FORCINGDAS: UNIFIED AND ROBUST DATA ASSIMILATION VIA DIFFUSION FORCING

Anonymous authors
Paper under double-blind review

ABSTRACT

Data assimilation (DA) estimates the state of an evolving dynamical system from noisy, partial observations, and is widely used in scientific simulation as well as weather and climate science. In practice, filtering methods rely on frame-to-frame transition models. However, these models accumulate errors over long horizons, and they are further fragile when observations are non-Markovian. At the same time, learned DA methods typically commit to a single regime, either filtering (nowcasting, real-time forecasting) or smoothing (retrospective reanalysis), which splits what should be a shared prior across application-specific pipelines. To address both issues, we introduce **ForcingDAS**, a *unified* and *robust* DA framework. Built on Diffusion Forcing with an independent noise level assigned to each frame, ForcingDAS learns a joint-trajectory prior instead of frame-to-frame transitions. This allows it to capture long-horizon temporal dependencies and reduce error accumulation. In addition, the same trained model spans the full filtering to smoothing spectrum at inference time. Specifically, nowcasting, fixed-lag smoothing, and batch reanalysis are selected through the inference schedule alone, without retraining. We evaluate ForcingDAS on 2D Navier–Stokes vorticity, precipitation nowcasting, and global atmospheric state estimation. Across all settings, a single model is competitive with or outperforms both learned and classical baselines that are specialized for individual regimes.

1 INTRODUCTION

Accurate state estimation from noisy, incomplete data is critical across many scientific domains, such as weather forecasting (Bocquet et al., 2015; Reichle, 2008; Lahoz et al., 2010; Law et al., 2015; Carrassi et al., 2018), oceanography (Cummings, 2005; Cummings & Smedstad, 2013), and seismology (Werner et al., 2011; Banerjee et al., 2023). Mathematically, a discrete-time stochastic dynamical system is described by:

$$\boldsymbol{x}_{k+1} = \Psi(\boldsymbol{x}_k) + \boldsymbol{\xi}_k, \tag{1}$$

$$\boldsymbol{y}_{k+1} = \mathcal{A}(\boldsymbol{x}_{k+1}) + \boldsymbol{\eta}_{k+1}, \tag{2}$$

where $\boldsymbol{x}_k \in \mathbb{R}^D$ is the state with $1 \le k \le K$, and $\Psi(\cdot)$ is the transition map typically governed by nonlinear partial differential equations (PDEs) (Brunton & Noack, 2015; Duraisamy et al., 2019), $\boldsymbol{\xi}_k \in \mathbb{R}^D$ is stochastic forcing, and $\boldsymbol{y}_k \in \mathbb{R}^M$ is partial observation through the sensing operator $\mathcal{A}(\cdot)$ with noise $\boldsymbol{\eta}_k \sim \mathcal{N}(\mathbf{0}, \sigma_y^2 \boldsymbol{I}_M)$.

Data assimilation (DA) integrates physical models with observational data to yield physically consistent state estimates (Rabier, 2005; Geer et al., 2018; Fletcher, 2017; Carrassi et al., 2018). Depending on the temporal availability of observations, DA is generally classified into several distinct regimes: (*i*) *Filtering*, which recursively estimates the current state $\boldsymbol{x}_k$ "online" using observations $\boldsymbol{y}_{1:k}$ available up to the present time, serving as the basis for real-time forecasting and nowcasting; (*ii*) *Smoothing*, an "offline" batch process that refines past trajectories using both past and future observations $\boldsymbol{y}_{1:K}$, is used for retrospective reanalysis and climate science (e.g., the ERA5 reanalysis dataset is produced this way).

Limitations of existing DA solvers. Despite their importance, both classical and learned DA solvers commit to one regime by design. Classical methods carry well-known bottlenecks: the (Ensemble) Kalman Filter (Kalman, 1960; Evensen, 2003) assumes linear-Gaussian dynamics, particle filters

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.

