CHROMATIC GIBBS SAMPLING FOR LONG-RANGE DATA ASSIMILATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Estimating the state of an evolving system from intermittent, noisy observations is a central scientific problem known as data assimilation (DA). For long-range DA, such as atmosphere or ocean state estimation, training generative models directly can be prohibitively expensive. The standard alternative is to compose models with short time horizons at inference. Existing composition methods, however, fail to sample from the target assimilation distribution, either due to approximations or intrinsic flaws. Drawing from the Gibbs sampling and graphical model literature, we introduce GiBBS, an iterative method for composing Markov-local generative models that provably converges to the target distribution under reasonable assumptions. We validate and ablate our method on the classical Ornstein-Uhlenbeck and Lorenz-63 systems, before scaling it to high-dimensional Kuramoto-Sivashinsky and turbulent Navier-Stokes systems. Across these experiments, GiBBS consistently and substantially improves upon its initial guess in few (1-16) cycles, highlighting its potential for operational assimilation.

1 INTRODUCTION

Reconstructing the state of a complex dynamical system from partial and noisy observations, a problem known as data assimilation (Lorenc, 1986; Dimet & Talagrand, 1986; Carrassi et al., 2018; Rozet & Louppe, 2023; Huang et al., 2024; Savary et al., 2026), is central to weather forecasting and reanalysis, ocean state estimation, epidemiology and many other scientific and engineering applications (Andry et al., 2025; Ghil & Malanotte-Rizzoli, 1991; Bettencourt et al., 2007; Nadler et al., 2020; Pandya et al., 2022). Rolling out an emulator forward in time is sufficient for forecasting, but assimilation asks for more: observations are scattered along the whole trajectory, and every state must be consistent with those that come after it as much as with those that come before. The task is therefore to sample the joint posterior $p(x_{1:L} \mid y)$ over an entire trajectory $x_{1:L}$ of length L, where y denotes partial and noisy observations of the trajectory. Training a generative model over such horizons is prohibitively expensive, so practitioners instead rely on models with short temporal receptive fields, emulating windows of length $W \ll L$, which are composed at inference to cover the longer horizon L. Composing such local models to sample from $p(x_{1:L} \mid y)$ is the central challenge we address.

Autoregressive rollout (AR) The first and most natural approach (Figure 1A), applicable to any conditional generative model, is to generate the trajectory one window at a time, in an autoregressive rollout. The posterior factorizes exactly as

$$p(x_{1:L} \mid y) = \prod_{i=1}^{L} p(x_i \mid x_{1:i-1}, y), \tag{1}$$

and a local model with receptive field W approximates each factor by truncating the context to the C previous states and the observations of the current window, $p(x_i \mid x_{1:i-1}, y) \approx p(x_i \mid x_{i-C:i-1}, y_i)$. Shysheya et al. (2024) exploit this and roll out conditional diffusion processes, generating $W - C$ states per step conditioned on the C previous ones. Each window is thus conditioned on an already-generated, self-consistent past, but the scheme is sequential in L and every window is committed before the next is drawn, so errors accumulate along the rollout. Observations are incorporated at the level of each window, either through architectural conditioning or through zero-shot posterior

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

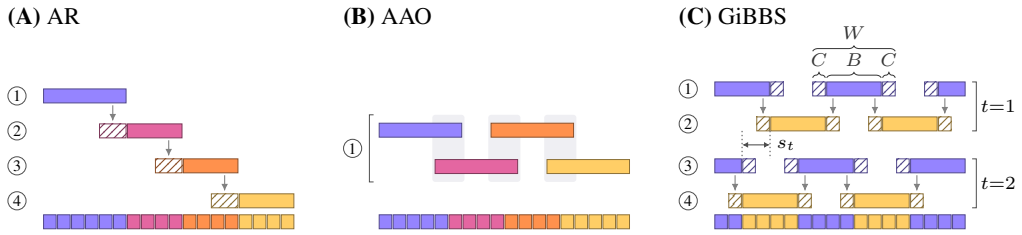


Figure 1. Composition strategies for local diffusion models. Colors indicate which window contributes to each portion of the long trajectory. Pink segments highlight conditioning. **(A)** Autoregressive (AR) sequential rollout. **(B)** All-at-once (AAO): all windows are sampled in parallel. **(C)** Two cycles of Gibbs Block Blanket Sampling (GiBBS), highlighting the shift s_t.

sampling (Shysheya et al., 2024; Chung et al., 2023; Rozet & Louppe, 2023; Rozet et al., 2024; Song et al., 2023). This leaves the sampler blind to future observations by construction, since neither mechanism can constrain states that have not been drawn yet, and the resulting trajectories do not follow the correct posterior.

All-at-once sampling (AAO) The second approach (Figure 1B), introduced by Rozet & Louppe (2023), removes this sequentiality by generating arbitrarily long trajectories in a single pass. Unlike AR, it is specific to score-based generative models, whose reverse diffusion is driven by the score of the noise-perturbed distribution at each noise level. Composing local models then amounts to assembling the score of the whole trajectory from scores of short windows which is made possible by the Markovian structure of dynamical systems. Writing x_{-i} for all states except x_i, a *Markov blanket* of x_i is a subset $\mathcal{M}(x_i)$ such that

$$p\big(x_i \mid x_{-i}\big) = p\big(x_i \mid \mathcal{M}(x_i)\big)\,. \tag{2}$$

In other words, a variable is independent of all others given its Markov blanket. For a Markovian system of order k, the smallest blanket is the k states on either side, $\mathcal{M}(x_i) = x_{i-k:i+k} \setminus \{x_i\}$. When the likelihood factorizes over frames, $p(y \mid x_{1:L}) = \prod_i p(y_i \mid x_i)$, the same locality holds for the posterior, $p(x_i \mid x_{-i}, y) = p(x_i \mid \mathcal{M}(x_i), y_i)$, where y_i denotes the observations of x_i. The score of the trajectory then decomposes element-wise as $\nabla_{x_i} \log p(x_{1:L} \mid y) = \nabla_{x_i} \log p(x_i \mid \mathcal{M}(x_i), y_i)$, so that each element depends only on the local window $x_{i-k:i+k}$ and can be evaluated independently and *in parallel*. This decomposition is exact for the clean trajectory and remains accurate at low diffusion noise levels (Rozet & Louppe, 2023), but becomes a rough approximation at high noise levels, requiring Langevin-based correction steps and limiting the speed at which information from the observations is propagated along the time axis. The method therefore struggles in forecasting or temporally-sparse observation settings (Shysheya et al., 2024; Rozet, 2026).

2 Gibbs Block Blanket Sampling (GiBBS)

Gibbs sampling (Geman & Geman, 1984; Gelfand & Smith, 1990) is a Markov chain Monte Carlo method for sampling from joint distributions that are intractable but whose conditionals are not. It redraws the variables in *cycles* from their conditional given all the others, and after enough cycles, the chain is distributed according to the joint. For a dynamical system, Markov's properties make each of these updates local

$$x_i \sim p(x_i \mid x_{-i}, y) = p(x_i \mid \mathcal{M}(x_i), y_i), \qquad i = 1, \dots, L. \tag{3}$$

We rely on this property to propose GiBBS (Algorithm 1), a recomposition scheme that, unlike previous methods, samples from the target assimilation posterior $p(x_{1:L} \mid y)$, while remaining as parallelizable as all-at-once sampling.

Block & Blanket Proceeding one frame x_i at a time converges to the posterior $p(x_{1:L} \mid y)$ (Geman & Geman, 1984), but requires L model evaluations per cycle and leaves most of the receptive field of the local model unused. Gibbs sampling, however, remains valid when several variables are redrawn

jointly, and larger groups typically mix faster. We therefore work with blocks rather than single frames, splitting $x_{1:L}$ into N disjoint blocks $b_{1:N}$ of size B, updated as

$$b_n \sim p(b_n \mid \mathcal{M}(b_n), y_{b_n}), \qquad \mathcal{M}(b_n) = \{b_{n-1}[-k:]\,;\, b_{n+1}[:k]\} \qquad n = 1, \dots, N, \tag{4}$$

so that each block is conditioned only on the k boundary states of its two neighbors and on the observations y_{b_n} falling inside it. This is the minimal Markov blanket, but any wider strip of $C \geq k$ states on either side remains a valid blanket as well and leaves the conditional unchanged. Each block being drawn from its exact conditional, the blocked Gibbs sampler retains the convergence (Gelfand & Smith, 1990; Gonzalez et al., 2011). In practice, we choose $C \leq B$ such that a block and its blanket fill the receptive field of the local model, $B + 2C = W$ (Figure 1). Nothing beyond sampling a block conditioned on its blanket is required, so the scheme applies to any conditional generative model. In this work, we instantiate it with diffusion models, i.e. each block update is a full reverse diffusion from pure noise, with the blanket held fixed and observations enforced by architectural conditioning or zero-shot posterior sampling, as in AR and AAO.

