missing	1.3761 ± 1.6238	0.5143 ± 0.1112	0.4233 ± 0.0563	0.5134 ± 0.1505

Table 23: Efficiency on one NVIDIA GeForce RTX 5060 Ti. Inference latency and peak memory use batch size 1 and full-field decoding, excluding data loading. Steps and sampled queries cover 45 epochs: continuous state training for recurrent assimilation, or 30 Base epochs followed by 15 frozen-Base correction epochs for GAUSSASSIM. Train h is estimated from the modification-time difference between the run configuration and final checkpoint; PySHRED uses direct timing from its training log. †Architecture-inspired local controls (Appendix C).

Method	Params.	Latency (ms)	Peak MiB	Steps	Query samples	Train h
Compared baselines						
PySHRED (Ye et al., 2025)	0.46M	0.92	57	108.1k	–	0.17
Senseiver (Santos et al., 2023)	1.34M	15.05	113	108.1k	7.08B	2.72
S-NOT† (Liu et al., 2026)	0.32M	2.25	31	108.1k	7.08B	1.23
FieldFormer† (Bhardwaj et al., 2026)	0.35M	3.58	36	108.1k	7.08B	1.69
Gaussian controls (this work)						
Current-frame encoding	0.92M	3.66	42	108.1k	7.08B	1.69
Joint-window encoding	1.03M	3.89	44	108.1k	7.08B	1.39
Recurrent assimilation	1.06M	16.32	45	108.1k	7.08B	2.63
GAUSSASSIM	1.28M	18.72	40	108.1k	7.08B	2.84

On the complete held-out interval, GAUSSASSIM has the lowest joint normalized error and wake-region error, while recurrent assimilation gives the lowest raw u/v RMSE and current-frame encoding gives the lowest divergence, vorticity, and boundary-normal errors. The complete model gives the lowest pressure RMSE. These rankings distinguish field-value accuracy from agreement with reference derivatives and near-boundary velocity; none of these models is trained with divergence, vorticity, or boundary residuals. Under normalized additive noise, GAUSSASSIM changes from 0.0202 to 0.0340 at 0.02σ and 0.0666 at 0.05σ. Table 22 characterizes zero-shot robustness under progressive sensor removal. Full-field decoding takes 18.7 ms at batch size one on the reported RTX 5060 Ti, compared with 16.3 ms for the state-only Gaussian model.

D.5 RUNTIME ROLE DIAGNOSTIC

We evaluate inference-time path switches on the trained dual-path checkpoints for seeds 123/456/789, without retraining. Each seed uses five evenly spaced eligible test windows per held-out Reynolds trajectory on 128_{inc} and the first eligible test window per NURBS geometry: $3 \times 11 \times 5 = 165$ and $3 \times 84 = 252$ window evaluations, respectively. Predictions are paired at the same fluid queries within each window. With query correction disabled, repeated assimilation is compared with initialization-only inference after each sampled observation:

$$B^{(m)}_{\text{state}} = \frac{1}{|\mathcal{R}|} \sum_{r \in \mathcal{R}} \left(e^{(m)}_{r,\text{init}} - e^{(m)}_{r,\text{repeat}} \right). \tag{25}$$

Here m indexes M4 observations, $\mathcal{R}$ contains the Re cases or geometries, and $e^{(m)}_{r,j}$ averages joint normalized relative error over sampled windows and seeds for case r under switch j.

The error increase from disabling repeated assimilation grows across M4, reaching 0.542 on 128_{inc} and 0.743 on NURBS.

For spatial comparisons at the target time, define $d_j(x) = \|\tilde{\hat{q}}_j(x) - \tilde{q}(x)\|_2$ using training-normalized channels, with $j \in \{\text{init}, \text{repeat}, \text{full}\}$. The first two settings disable query correction; the full setting enables both paths. Pointwise state and query benefits are $d_{\text{init}} - d_{\text{repeat}}$ and $d_{\text{repeat}} - d_{\text{full}}$, respectively. Correction magnitude is $\|\tilde{\hat{q}}_{\text{full}} - \tilde{\hat{q}}_{\text{repeat}}\|_2$, and Base error is d_{repeat}.

Pearson correlations and top-20% benefit-set Jaccard overlaps are computed within each window and averaged over windows and seeds.