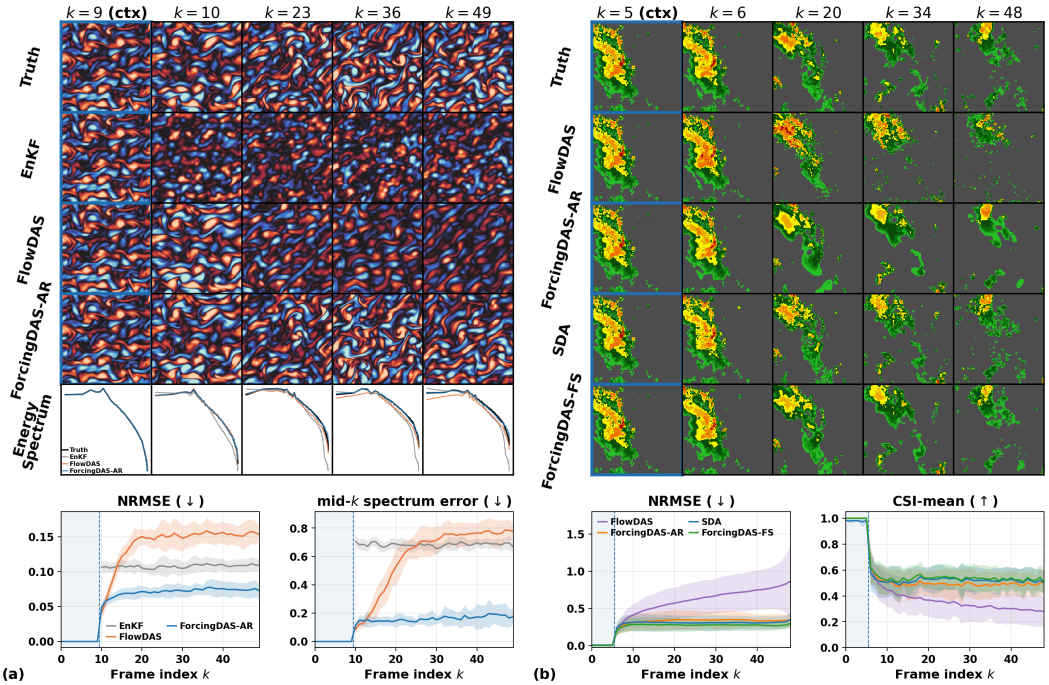

Figure 1: **ForcingDAS is unified and robust. (a)** Navier–Stokes vorticity (SO-5%) and **(b)** SEVIR VIL precipitation (SO-10%), with clean context frames marked in blue. A single trained ForcingDAS model runs as a filter (AR) and as a full-sequence smoother (FS) without retraining, and its error stays flat over the whole window, whereas the per-step learned filter FlowDAS accumulates error. Curves show mean $\pm$ 1 s.d. over the 16 held-out trajectories (a) or events (b); numbers are in Tables S3 and 1.

(Gordon et al., 1993) scale poorly with state dimension, and 4D-Var (Talagrand & Courtier, 1987; Courtier et al., 1994) requires hand-crafted adjoint models that are expensive to build and run at an operational scale. Recently, learned solvers alleviate the computational cost but are yet restricted to one regime only: trajectory diffusion priors such as SDA (Rozet & Louppe, 2023) target offline reanalysis and cannot run as real-time filters, whereas per-step models such as FlowDAS (Chen et al., 2025) and Fourier Neural Processes (Chen et al., 2024b) commit to learned one-step transitions, which (a) accumulate errors over long horizons since each frame is committed before the next observation arrives, and (b) condition only on a short, fixed window of recent frames. This limitation matters in real-world applications such as weather forecasting. Each measured frame (e.g., a few coarse-grid surface variables observed at a fixed cadence) captures only a partial slice of a much larger atmospheric state, which includes unresolved subgrid dynamics, unobserved vertical and chemical variables, and processes evolving faster than the observation interval. As a result, past observations can carry information about hidden degrees of freedom that are not contained in the current frame alone; consequently, the future is not merely determined by the present observation. In summary, across all these methods, the regime choice is baked in at design or training time.

Why do we need a unified DA approach? Ideally, one aims to learn a single model that generalizes across all DA paradigms, given that the underlying dynamical system remains consistent regardless of the observation window. For example, a weather center runs nowcasting (filtering) and reanalysis (smoothing) on the same atmosphere, and the two should share a single learned prior over the dynamics. Training separate models for each DA paradigm not only fragments operational pipelines but also fails to fully exploit the shared dynamical structure inherent in the data. Therefore, in this work, we propose training a unified model across all paradigms (i.e., filtering, smoothing, and forecasting), and the specific paradigm is only chosen at inference, analogous to a foundation model where one set of weights serves multiple downstream tasks.