Parallelism As we impose $B \geq C$, the conditioning strip of a block lies entirely within blocks of opposite parity. Same-parity blocks are therefore conditionally independent given their blankets, and can all be updated at once in parallel. We call such a simultaneous update a *half-cycle* and a *cycle* the pair covering the whole trajectory, so that a full cycle costs two parallel passes rather than N sequential ones. This is precisely the 2-colorable model of Gonzalez et al. (2011), whose idea is to define a K-*coloring* of a Markov random field, and sample the variables (here blocks b_n) of a same color in parallel. Despite breaking the sequentiality of the classical Gibbs sampler, the parallel Chromatic sampler is ergodic and converges to the target distribution $p(x_{1:L} \mid y)$. GiBBS could generalize to any number of colors, but our checkerboard layout admits a natural two-coloring, which is maximally parallel. Gonzalez et al. (2011) further showed that given p processors, a K-coloring of the trajectory and N variables (here drawn together as blocks b_n), the algorithm generates a new joint sample in $O(N/p + K)$ running time. In our case, given sufficient parallel resources $p \in O(N)$, GiBBS generates a new joint sample in constant running time $O(1)$, independent of N. The number of cycles to converge however grows with the correlation length of the target posterior (Appendix C).

Shift GiBBS further introduces a periodic *shift* s_t of the block layout (Algorithm 1), which substantially improves mixing. With a fixed layout, the Markov blanket $\mathcal{M}(b_n)$ of a block always consists of the same states: they condition b_n at every cycle, but are never redrawn together with it, so a state and its neighbours across a block boundary are never resampled jointly and the chain mixes slowly. We therefore shift the whole layout by $\lfloor B/2 \rfloor$ frames every other cycle, so that states separated by a boundary in one cycle fall inside the same block in the next. Each half-cycle remains an exact block-Gibbs update, so the posterior is left invariant, and the shift costs nothing. Beyond mixing, it also shortens the burn-in of the chain, as shown in Section 3.

Algorithm 1: Gibbs Block Blanket Sampling (GiBBS)

Input: observations y, number of cycles T, block size B, Markov order k
Output: trajectory $x_{1:L} \sim p(x_{1:L} \mid y)$ (approximately)

$x_{1:L} \leftarrow$ `init()` `// data sample, AR pass, or AAO sample`
for $t = 1$ **to** T **do**
  $s_t = \big((t-1) \bmod 2\big)\lfloor B/2 \rfloor$
  Partition $x_{1:L}$ into $N = \lceil L/B \rceil$ disjoint blocks, $b_n = x_{(n-1)B+1+s_t\,:\,nB+s_t}$
  for $n \in \{1, 3, 5, \dots\}$ **do in parallel**
    $b_n \sim p\big(b_n \mid \mathcal{M}(b_n),\, y_{b_n}\big)$ `//` $\mathcal{M}(b_n) = \{b_{n-1}[-k:]\,;\, b_{n+1}[:k]\}$
  for $n \in \{2, 4, 6, \dots\}$ **do in parallel**
    $b_n \sim p\big(b_n \mid \mathcal{M}(b_n),\, y_{b_n}\big)$
return $x_{1:L}$

3 EXPERIMENTS

3.1 ORNSTEIN-UHLENBECK

We discretize the Ornstein-Uhlenbeck (OU) (Uhlenbeck & Ornstein, 1930) process as $x_{i+1} = \alpha x_i + \sqrt{1-\alpha^2}\,\varepsilon_i$, with $\alpha = 0.99$ and $\varepsilon_i \sim \mathcal{N}(0,1)$. The process being Gaussian and stationary, the exact denoiser follows analytically from Gaussian identities (Särkkä, 2013), so no network is trained in this section. Every posterior of interest is available in closed form as well: AR and GiBBS use an exact conditional denoiser, while AAO uses the prior denoiser its own recomposition scheme requires. All analytical denoisers are defined on windows of size $W = 16$, matching the short receptive fields assumed throughout.

For GiBBS, blocks are conditioned on a single neighboring frame on each side ($k = 1$), which suffices since OU processes are first-order Markov processes. The chain is initialized without any knowledge of the observations, from a random trajectory drawn from the dataset, and run for $T = 64$ cycles to ensure convergence.

3.1.1 INTERPOLATION TASK

The first and main task we consider is interpolation, which allows us to measure how information propagates, both qualitatively and quantitatively. The first and last windows of size $W = 16$ of a trajectory of length $L = 100$ are observed, leaving a gap of $L - 2W = 68$ unobserved frames in between. For AR and GiBBS, the observed frames are simply held fixed and enter the exact conditional denoiser. For AAO, they are imposed via zero-shot posterior sampling through MMPS (Rozet et al., 2024) with a small observation covariance $\sigma_y^2 = 10^{-5}$, chosen for numerical stability. Since the prior is Gaussian and the observation operator linear, the resulting guidance is exact for that observation model. The composition strategy therefore remains the only source of error.

Results Metrics are computed over 64 held-out trajectories, with 128 posterior samples each. AR visibly fails to anticipate future observations (Figure 2A): its trajectories drift back towards the prior across the unobserved gap, then are pulled abruptly onto the final constrained window, producing a discontinuity at the junction. The shape of the samples yielded by AAO does not follow the exact posterior envelope. GiBBS shows no such artifact and meet the constraints at both ends smoothly. The CRPS of the GiBBS chain decreases monotonically with the number of cycles and settles at the exact posterior level (Figure 2B), confirming convergence despite an initialization blind to the observations.

Table 1 confirms this quantitatively. GiBBS is well calibrated, attains the lowest skill and CRPS, and reaches the lowest W_1 to exact posterior samples of the three schemes, all while converging quickly to the posterior, forgetting its observation-blind start within twenty cycles.

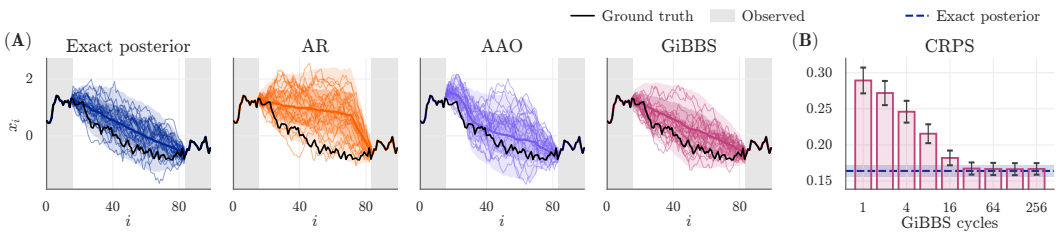


Figure 2. **(A)** Posterior samples for the Ornstein-Uhlenbeck interpolation task, 24 per method with their mean and quantile envelope. AR fails to anticipate the final observation window and reaches it through a visible discontinuity. **(B)** CRPS (↓) against the number of cycles. GiBBS converges to the exact posterior given enough cycles, even from an initialization blind to the observations.

Table 1. **Ornstein-Uhlenbeck** (mean ± std over 64 ground truths). Interpolation observes the first and last 16 frames of $L = 100$, data assimilation observes one frame in 15 of $L = 226$ with noise variance 0.5. W_1 is measured against the exact posterior on the unobserved frames.

SETUP	METHOD	Skill (↓)	SSR (→ 1)	CRPS (↓)	W_1 (↓)
Interpolation	AR	0.32 ±0.17	**0.98** ±0.68	0.22 ±0.12	4.01 ±1.41
	AAO	0.34 ±0.16	0.88 ±0.54	0.24 ±0.12	4.39 ±1.49
	GiBBS (ours)	**0.23** ±0.10	1.05 ±0.50	**0.17** ±0.06	**2.39** ±0.06
Data assimilation	AR	0.51 ±0.14	0.98 ±0.30	0.36 ±0.10	9.62 ±1.98
	AAO	0.40 ±0.09	0.96 ±0.25	0.29 ±0.06	7.49 ±0.66
	GiBBS (ours)	**0.35** ±0.07	**1.00** ±0.20	**0.25** ±0.05	**5.97** ±0.06

Effect of initialization and shift On this same setup, we study how the initialization of the chain and the layout shift s_t affect the burn-in and mixing of GiBBS.