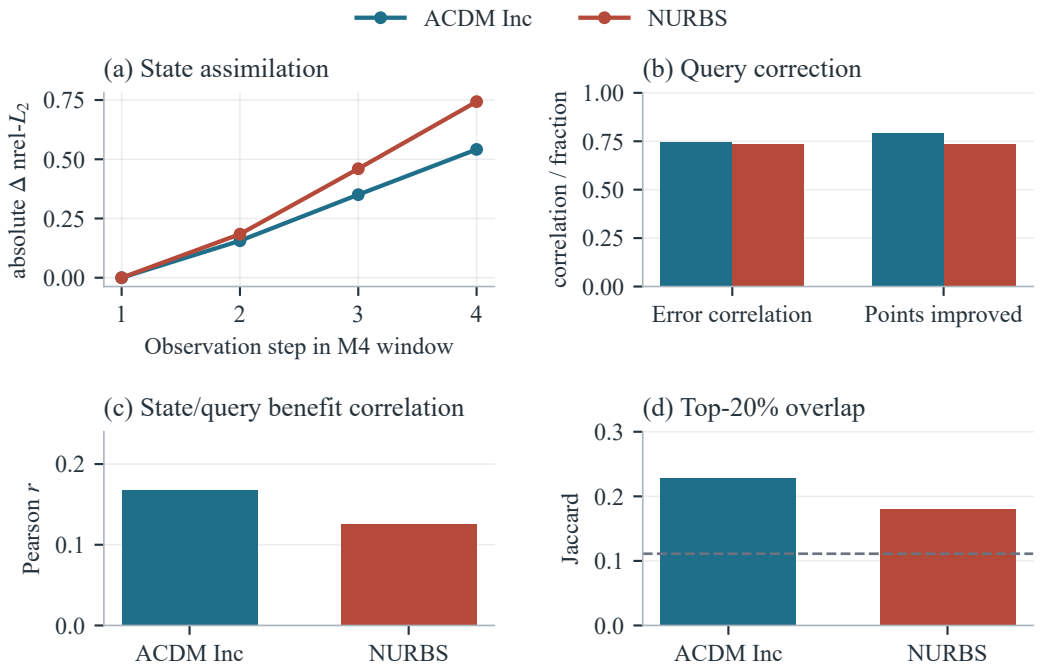

Figure 5: **Roles of the two observation paths.** (a) Disabling repeated assimilation increases error more at later M4 steps. (b) Larger corrections coincide with higher Base error, and query correction improves most evaluated points. (c) Pointwise state/query benefits are weakly correlated. (d) Their top-20% benefit subsets overlap partially; the dashed line marks the 0.111 independent-set Jaccard reference.

These switches characterize the trained model's behavior; Table 3 evaluates separately trained configurations. Within each window, the Base high-error quartile receives 0.0282 more mean query benefit than the remaining points on 128_{inc} and 0.0940 more on NURBS, after averaging over windows and seeds. Pointwise 95% intervals exclude zero in both settings; they use 100,000 bootstrap resamples of Re/geometry means after averaging the selected windows and three seeds within each case.

E Extended Evaluation Settings

E.1 Pressure-only cylinder reconstruction

A pressure-only variant reconstructs the 128×64 field from eight surface-adjacent probes over four consecutive, target-inclusive snapshots. It retains the cylinder's 70/10/11 Reynolds-stratified split and 200-frame startup removal (Appendix A). Targets use stride two; pressure normalization uses every twentieth training frame. Normalized pressure relative errors over fluid cells are averaged across 20 evenly spaced eligible fields per test trajectory (220 per seed), then summarized as mean $\pm$

sample SD across seeds 123/456/789. The 30-epoch Gaussian Base obtains 0.02070 ± 0.00212; 15 query-training epochs on the frozen Base yield 0.01433 ± 0.00346.