Our approach: ForcingDAS. Following this principle, we propose **ForcingDAS**, a unified DA framework. We train a single joint-trajectory diffusion prior over full trajectories in which each frame

$\boldsymbol{x}_k$ carries its own diffusion-time noise level t_k, in contrast to standard video diffusion where all frames share one noise level (Chen et al., 2024a). At inference, this per-frame structure lets a scheduling matrix over the noise levels select the assimilation regime without retraining: a sequential schedule yields autoregressive filtering; a sliding-window pyramid schedule yields fixed-lag smoothing, with future observations refining past states via backward gradients; and a synchronous schedule yields full-sequence smoothing. Two further components close the gap to a usable solver: causality-aware training, which biases training noise toward the monotone patterns the scheduling matrix induces at inference, and noise-level-aware observation guidance, which scales each frame's correction by its current signal quality. See Figure 2 for a visual overview.

In summary, our contributions can be highlighted as follows.

1. **A unified data-assimilation framework.** We propose ForcingDAS (§3), to our knowledge, as the first unified model to encompass filtering, fixed-lag smoothing, and full-sequence smoothing. Crucially, the operational regime is flexibly determined at inference time via a specialized scheduling matrix.
2. **Robustness over long horizons.** Joint denoising across frames mitigates the error accumulation that has limited autoregressive learned DA, and modeling full trajectories rather than per-step transitions allows the prior to capture dependencies on hidden degrees of freedom that the current frame alone does not reveal (Fig. 1).
3. **Strong empirical results on three benchmarks.** A single ForcingDAS model is competitive with or outperforms specialized learned and classical baselines on 2D Navier–Stokes vorticity, SEVIR-VIL precipitation nowcasting, and ERA5 global atmospheric state estimation.

2 Background

2.1 Data Assimilation

The goal of DA is to estimate the state trajectory $\boldsymbol{x}_{1:K} = (\boldsymbol{x}_1, \dots, \boldsymbol{x}_K)$ of the dynamical system (1) from partial observations $\boldsymbol{y}_{1:K} = (\boldsymbol{y}_1, \dots, \boldsymbol{y}_K)$ described in (2). Two regimes are most relevant (Asch et al., 2016). **Filtering** targets $p(\boldsymbol{x}_k \mid \boldsymbol{y}_{1:k})$, the current-state posterior given only past and present observations, and is typically run online: a forecast from a predictive model is corrected by the latest observation to produce a state estimate consistent with both the dynamics and the data. **Smoothing** targets $p(\boldsymbol{x}_k \mid \boldsymbol{y}_{1:k+\Delta K})$, refining $\boldsymbol{x}_k$ using observations up to ΔK steps ahead; *fixed-lag smoothing* fixes ΔK to a rolling look-ahead window, while *full-sequence smoothing* sets $\Delta K = K - k$ and uses the entire observed sequence $\boldsymbol{y}_{1:K}$, the standard setting for offline reanalysis. An extended discussion of related work covering classical and learned DA solvers is deferred to Appendix S1.

2.2 Diffusion Forcing (DF)

A standard diffusion model (Ho et al., 2020; Sohl-Dickstein et al., 2015) corrupts a clean data sample $\boldsymbol{x}^{(0)}$ at a diffusion step t via $\boldsymbol{x}^{(t)} = \sqrt{\bar{\alpha}_t}\,\boldsymbol{x}^{(0)} + \sqrt{1-\bar{\alpha}_t}\,\boldsymbol{\epsilon}$, where $\bar{\alpha}_t$ is the cumulative signal-retention coefficient and $t \in \{1, \dots, T\}$ indexes the diffusion step. A neural network $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{x}^{(t)}, t)$ is trained to predict the noise; samples are drawn by iterating the reverse process (e.g., DDIM (Song et al., 2021)) from pure noise back to clean data.