The initialization of the chain affects the quality of its early samples. As shown in Figure 3A, an AAO start yields better samples than a blind start in the early cycles but shortens the burn-in only marginally. Indeed, AAO gets local portions of the trajectory right and its remaining error lies in the same slow, global mode as that of a blind start, which takes nearly as many cycles to correct. Conversely, starting from the exact posterior (available in closed form for OU) leaves the chain at the target, confirming that $p(x_{1:L} \mid y)$ is invariant under GiBBS. Adding independent white noise to each frame of that exact start shortens the burn-in. Frame-wise noise is mostly a fast and local error that a few cycles suffice to remove.

The shift acts on the performance of GiBBS at two levels. First, as shown in Figure 3B (left), shifting every block b_n by $\lfloor B/2 \rfloor$ frames every other cycle shortens the burn-in, halving the number of cycles required to reach the target posterior.

Second, it accelerates mixing. To decouple this effect from the burn-in, the 128 chains are initialized from the target posterior. Mixing is then measured through the autocorrelation of the chain: for a frame i, the Pearson correlation between two samples separated by Δ cycles is $\rho_i(\Delta) = \mathrm{Cov}(x_i^{(n)}, x_i^{(n+\Delta)})/\sigma_i^2$, estimated over the chains and all cycles n, which we average over the l frames of the interpolation gap, $\rho(\Delta) = \frac{1}{l}\sum_{i=1}^{l} \rho_i(\Delta)$. Shifting the block layout by $\lfloor B/2 \rfloor$ frames every other cycle decorrelates successive samples faster (Figure 3B, right), a sign of accelerated mixing.

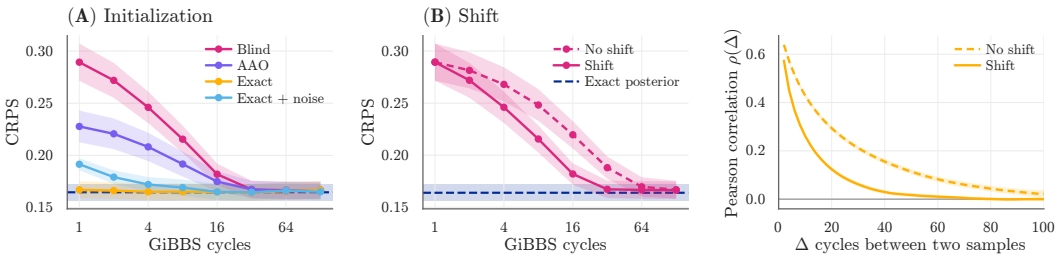


Figure 3. **Effect of initialization and shift on the burn-in and mixing. (A)** Influence of the initialization on the burn-in. **(B)** Influence of the shift on both the burn-in (left) and the mixing (right) of the chain.

3.1.2 DATA ASSIMILATION

The second task is a classical data assimilation setup: noisy observations are placed periodically along the trajectory, every 15 frames of $L = 226$, while the autocorrelation length of the system is $-1/\ln \alpha \approx 100$ frames. Despite substantial observation noise ($\sigma_y^2 = 0.5$), the strong correlation of the system carries their influence far enough. Figure 4 shows GiBBS converging to the exact posterior in this setup, while AR accumulates errors along its rollout despite the correlated observations.

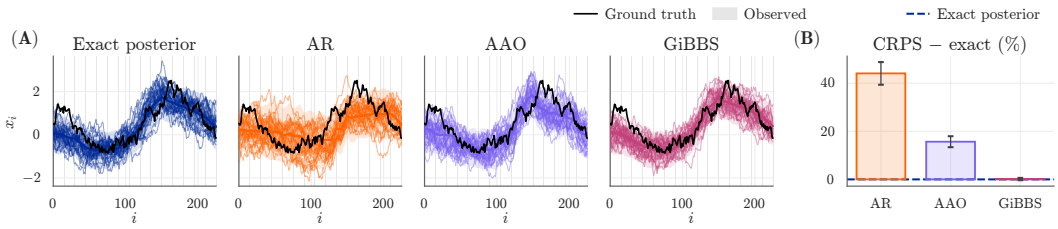


Figure 4. **(A)** Posterior samples for the Ornstein-Uhlenbeck Data Assimilation task, 24 per method with their mean and quantile envelope. One frame in 15 is observed with noise variance $\sigma_y^2 = 0.5$, a spacing much shorter than the correlation time of the process (≈ 100 frames). **(B)** CRPS excess (↓) over the exact posterior for all methods. While GiBBS converges given enough cycles, other methods remain substantially distant from the true target distribution.

3.2 Lorenz-63, Kuramoto-Sivashinsky & Kolmogorov

3.2.1 Experimental setup

Systems All systems in this section are chaotic, with increasing dimensionality. Lorenz-63 (Lorenz, 1963) is the 3-dimensional ODE $\dot{a} = \sigma(b - a)$, $\dot{b} = a(\rho - c) - b$, $\dot{c} = ab - \beta c$, with $\sigma = 10$, $\rho = 28$, $\beta = \frac{8}{3}$, integrated with a fourth-order Runge-Kutta scheme at $\Delta\tau = 0.025$. Kuramoto-Sivashinsky (KS) (Kuramoto, 1978; Sivashinsky, 1977) is a fourth-order nonlinear 1D PDE describing flame fronts, where the flame displacement u evolves as $\partial_\tau u + u\partial_x u + \partial_x^2 u + \nu\partial_x^4 u = 0$ with $\nu = 1$, solved by an exponential time-differencing scheme on a periodic domain of length 64 discretized over 64 points, at $\Delta\tau = 0.8$. Kolmogorov flow (Chandler & Kerswell, 2013) is a 2D incompressible Navier-Stokes flow at Reynolds number $Re = 1000$, whose velocity field $\boldsymbol{u}$ obeys $\dot{\boldsymbol{u}} = -(\boldsymbol{u} \cdot \nabla)\boldsymbol{u} + \frac{1}{Re}\nabla^2\boldsymbol{u} - \nabla p + \boldsymbol{f}$, with $\nabla \cdot \boldsymbol{u} = 0$. It is integrated in vorticity form at 256^2 and stored coarsened to 64^2, at $\Delta\tau = 0.2$. Despite their deterministic equations, the three systems are made stochastic (see Appendix B).

Architectural conditioning We train a single local score network per dataset, shared across all methods, using a *masked denoiser* as in MCVD (Voleti et al., 2022): a binary mask $b \in \{0, 1\}^{n+1}$ concatenated to the input marks which frames of the window $z^{i:i+n}$ are known and passed to the denoiser without noise. During training, the mask is drawn from four modes with equal probability: (1) *unconditional*, $b = \mathbf{0}$, used by AAO; (2) *forward*, $b = \mathbf{1}_{[0,\,C)}$ with $C \sim \mathrm{Uniform}(\{0, \dots, 2k\})$, used by AR; (3) *backward*, $b = \mathbf{1}_{[n+1-C,\,n+1)}$, its time-reversed counterpart; and (4) *two-sided*, $b = \mathbf{1}_{[0,\,C)\,\cup\,[n+1-C,\,n+1)}$ with $C \sim \mathrm{Uniform}(\{0, \dots, k\})$, required by GiBBS. Denoisers see windows of $W = 9$ frames for Lorenz-63 and KS, and $W = 5$ for Kolmogorov.

Each block b_n is conditioned on a strip of C states on either side. This strip coincides with the Markov blanket as soon as $C \geq k$, where k is the Markov order of the system; below that, the conditional is no longer exact. Lorenz-63 and KS are first-order Markovian, so $C = 1$ already suffices. For Kolmogorov, the shorter windows ($W = 5$) leave $C = 1$ as the only admissible choice. AR runs on the same grid: each step generates $B = W - 2C$ frames from the $2C$ previous ones, so both methods condition on the same number of frames, one-sided for AR and split across both sides for GiBBS. Further details are given in Appendix B.