E.2 RESTRICTED-ALE PROPAGATION

Restricted-ALE replaces the default center transition with self-advection while retaining state assimilation and query correction. Let $\bar{Z}_{\tau-1}$ be the previous state after primitive self-attention and layer normalization, with centers and amplitudes unchanged. At each center, its primitive field gives normalized UVP, whose first two channels are inverted using the training means and standard deviations to obtain $\hat{\mathbf{u}}_k = (\hat{u}_k, \hat{v}_k)$ in the dataset's velocity units. Define the norm clip $\mathcal{C}_a(v) = v \min\{1, a/(\|v\|_2 + 10^{-8})\}$ and set $(\bar{u}_k, \bar{v}_k) = \mathcal{C}_{v_{\max}}(\hat{\mathbf{u}}_k)$. On FlowBench, centers use normalized grid-axis order for a grid of shape (n_0, n_1), so the velocity components are swapped and divided by the corresponding grid intervals:

$$\mathbf{v}_k^{\text{coord}} = \left(\frac{\bar{v}_k}{n_0 - 1}, \frac{\bar{u}_k}{n_1 - 1} \right), \tag{26}$$

$$\rho_k = \text{sigmoid}\, M_\theta(\bar{Z}_{\tau-1,k}, g(\mu_k), \zeta, \mathbf{v}_k^{\text{coord}}), \tag{27}$$

$$\widetilde{\mu}_{\tau k} = \mu_{\tau-1,k} + f_k^2(3 - 2f_k)(p_k - \mu_{\tau-1,k}). \tag{28}$$

Here $p_k = \text{clamp}_{[0,1]^2}\big(\mu_{\tau-1,k} + \mathcal{C}_{\delta_{\max}}(\eta \rho_k \mathbf{v}_k^{\text{coord}})\big)$ and $f_k = \text{clamp}_{[0,1]}\, F(p_k)$, where F is the bilinearly sampled fluid-fraction channel. The FlowBench runs use $(n_0, n_1) = (128, 512)$, $\eta = 1$, $v_{\max} = 2$ in dataset velocity units, and $\delta_{\max} = 0.01$ in normalized coordinates. The configured step multiplier η operates on grid-scaled velocity; it is not identified with the physical sampling interval. The factor $f_k^2(3 - 2f_k)$ smoothly attenuates displacement near solids rather than projecting the center onto the fluid domain. The cylinder variant instead uses normalized physical (x, y) coordinates and $\mathbf{v}_k^{\text{coord}} = (\bar{u}_k/L_x, \bar{v}_k/L_y)$ with domain lengths $(L_x, L_y) = (4, 2)$, configured step multiplier $\eta = 0.05$, and $\delta_{\max} = 0.025$. Its velocity clip is also $v_{\max} = 2$; the same attenuation samples the cylinder's fluid mask. Other primitive attributes retain the default learned transition with its center increment disabled. The self-advection step precedes current-snapshot assimilation and uses the previous primitive field, not the query-corrected output. Dense UVP fields supervise reconstruction; there is no separate ground-truth velocity loss on center motion. This learned restricted-ALE prior couples center motion to the model's own reconstructed velocity.

Table 24: Default and restricted-ALE transitions, with and without query correction. Entries are frame-equal normalized relative error, mean ± sample standard deviation over three seeds. The cylinder uses the 220-field subset; periodic families use complete test intervals.

Setting	Recurrent assimilation	GAUSSASSIM	Restricted-ALE state	Restricted-ALE + query
ACDM *Inc*	0.0220 ± 0.0015	0.0202 ± 0.0026	0.0232 ± 0.0029	0.0215 ± 0.0027
NURBS	0.1043 ± 0.0041	0.0777 ± 0.0010	0.1223 ± 0.0097	0.0789 ± 0.0057
Harmonics	0.1202 ± 0.0027	0.0929 ± 0.0050	0.1299 ± 0.0075	0.0898 ± 0.0020
Skeleton	0.1201 ± 0.0009	0.0961 ± 0.0112	0.1501 ± 0.0083	0.0948 ± 0.0024
Periodic avg.	0.1149	0.0889	0.1341	0.0878

Table 25: Geometry-level paired effect of restricted-ALE self-advection in the dual-path model. Positive differences favor restricted-ALE.

Family	Default−ALE	95% interval	Wins
NURBS	-0.0004	[-0.0036, 0.0015]	49/84
Harmonics	0.0032	[0.0020, 0.0044]	66/83
Skeleton	0.0023	[-0.0015, 0.0087]	26/53

With query correction, restricted-ALE attains a periodic three-family mean of 0.0878, compared with 0.0889 for the default transition. Its frame-equal mean improves on Harmonics and Skeleton but increases on NURBS and the cylinder; only the Harmonics paired interval excludes zero (Table 25). Observation correction improves both transitions, while the relative benefit of self-advection varies with the task.