Diffusion Forcing (DF) (Chen et al., 2024a) generalizes this to temporal sequences by assigning an *independent* diffusion step to each frame in a trajectory $\boldsymbol{x}_{1:K}$:

$$\boldsymbol{x}_k^{(t_k)} = \sqrt{\bar{\alpha}_{t_k}}\,\boldsymbol{x}_k + \sqrt{1-\bar{\alpha}_{t_k}}\,\boldsymbol{\epsilon}_k, \quad \boldsymbol{\epsilon}_k \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}), \quad k = 1, \dots, K. \tag{3}$$

A single denoising network $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{x}_{1:K}^{(\boldsymbol{t})}, \boldsymbol{t})$ takes the full noisy trajectory together with the per-frame diffusion-step vector $\boldsymbol{t} = (t_1, \dots, t_K)$ and predicts the noise on each frame. The temporal backbone uses *causal attention*: frame k attends only to frames $k' \le k$. At sampling time, a **scheduling matrix** $\boldsymbol{S} \in \{0, \dots, T\}^{K \times L}$ specifies how each frame's diffusion step descends across the L reverse-process iterations (i.e., $S_{k,\ell}$ is the diffusion step of frame k at iteration ℓ, with $S_{k,0} = T$ and $S_{k,L} = 0$); different scheduling matrices recover different sampling strategies from the same trained network. A more complete recap of DF and its relation to ForcingDAS is given in Appendix S1.2.

3 PROPOSED METHOD

Building on Section 2, we introduce **ForcingDAS**, a unified DA framework that elevates DF from a generative modeling paradigm to a complete and versatile DA solver, seamlessly unifying filtering, fixed-lag smoothing, and full-sequence smoothing within a single trained model and selecting the regime entirely at inference through the scheduling matrix. We first identify three challenges that prevent DF from being used as a plug-and-play DA solver, give an overview of how ForcingDAS addresses them (§3.1), and then provide extra implementation details (§3.2).

Challenges of Applying Diffusion Forcing to DA Although the per-frame noise structure makes DF a natural candidate prior for DA, three challenges prevent it from being used as a plug-and-play DA solver. (*i*) *Causal mismatch between training and inference.* Every DA scheduling matrix imposes a causally monotone pattern at inference due to the underlying physics, but DF training samples $\boldsymbol{t}$ i.i.d. uniformly, leaving the model effectively less trained on the configurations it sees most at inference; this gap is benign on natural-video data but matters for scientific dynamics, as we discuss in detail in §3.1. (*ii*) *No mechanism for observation guidance.* DF models only the prior $p_{\boldsymbol{\theta}}(\boldsymbol{x}_{1:K})$, lacking a mechanism to integrate noisy partial measurements $\boldsymbol{y}_{1:K}$ during the reverse sampling process. However, enforcing these observation constraints is essential for state correction and accurate prediction. (*iii*) *Limited adaptivity to different DA regimes.* Using a single trained model to unify filtering, fixed-lag smoothing, and full-sequence smoothing requires a principled family of scheduling matrices. However, as DF employs arbitrary scheduling matrices, it does not explicitly adapt to the corresponding DA regimes.

3.1 OVERVIEW OF OUR METHOD

We address the three challenges above in turn, with one targeted innovation for each.

Causality-Aware Training (CAT). Standard DF draws the per-frame diffusion step vector $\boldsymbol{t}$ i.i.d. from $\mathrm{Uniform}\{0,\dots,T\}^K$, whereas every DA scheduling matrix imposes a *causally monotone* pattern at inference (earlier frames at no-higher diffusion steps than later ones). This discrepancy is generally acceptable for natural videos, as adjacent frames are often redundant. However, it significantly impacts the scientific systems considered in this work (NS, SEVIR, ERA5). Because these systems exhibit broadband spatial spectra and strong forward-in-time dependence (Mojgani et al., 2024), a clean historical state conveys fundamentally more information than a noisy current observation. Therefore, it is critical to train the model using the identical clean-past and noisy-future conditioning it experiences at inference time. We therefore replace the i.i.d. sampling with a mixture that biases training toward causally monotone patterns while keeping i.i.d. samples as a regularizer:

$$\mathcal{L}_{\mathrm{CAT}} = \mathbb{E}_{\boldsymbol{x}_{1:K}}\, \mathbb{E}_{\boldsymbol{t}\sim p_\rho} \sum_{k=1}^{K} \lambda(t_k) \left\| \boldsymbol{\epsilon}_k - \boldsymbol{\epsilon}_{\boldsymbol{\theta}}\Big(\boldsymbol{x}_{1:K}^{(\boldsymbol{t})}, \boldsymbol{t}\Big)_k \right\|^2, \qquad p_\rho = \rho\, p_{\mathrm{sorted}} + (1-\rho)\, p_{\mathrm{iid}}, \quad (4)$$

where p_{iid} is the standard $\mathrm{Uniform}\{1,\dots,T\}^K$, p_{sorted} is its pushforward under sorting in non-decreasing order, $\rho \in [0,1]$ is the *causal ratio*, and $\lambda(t_k)$ is an optional per-step weight; $\rho = 0$ recovers standard DF. Detailed motivation and ablations on ρ are in Appendix S3.4.