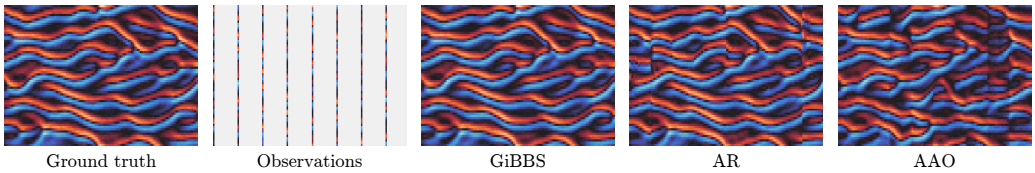
Observation setup Observations are generated as $y = \mathcal{A}(x) + \epsilon$, with $\epsilon \sim \mathcal{N}(0, \sigma_y^2 I)$ and $\sigma_y = 0.01$ in normalized units for Lorenz-63 and KS, 0.1 for Kolmogorov. The observation operator $\mathcal{A}$ determines which part of the state is seen: it can select a subset of frames, retain only some components, or coarsen the field spatially, and combinations thereof. For Lorenz-63 ($L = 128$), only the first component a is observed, on the first and last 9 frames. For KS ($L = 254$), we observe one frame in 16 at full spatial resolution, so sparsity is confined to the time axis. For Kolmogorov ($L = 32$), one frame in 4, spatially coarsened by an $8\times$ block mean: the hardest setup, sparse in both time and space. Metrics are computed over 64 held-out trajectories with 256 posterior samples each for Lorenz-63, and 48 ground truths with 16 members each for KS and Kolmogorov.

Initialization The chain converges from any initialization given enough cycles, but an informed start yields better samples in the early cycles (as shown in Figure 3A). For this reason, we initialize

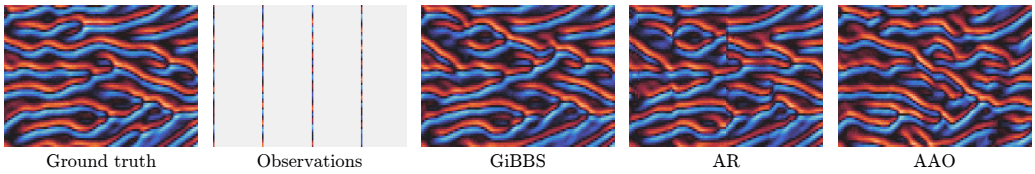
Lorenz-63 and KS from an AR pass, since AAO degrades over their sparse-in-time setting, and Kolmogorov from an AAO sample, which is cheaper as the trajectory is drawn in parallel. Using both lets us assess either option.

Table 2. Quantitative ensemble metrics (mean ± std) for different DA setups for all three learned systems. GiBBS is initialized from the AR pass on Lorenz-63 and Kuramoto-Sivashinsky and from the AAO draw on Kolmogorov. On Lorenz-63, AR with $C = 1$ diverges on 5.7% of its members. The starred entries are dominated by those samples. Appendix D.1.2 gives Lorenz-63 at $C = 2$, where no member diverges.

System	Method	Skill (↓)	SSR (→ 1)	CRPS (↓)
Lorenz-63	AR	1.36^{*} ±2.55	15.62^{*} ±4.59	0.23 ±0.08
	AAO	0.85 ±0.07	**0.90** ±0.06	0.44 ±0.04
	GiBBS (ours)	**0.51** ±0.20	1.16 ±0.31	**0.21** ±0.08
KS	AR	0.27 ±0.02	1.15 ±0.09	0.11 ±0.01
	AAO	0.43 ±0.02	1.26 ±0.05	0.16 ±0.01
	GiBBS (ours)	**0.12** ±0.01	**1.09** ±0.11	**0.05** ±0.002
Kolmogorov	AR	2.69 ±0.32	**0.99** ±0.06	1.33 ±0.17
	AAO	1.72 ±0.13	0.76 ±0.03	0.93 ±0.08
	GiBBS (ours)	**1.60** ±0.13	0.81 ±0.06	**0.84** ±0.07



(a) 1 out of 16 frames observed.



(b) 1 out of 32 frames observed.

Figure 5. **Segment of a KS trajectory sampled from intermittent, full spatial resolution observations, for all methods.** AR diverges between observation sites and is pulled back at each one, leaving visible discontinuities. AAO also displays discontinuities, but at the window level as well: given the small overlap between windows, little information is exchanged between them, which causes inconsistencies in the trajectory. GiBBS, initialized from an AR pass, resolves the rollout artifacts and stays closest to the truth.

3.2.2 Results

Table 2 reports ensemble metrics for the three schemes on the data assimilation setups described above. GiBBS obtains the best skill and CRPS on every system, with the largest margin on KS, where it halves the CRPS of AR and cuts that of AAO by more than half. Spread-skill ratios are more mixed: AR is best calibrated on Kolmogorov and AAO on Lorenz-63, but both at the cost of a much larger error (see CRPS).

Qualitative results illustrate how the composition strategies differ. On KS (Figure 5), the autoregressive rollout visibly diverges between observation sites before snapping back onto the next one, whereas AAO produces globally smoother trajectories that track the ground truth less closely. Figure 6 resolves the CRPS frame by frame to show where these errors arise. As expected, AAO fails to

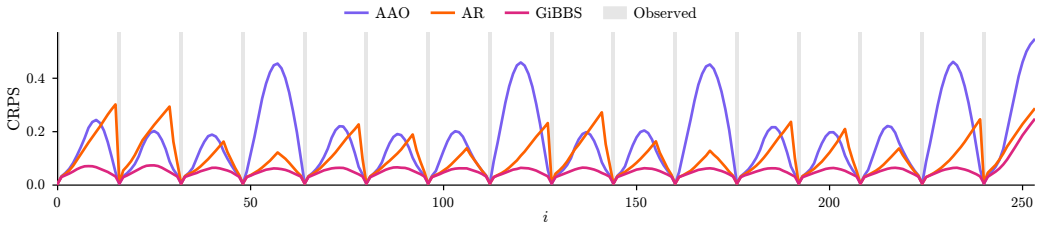


Figure 6. **CRPS over the trajectory for the three methods on Kuramoto-Sivashinsky**, averaged over 48 ground truths. AAO fails to propagate information away from the observation sites, while AR stays consistent longer but eventually drifts and is abruptly stitched back onto the observations. GiBBS propagates information correctly, reaching a markedly lower CRPS than the other methods, and does so without inheriting the divergence of the AR pass it is initialized from.

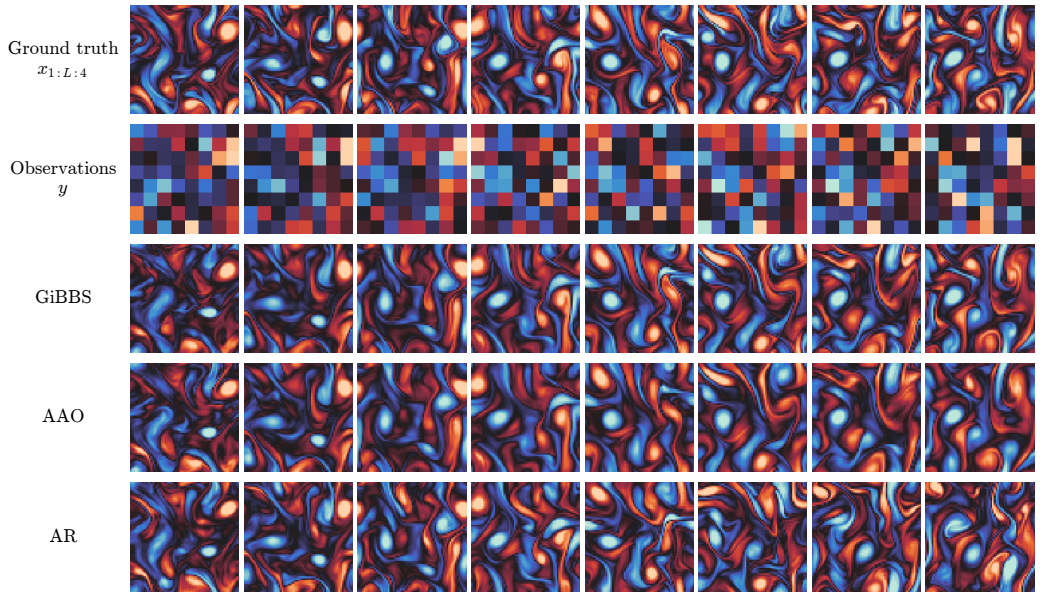


Figure 7. **One sample of a Kolmogorov trajectory for all methods**, in a 1 frame out of 4, 8× block mean observation setup, shown as vorticity at the observed frames $x_{1:L:4}$. Positive values (red) indicate counter-clockwise rotation and negative values (blue) clockwise rotation. GiBBS restores the fine details and high-frequency content of its AAO initialization, which gets the large-scale features right, but is overly-smoothed. AR fails to stay consistent, likely because observations that are coarse, intermittent and noisy constrain each window too weakly to correct the errors accumulated along its rollout.

propagate information away from the observation sites, while AR stays consistent longer but drifts as it advances without observations, blind to the future, and is abruptly pulled back once it reaches the next one. GiBBS, although initialized from an AR pass, inherits neither its drift nor its discontinuities, and propagates information correctly, with a much lower CRPS than both baselines.