F GEOMETRY-OOD RECONSTRUCTION

Table 26: Geometry-OOD reconstruction: normalized UVP relative error, mean ± sample standard deviation over three seeds. Each test geometry contributes ten frames; Average weights the three families equally. Best means in this comparison are bolded. †Architecture-inspired local controls (Appendix C).

Method	NURBS	Harmonics	Skeleton	Average
Compared baselines				
S-NOT† (Liu et al., 2026)	0.4816 ± 0.0317	0.6113 ± 0.0253	0.5960 ± 0.0167	0.5630
FieldFormer† (Bhardwaj et al., 2026)	0.4619 ± 0.0104	0.5927 ± 0.0341	0.5207 ± 0.0932	0.5251
Senseiver (Santos et al., 2023)	0.4674 ± 0.0167	0.5908 ± 0.0055	0.4838 ± 0.1067	0.5140
Gaussian variants (this work)				
Current-frame encoding	0.4677 ± 0.0427	0.6441 ± 0.0306	0.5561 ± 0.0341	0.5560
Joint-window encoding	0.4620 ± 0.0387	0.5867 ± 0.0745	0.5047 ± 0.0766	0.5178
Recurrent assimilation	0.4541 ± 0.0608	0.5453 ± 0.0095	**0.4699 ± 0.0414**	0.4898
Restricted-ALE state	0.4244 ± 0.0202	0.5979 ± 0.0426	0.5105 ± 0.0738	0.5109
Restricted-ALE + query	**0.4239 ± 0.0314**	0.5952 ± 0.0326	0.5081 ± 0.0716	0.5090
GAUSSASSIM	0.4280 ± 0.0451	**0.5450 ± 0.0103**	0.4728 ± 0.0379	**0.4819**

Task and partitions. This stress test reconstructs fields on held-out geometries using fixed family-specific geometry partitions established after exploratory experiments. Each geometry contributes one Reynolds-number trajectory; the available inputs are S16/M4 near-body observations, geometry features, and normalized $\log Re$ (current-frame encoding uses only the current snapshot). NURBS has 50 initial training, 16 validation, and 17 test geometries; final fitting combines the first two groups into 66 training geometries. Harmonics uses 62 development and 15 test geometries; three development-only folds select training duration before refitting on all 62. Their training/validation/development-check sizes are 40/11/11, 42/10/10, and 42/10/10 geometries, respectively; none includes the 15 final test geometries. Skeleton uses 45 training geometries and five test geometries: the first 85% of eligible frames on the 45 geometries train the models, and the remaining 15% form the validation set on those same geometries. All methods use seeds 123, 456, and 789 and the same final test membership within each family.

Training and selection. For NURBS, each method's refit duration is the median validation-best epoch from its three initial runs; Harmonics uses the median validation-best epoch across its three development folds. In the order current-frame encoding, joint-window encoding, Senseiver, FieldFormer, S-NOT, and recurrent assimilation, the NURBS durations are 15/15/11/8/15/8 epochs and the Harmonics durations are 8/8/10/19/5/19 epochs. NURBS and Harmonics evaluate the final fixed-epoch checkpoint; their retained monitoring subsets are part of refit training and do not select checkpoints. Skeleton trains for up to 30 epochs, with patience five, and selects the checkpoint with the lowest geometry-averaged validation relative error. The query branch then trains for 15 additional epochs on the frozen, selected Gaussian Base and uses its final checkpoint; restricted-ALE variants retain the corresponding family schedule and use a final 15-epoch query stage when present. The Harmonics Gaussian configuration additionally modulates sensor embeddings with the global geometry/condition code before sensor self-attention; this option is not used in the primary experiments.

Scoring and outcome. Targets lie in stored frames $[60, 240)$, with candidate stride two and observation-history stride four. Ten evenly spaced eligible target frames per test geometry are evaluated on the full fluid grid, giving 170/150/50 fields per seed for NURBS/Harmonics/Skeleton. For each frame, joint normalized UVP relative error is $\|\hat{q} - q\|_2 / \max(\|q\|_2, 10^{-8})$ over all fluid points and channels; fields use training-derived channel normalization. Frame errors are averaged within each seed, then reported as mean ± sample standard deviation across the three seeds; equal frame counts also give equal geometry weighting. The final column averages the three family means equally. The dual-path model leads this average, while family-specific leaders differ on NURBS and Skeleton.