Observation-Consistent Sampling. DF's reverse process samples from the trajectory prior $p_{\boldsymbol{\theta}}(\boldsymbol{x}_{1:K})$ but provides no mechanism to condition on partial measurements $\{\boldsymbol{y}_k\}$. We close this gap with a measurement-likelihood gradient added to each DDIM step, in the spirit of test-time gradient guidance for inverse problems (Chung et al., 2023). A constant guidance scale, however, is suboptimal in the per-frame setting where different frames are simultaneously at different diffusion steps: strong corrections distort heavily noised frames, while weak corrections miss fine detail on clean ones. We derive an adaptive weighting from the Tweedie estimate's error variance. Recall $\hat{\boldsymbol{x}}_k^{(0)} = (\boldsymbol{x}_k^{(t_k)} - \sqrt{1-\bar\alpha_{t_k}}\,\boldsymbol{\epsilon}_{\boldsymbol{\theta}})/\sqrt{\bar\alpha_{t_k}}$; the estimation error $\hat{\boldsymbol{x}}_k^{(0)} - \boldsymbol{x}_k$ has variance scaling as $(1-\bar\alpha_{t_k})/\bar\alpha_{t_k}$, so the total variance of the observation residual is $\sigma_y^2 + \gamma\,(1-\bar\alpha_{t_k})/\bar\alpha_{t_k}$ with $\gamma > 0$ controlling the contribution of the prediction uncertainty. Weighting each frame's squared residual by the inverse of this variance, and taking the square root as in the unsquared residual norm of Chung et al. (2023), yields

$$\mathcal{L}_{\mathrm{obs}} = \Big(\sum_{k=1}^{K} w(t_k) \cdot \Big\| \boldsymbol{y}_k - \mathcal{A}(\hat{\boldsymbol{x}}_k^{(0)}) \Big\|_2^2 \Big)^{1/2}, \quad w(t_k) = \frac{1}{\sigma_y^2 + \gamma\, \frac{1-\bar\alpha_{t_k}}{\bar\alpha_{t_k}}}, \quad (5)$$