On Kolmogorov (Figure 7), the assimilation setup is particularly hard, with observations both coarsened and sparse in time. AR accumulates errors steadily along its rollout (Appendix D.3) and no longer necessarily reconnects with the observations. AAO already performs well in this setting, adapted from the experiments of Rozet & Louppe (2023), but running GiBBS on top of it resolves the fine structure of the vortices. Appendix D.3 further reports two spatial subsampling setups on Kolmogorov and the extrapolation setup of Rozet & Louppe (2023). To assess the methods over a broader range of conditions, we run two ablations on KS, varying the proportion of observed frames and the observation noise (Figure 8). When observations are dense, they constrain every method

strongly and all perform alike; as they become sparser, GiBBS significantly improves the propagation of information over its initialization. The same holds for observation noise, as long as observations remain informative.

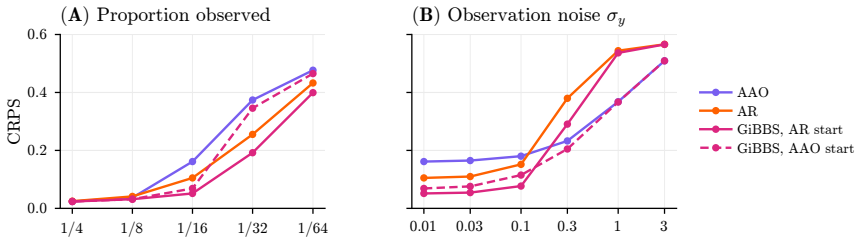


Figure 8. **Kuramoto-Sivashinsky: CRPS (↓) for (A) varying sparsity (fixed $\sigma_y = 0.01$) and (B) varying observation noise (1 frame in 16). (A)** When observations are dense, the three methods perform alike, as the states between observations are tightly constrained. As they become sparser, observations have to be carried further along the horizon, and GiBBS consistently improves on its initialization. **(B)** The same holds for observation noise up to a reasonable level. Once the noise is so large that observations are no longer informative, cycling stops improving the initialization, either because the trajectories are simply close to the prior, or because the learned models and zero-shot posterior sampling degrade out of distribution.

4 Discussion

We introduce GiBBS, a scheme for assembling local diffusion models into arbitrarily long trajectories that, unlike autoregressive rollouts and all-at-once sampling, converges to the target posterior $p(x_{1:L} \mid y)$. Casting composition as Gibbs sampling over disjoint blocks recovers the strengths of both baselines: blocks are conditioned on complete Markov blankets, so observations anywhere along the horizon constrain the trajectory, and the Markovian structure lets every block of a given parity be updated independently. On the Ornstein-Uhlenbeck process the chain converges to the exact posterior from an uninformed initialization, and on the chaotic Lorenz-63, Kuramoto-Sivashinsky and Kolmogorov systems it attains the best skill and CRPS of the three schemes.

While GiBBS asymptotically converges to the target posterior, we do not provide a rule regarding the number of cycles required to achieve convergence, nor a criterion to determine convergence. In practice the number of cycles is a hyperparameter, traded off against the cost of a reverse diffusion per block. Relatedly, block ordering is a natural direction of further study: it does not affect the stationary distribution, but markedly affects mixing (He et al., 2016). Therefore, schedules other than the odd-even one used here may require fewer cycles.

With analytical denoisers on Ornstein-Uhlenbeck, our experiments confirm the theoretical validity of the method. With learned denoisers, however, convergence holds only up to the approximations of the generative model and of the conditioning mechanism, here zero-shot posterior sampling. Quantifying these errors and how they affect convergence deserves further investigation, which we leave to future work.

Finally, although Kolmogorov flow is already high-dimensional, with tens of thousands of degrees of freedom, it remains far from operational data assimilation systems such as global weather forecasting, where a forecast of a few days reaches $O(10^{10})$ dimensions (Andry et al., 2025). We believe that GiBBS's ability to scale under operational conditions represents an important avenue for research.

AI USE STATEMENT

In this work, we used generative AI tools to assist with code development and with language editing of the manuscript, including translation and rephrasing of author-written drafts. We have not used generative AI tools for research ideation: the method, its theoretical grounding and the experimental design are our own. We have reviewed all AI-assisted work. LLM-generated code was verified and tested for correctness by us, and every AI-edited reformulation was checked against our original drafts to ensure its technical content was preserved. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Gérôme Andry, Sacha Lewin, François Rozet, Omer Rochman, Victor Mangeleer, Matthias Pirlet, Elise Faulx, Marilaure Grégoire, and Gilles Louppe. Appa: Bending weather dynamics with latent diffusion models for global data assimilation. *Machine Learning and the Physical Sciences Workshop (NeurIPS)*, 2025.

Luís M. A. Bettencourt, Ruy M. Ribeiro, Gerardo Chowell, Timothy Lant, and Carlos Castillo-Chavez. Towards real time epidemiology: Data assimilation, modeling and anomaly detection of health surveillance data streams. In Daniel Zeng, Ivan Gotham, Ken Komatsu, Cecil Lynch, Mark Thurmond, David Madigan, Bill Lober, James Kvach, and Hsinchun Chen (eds.), *Intelligence and Security Informatics: Biosurveillance*, pp. 79–90, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg.

Alberto Carrassi, Marc Bocquet, Laurent Bertino, and Geir Evensen. Data assimilation in the geosciences: An overview of methods, issues, and perspectives. *WIREs Climate Change*, 9(5): e535, 2018. doi: 10.1002/wcc.535.

Gary J. Chandler and Rich R. Kerswell. Invariant recurrent solutions embedded in a turbulent two-dimensional kolmogorov flow. *Journal of Fluid Mechanics*, 722:554–595, 2013. doi: 10.1017/jfm.2013.83.

Hyungjin Chung, Jeongsol Kim, Michael Thompson Mccann, Marc Louis Klasky, and Jong Chul Ye. Diffusion posterior sampling for general noisy inverse problems. In *The Eleventh International Conference on Learning Representations*, 2023.

François-Xavier Le Dimet and Olivier Talagrand. Variational algorithms for analysis and assimilation of meteorological observations: theoretical aspects. *Tellus A*, 38A(2):97–110, 1986. doi: 10.1111/j.1600-0870.1986.tb00459.x.

Alan E. Gelfand and Adrian F. M. Smith. Sampling-based approaches to calculating marginal densities. *Journal of the American Statistical Association*, 85(410):398–409, 1990. doi: 10.2307/2289776.

Stuart Geman and Donald Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-6 (6):721–741, 1984. doi: 10.1109/TPAMI.1984.4767596.

Michael Ghil and Paola Malanotte-Rizzoli. Data assimilation in meteorology and oceanography. In Renata Dmowska and Barry Saltzman (eds.), *Advances in Geophysics*, volume 33, pp. 141–266. Elsevier, 1991. doi: 10.1016/S0065-2687(08)60442-2.

Tilmann Gneiting, Adrian E. Raftery, Anton H. Westveld, and Tom Goldman. Calibrated probabilistic forecasting using ensemble model output statistics and minimum CRPS estimation. *Monthly Weather Review*, 133(5):1098–1118, 2005. doi: 10.1175/MWR2904.1.

Joseph Gonzalez, Yucheng Low, Arthur Gretton, and Carlos Guestrin. Parallel gibbs sampling: From colored fields to thin junction trees. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pp. 324–332. JMLR Workshop and Conference Proceedings, 2011.

Bryan He, Christopher De Sa, Ioannis Mitliagkas, and Christopher Ré. Scan order in gibbs sampling: Models in which it matters and bounds on how much. *Advances in neural information processing systems*, 29, 2016.

Langwen Huang, Lukas Gianinazzi, Yuejiang Yu, Peter D. Dueben, and Torsten Hoefler. DiffDA: a diffusion model for weather-scale data assimilation. In *Proceedings of the 41st International Conference on Machine Learning*, 2024.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

Yoshiki Kuramoto. Diffusion-induced chaos in reaction systems. *Progress of Theoretical Physics Supplement*, 64:346–367, 1978. doi: 10.1143/PTPS.64.346.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022.

A. C. Lorenc. Analysis methods for numerical weather prediction. *Quarterly Journal of the Royal Meteorological Society*, 112:1177–1194, 1986. doi: 10.1002/qj.49711247414.

Edward N. Lorenz. Deterministic nonperiodic flow. *Journal of the Atmospheric Sciences*, 20(2): 130–141, 1963. doi: 10.1175/1520-0469(1963)020<0130:DNF>2.0.CO;2.

Philip Nadler, Shuo Wang, Rossella Arcucci, Xian Yang, and Yike Guo. An epidemiological modelling approach for COVID-19 via data assimilation. *European Journal of Epidemiology*, 35 (8):749–761, 2020.

Dishant Pandya, Bhasha Vachharajani, and Rohit Srivastava. A review of data assimilation techniques: Applications in engineering and agriculture. *Materials Today: Proceedings*, 62:7048–7052, 2022. doi: 10.1016/j.matpr.2022.01.122.

François Rozet and Gilles Louppe. Score-based data assimilation. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

François Rozet, Gérôme Andry, François Lanusse, and Gilles Louppe. Learning diffusion priors from observations by expectation maximization. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

François Rozet. *Generative Modeling in Large-scale Dynamical Systems*. PhD thesis, University of Liège, Belgium, 2026.

Thomas Savary, François Rozet, and Gilles Louppe. Training-free bayesian filtering with generative emulators. In *International Conference on Machine Learning*, 2026.

Aliaksandra Shysheya, Cristiana Diaconu, Federico Bergamin, Paris Perdikaris, José Miguel Hernández-Lobato, Richard E. Turner, and Emile Mathieu. On conditional diffusion models for PDE simulations. *Proceedings of the 38th Conference on Neural Information Processing Systems (NeurIPS 2024)*, 2024.

G. I. Sivashinsky. Nonlinear analysis of hydrodynamic instability in laminar flames—i. derivation of basic equations. *Acta Astronautica*, 4(11–12):1177–1206, 1977. doi: 10.1016/0094-5765(77)90096-0.

Jiaming Song, Arash Vahdat, M. Mardani, and Jan Kautz. Pseudoinverse-guided diffusion models for inverse problems. In *International Conference on Learning Representations*, 2023.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021.

Simo Särkkä. *Bayesian Filtering and Smoothing*. Institute of Mathematical Statistics Textbooks. Cambridge University Press, 2013.

G. E. Uhlenbeck and L. S. Ornstein. On the theory of the brownian motion. *Physical Review*, 36(5): 823–841, 1930. doi: 10.1103/PhysRev.36.823.

Vikram Voleti, Alexia Jolicoeur-Martineau, and Christopher Pal. Mcvd: Masked conditional video diffusion for prediction, generation, and interpolation. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

A Evaluation metrics

For each of E ground-truth trajectories $x^e_{1:L}$, we draw an ensemble of M trajectories $\hat{x}^{e,m}_{1:L}$ from the posterior $p(x_{1:L} \mid y^e)$ and compute the metrics below for that ground truth. We then report their mean and standard deviation across the E ground truths.

Skill The root mean square error between the ensemble mean and the ground truth,

$$\mathrm{Skill}_i = \sqrt{\frac{1}{ED}\sum_{e=1}^{E}\sum_{d=1}^{D}\left(x^e_{i,d} - \frac{1}{M}\sum_{m=1}^{M}\hat{x}^{e,m}_{i,d}\right)^2}\,. \tag{5}$$

Spread The square root of the ensemble variance,

$$\mathrm{Spread}_i = \sqrt{\frac{1}{ED}\sum_{e=1}^{E}\sum_{d=1}^{D}\frac{1}{M-1}\sum_{m=1}^{M}\left(\hat{x}^{e,m}_{i,d} - \frac{1}{M}\sum_{n=1}^{M}\hat{x}^{e,n}_{i,d}\right)^2}\,. \tag{6}$$

Spread-skill ratio A calibrated ensemble has a spread matching its skill. Corrected for the finite ensemble size,

$$\mathrm{SSR} = \sqrt{\frac{M+1}{M}}\,\frac{\mathrm{Spread}}{\mathrm{Skill}}\,, \tag{7}$$

A ratio below one indicates an overconfident posterior, above one an overdispersed one. This is a necessary but not sufficient condition for calibration.

CRPS The continuous ranked probability score (Gneiting et al., 2005) assesses the full predictive distribution rather than its mean. We use the unbiased estimator

$$\mathrm{CRPS}_i = \frac{1}{ED}\sum_{e=1}^{E}\left(\frac{1}{M}\sum_{m=1}^{M}\left\|\hat{x}^{e,m}_i - x^e_i\right\|_1 - \frac{1}{2M(M-1)}\sum_{m=1}^{M}\sum_{n=1}^{M}\left\|\hat{x}^{e,m}_i - \hat{x}^{e,n}_i\right\|_1\right)\,. \tag{8}$$

The first term penalizes the distance to the ground truth, the second rewards ensemble spread, so the score is minimized when the sampled distribution matches the true posterior.

Wasserstein distance On OU only, where the posterior $p(x_{1:L} \mid y)$ is available in closed form, we additionally report the W_1 distance between the sampled and exact Gaussian posteriors.

B EXPERIMENT DETAILS

B.1 SYSTEMS

Table 3. Simulated systems and blanket geometry. W is the window size, C the size of the conditioning strip and $B = W - 2C$ the block size. Trajectory lengths satisfy $L = W + (N - 1)B$.

	State dimension	L	$\Delta\tau$	W	B	C	N
OU	1	100	—	16	14	1	7
Lorenz-63	3	128	0.025	9	7	1	18
KS	64	254	0.8	9	7	1	36
Kolmogorov	2×64^2	32	0.2	5	3	1	10

OU is simulated through its exact discretization, Lorenz-63 with RK4, KS with ETDRK4 using 8 substeps per outer step on a periodic domain of length 64 over 64 grid points, and Kolmogorov pseudo-spectrally in vorticity form at $Re = 1000$ on $[0, 2\pi]^2$.

B.2 STOCHASTICITY

We made Lorenz-63 and KS stochastic through noise added in the transition process, of standard deviation $\sigma\sqrt{\Delta\tau}$ so that its variance scales with the time step, as in an Euler-Maruyama discretization: $\sigma = 1$ for Lorenz (0.158 per step at $\Delta\tau = 0.025$), and $\sigma = 0.078$ for KS, spatially white (0.07 per step, so $\sim 5\%$ of the field's std at $\Delta\tau = 0.8$). Both sit above the observation noise, so the dominant uncertainty is the process rather than the sensor. Kolmogorov is stochastic by construction instead: it is solved at 256^2 and stored at 64^2, so the unresolved scales act as model error on the coarse dynamics.

B.3 MODELS

All models use EDM preconditioning (Karras et al., 2022) with a rectified noise schedule Liu et al. (2022), trained for 10^5 updates with AdamW on 1024 trajectories, with 256 held out for validation and 128 for testing. For Lorenz-63, the state has no spatial extent, so the convolutions run along time: a residual 1-D CNN over the three components. For KS and Kolmogorov the W frames of a window are stacked along the channel dimension and the convolutions act on space alone, with circular padding since both domains are periodic: a 1-D U-Net over the 64 grid points for KS, and a 2-D U-Net over the 64×64 velocity field for Kolmogorov. The same model is used for AR, AAO and GiBBS thanks to the masking strategy defined in Section 3.

Table 4. Architectures and training configurations.

	Hidden channels	Params	Batch	Learning rate	Schedule
Lorenz-63	[64]	132k	64	10^{-3}	linear
KS	[32, 64, 128, 128]	1.5M	64	3×10^{-4}	cosine
Kolmogorov	[96, 192, 384]	24.9M	128	5×10^{-4}	cosine

OU requires no network: its prior is Gaussian, so the exact denoiser is available in closed form (Särkkä, 2013).

B.4 SAMPLING

Trajectories are drawn with a predictor-corrector sampler (Song et al., 2021; Rozet & Louppe, 2023): each step runs a number of Langevin corrections of amplitude $\delta = 0.1$, followed by one predictor step.

Table 5. Sampling configurations. GiBBS is initialized from an AR pass on Lorenz-63 and KS, from an AAO sample for Kolmogorov and from an uninformed start (random trajectory from the dataset) for OU.

	Steps	Corrections	MMPS iterations	Members	GiBBS cycles
OU	128	2	32	128	64
Lorenz-63	128	2	2	256	16
KS	128	2	4	16	16
Kolmogorov	64	1	2	16	16

B.5 Observation setups

Observations are generated as $y = \mathcal{A}(x) + \epsilon$ with $\epsilon \sim \mathcal{N}(0, \sigma_y^2 I)$, where $\mathcal{A}$ selects frames, components or spatial block means (Table 6). On Ornstein-Uhlenbeck, the interpolation task is noiseless ($\sigma_y^2 = 10^{-5}$, kept non-zero for numerical stability for the conditioning of AAO, other setups are analytical), while the data assimilation task observes one frame in 15 with a substantial noise variance $\sigma_y^2 = 0.5$. On the learned systems, $\sigma_y = 0.01$ in normalized units for Lorenz-63 and KS, below their process noise, so the dominant uncertainty is the dynamics rather than the sensor; Kolmogorov uses $\sigma_y = 0.1$, the value used by Rozet & Louppe (2023) for the same observation process. On Lorenz-63, only the first component, a, is observed. The additional setups of Appendix D are described in their captions.

Table 6. Observation setups reported in the main text.

System	Setup	Noise	Observed fraction	Ground truths
OU (1st)	frames 0–15 and 84–99	$\sigma_y^2 = 10^{-5}$	32%	64
OU (2nd)	1 frame in 15	$\sigma_y^2 = 0.5$	7.1%	64
Lorenz-63	a only, frames 0–8 and 119–127	$\sigma_y = 0.01$	4.7%	64
KS	1 frame in 16, full resolution	$\sigma_y = 0.01$	6.3%	48
Kolmogorov	1 frame in 4, 8× block mean	$\sigma_y = 0.1$	0.391%	48

C GiBBS: additional details on convergence

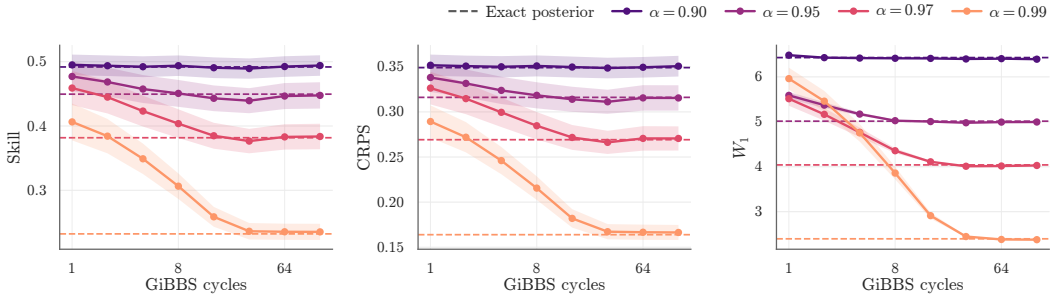


Figure 9. **Convergence on Ornstein-Uhlenbeck: ablation on α.** The autocorrelation length of OU is $l_{\text{autocorr}} = -1/\ln \alpha$, so a smaller α gives a shorter one. The shorter it is, the fewer GiBBS cycles are needed to reach the true posterior: for a given L, observations constrain a smaller neighborhood, leaving a posterior closer to the prior away from them, so less information has to travel along the horizon. The observation setup is the interpolation one, with $L = 100$, $B = 14$, $C = 1$, GiBBS initialized blindly from a trajectory of the process, metrics reported with 64 ground truths with 128 members each.

D ASSIMILATION RESULTS

D.1 LORENZ-63

D.1.1 QUALITATIVE RESULTS

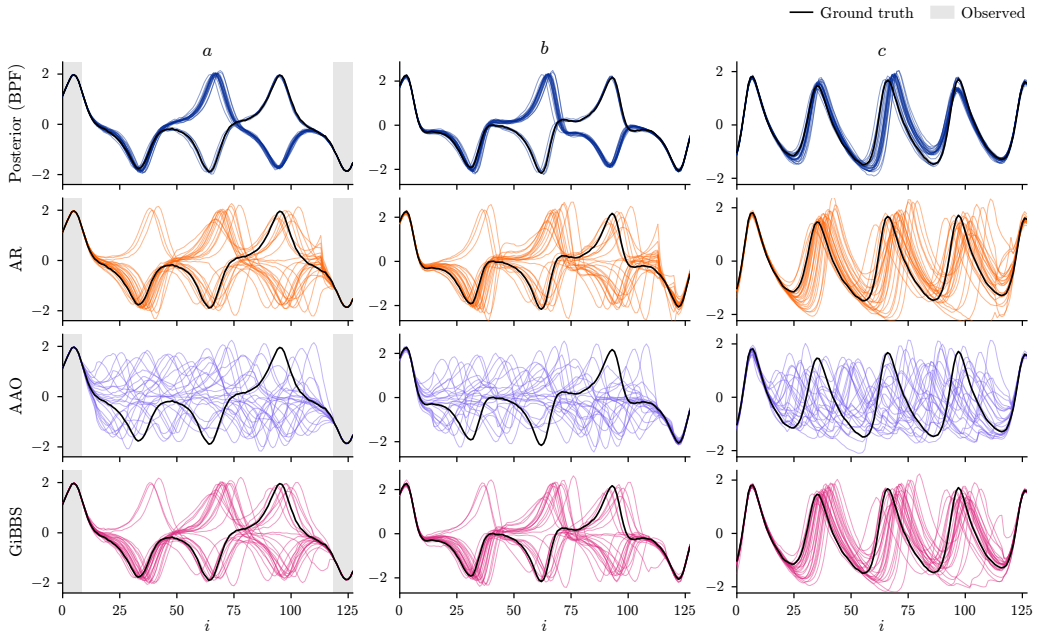


Figure 10. **Trajectories from Lorenz-63 in a sparse interpolation setup, for all methods.** Each panel shows 24 of 256 members over the ground truth (black). The top row is the reference: a bootstrap particle filter (BPF) with 2^{18} particles to draw trajectories from the exact posterior $p(x_{1:L} \mid y)$. AAO gives trajectories that no longer even resemble the prior and are outright discontinuous, a symptom of the limited overlap between windows. AR performs reasonably well until the end of its trajectories, which have drifted far from the last observation and are therefore stitched back onto it discontinuously. GiBBS resolves this and smoothly accommodates the last observed window.

D.1.2 ON THE BLOW UP OF C = 1 FOR THE AR

Table 7. Lorenz-63 with $C = 2$ (mean $\pm$ std over 64 ground truths). Same setup as Table 2, with blocks of $B = 5$ frames and a shift of 2. With a wider conditioning strip, AR no longer diverges: each window is generated from $2C = 4$ past frames instead of 2, so the rollout tracks the dynamics more closely and reaches the far observations with a smaller mismatch. At $C = 1$, observing a single component at both ends of a long gap makes the problem very sparse and ill-conditioned for AR. The rollout drifts across the gap, so that once it reaches the right-boundary window, the observations there sit far from its current state, and MMPS must pull it back abruptly, which destabilizes posterior sampling in that window.

	Lorenz-63, $C = 2$		
	Skill (↓)	SSR (→ 1)	CRPS (↓)
AR	$0.53_{\pm 0.20}$	$1.10_{\pm 0.23}$	$0.23_{\pm 0.09}$
AAO	$0.86_{\pm 0.07}$	$\mathbf{0.91}_{\pm \mathbf{0.06}}$	$0.45_{\pm 0.04}$
GiBBS	$\mathbf{0.49}_{\pm \mathbf{0.21}}$	$1.12_{\pm 0.27}$	$\mathbf{0.21}_{\pm \mathbf{0.09}}$

D.2 KURAMOTO-SIVASHINSKY

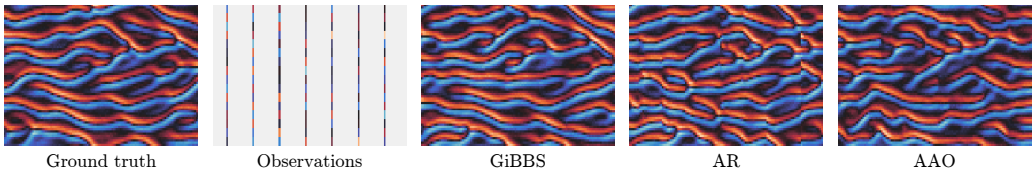


Figure 11. One sample of a segment of a KS trajectory for all methods, in a 1 frame out of 16, 4× block mean observation setup.

D.3 KOLMOGOROV

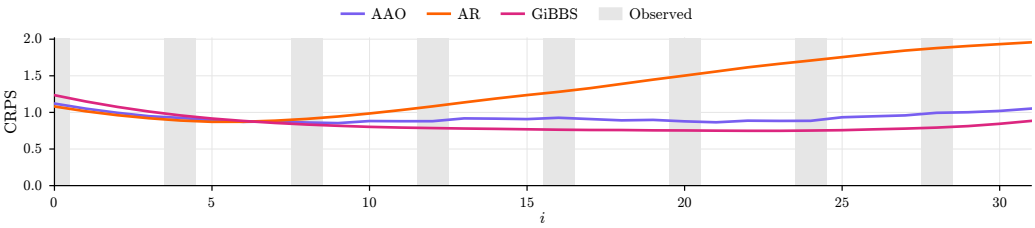
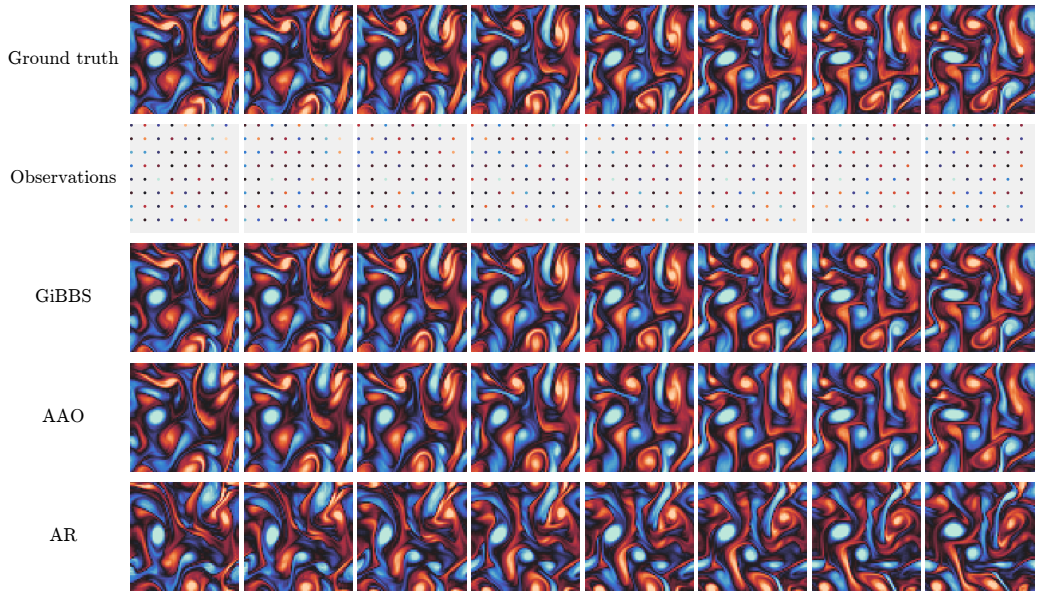


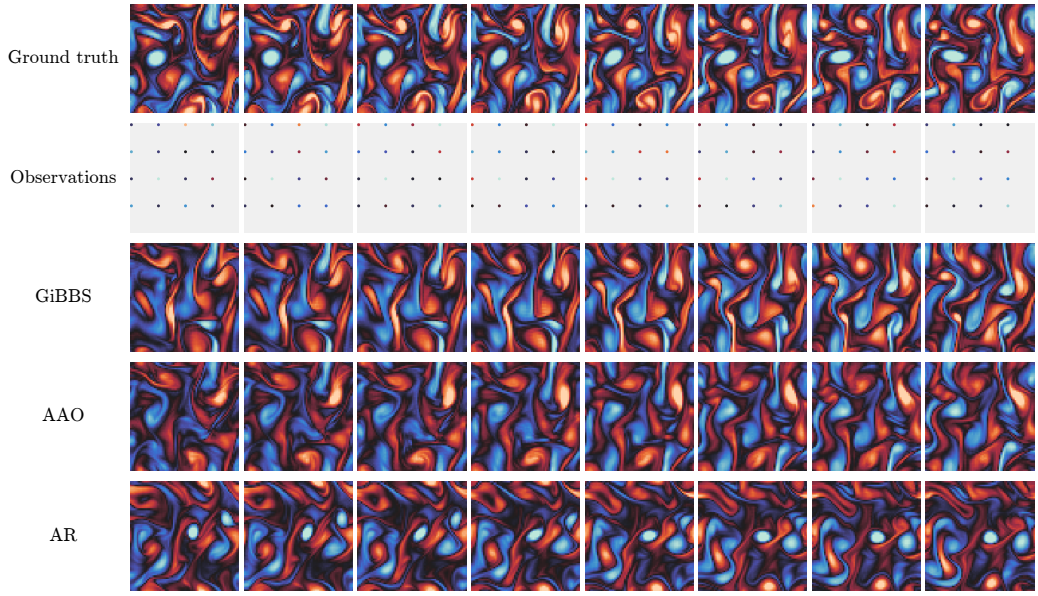
Figure 12. **CRPS over the trajectory for the three methods on Kolmogorov**, averaged over 48 ground truths. Observations are 8× block mean coarsened and are collected every 4 frames. The window size being $W = 5$, every one of them receives a noisy observation ($\sigma_y = 0.1$), which explains the absent drift-then-pull back phenomenon for AR, which accumulate errors along its rollout. AAO does not suffer from this divergence and GiBBS improves its CRPS.

E COMPUTE RESOURCES

All models were trained on a single NVIDIA RTX A5000 GPU on an internal academic cluster. Training takes under an hour for Lorenz-63, a few hours for KS, and about a day for Kolmogorov, whose network is two orders of magnitude larger. The code will be released with the extended version of this work.



(a) Subsampling every 8 grid points.



(b) Subsampling every 16 grid points.

Figure 13. **Spatial subsampling of the grid for Kolmogorov flow, $L = 32$, last 8 frames.** Every frame is observed on a regular subgrid, keeping one point every 8 (a) or 16 (b) grid points in each direction. GiBBS and AAO both recover the large-scale vortex structure, while AR drifts away from the ground truth along its rollout. GiBBS resolves some parts of the fine structures that were absent with AAO.

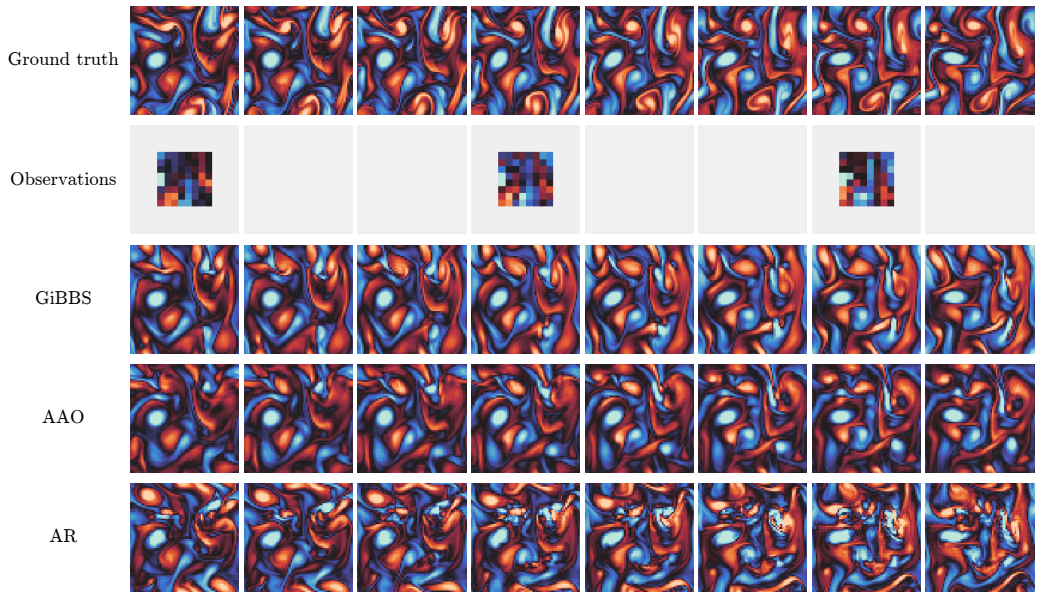


Figure 14. **Extrapolation setup for Kolmogorov flow, $L = 32$, last 8 frames.** The sensor reports a $4\times$ block mean of the velocity over the central 8×8 coarse cells only, on one frame in three, with noise $\sigma_y = 0.01$, as in the extrapolation setup of Rozet & Louppe (2023).