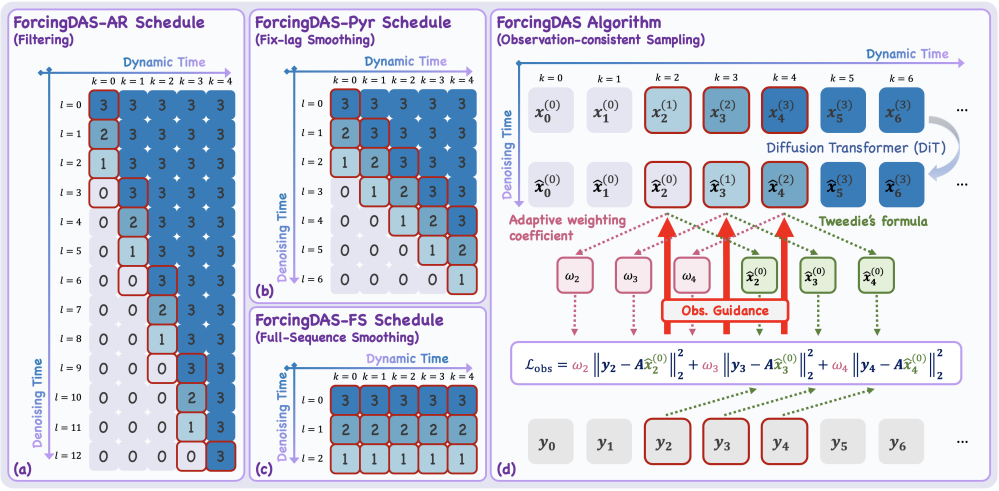

Figure 2: **ForcingDAS at a glance.** (a–c) The scheduling matrix $\boldsymbol{S}^{(u)}$ (rows: denoising iteration ℓ; columns: trajectory frame k; cells: diffusion step) selects the DA regime via the uncertainty scale u: autoregressive filtering ($u = T$), pyramid fixed-lag smoothing ($0 < u < T$), and full-sequence smoothing ($u = 0$). Active frames at each iteration are outlined in red. (d) Observation-consistent sampling: per-frame Tweedie estimates $\hat{\boldsymbol{x}}_k^{(0)}$ are scored against observations $\boldsymbol{y}_k$ through adaptive weights w_k tied to each frame's noise level (Eq. 5); the aggregated loss $\mathcal{L}_{\text{obs}}$ is backpropagated through the shared DiT backbone, routing later-frame observation gradients to earlier active frames.

so that frames with reliable Tweedie estimates receive stronger guidance and heavily noised frames are down-weighted. The gradient correction at each DDIM step is

$$\boldsymbol{x}_{1:K}^{(t-1)} \leftarrow \text{DDIM-step}(\boldsymbol{x}_{1:K}^{(t)}) - \zeta \nabla_{\boldsymbol{x}_{1:K}^{(t)}} \mathcal{L}_{\text{obs}}. \tag{6}$$

For schedules with multiple simultaneously active frames at different diffusion steps that we discuss in the following, the residuals in Eq. (5) are summed over the active window before differentiation; causal attention then routes each later frame's gradient backward to earlier active frames, producing the smoothing pathway analyzed in Appendix S2.1.

Unified DA View of the Scheduling Matrix. Building on these innovations for training and sampling in the DA setting, we now establish a unifying framework. We demonstrate that the scheduling matrix inherently acts as a DA-algorithm selector, defining a single-parameter family that seamlessly spans the entire filtering-smoothing spectrum. While the original DF formulation permits arbitrary scheduling matrices, it lacks this algorithmic interpretation. By making this mapping explicit, we enable a single scalar $u \geq 0$ for *uncertainty scale* to recover any standard DA regime without requiring any model retraining. We describe the elements of the scheduling matrix $\boldsymbol{S}$ as

$$S_{k,\ell}^{(u)} = \text{clip}\Big(T - \ell + u \cdot (k-1),\ 0,\ T\Big), \quad k = 1, \ldots, K, \ \ \ell = 0, \ldots, L-1, \tag{7}$$

where $L = T + u(K-1)$ and $\text{clip}(\cdot, 0, T)$ saturate each entry to the valid diffusion-step range $[0, T]$. As shown in Figure 2, the schedule continuously interpolates between three DA regimes by varying u: (i) *filtering* ($u = T$), where each frame fully denoises before the next begins, with posterior $p(\boldsymbol{x}_k \mid \boldsymbol{x}_{1:k-1}, \boldsymbol{y}_{1:k})$; (ii) *fixed-lag smoothing* ($0 < u < T$, pyramid), where $W \approx \lceil T/u \rceil$ frames are concurrently active at staggered diffusion steps, and observations on later active frames refine earlier ones through windowed observation aggregation; and (iii) *full-sequence smoothing* ($u = 0$), where all frames descend in lockstep, every observation jointly refines every frame, and the inference reduces to a batch smoother under the constraint of causal attention (see Appendix S2.1 for the past–future asymmetry that this entails).

Advantages of ForcingDAS. In contrast to specialized solvers, ForcingDAS offers two distinct advantages: (*i*) *Unification across the DA spectrum*, our approach enables a single trained model to perform filtering, fixed-lag, and full-sequence smoothing. By adjusting the uncertainty scale u at inference, the model adapts to different observation windows, such as real-time forecasting or

retrospective reanalysis, similar to how foundation models support multiple downstream tasks. (*ii*) *Robustness over long horizons*. By modeling the joint trajectory prior rather than a per-step transition, ForcingDAS employs joint denoising to suppress the autoregressive error accumulation that limits traditional learned solvers.

3.2 Implementation Details

Training. For each minibatch trajectory $\boldsymbol{x}_{1:K}$, we sample the per-frame diffusion-step vector $\boldsymbol{t} = (t_1, \ldots, t_K)$ from the CAT mixture in Eq. (4): with probability ρ we sort an i.i.d. $\mathrm{Uniform}\{1, \ldots, T\}$ draw into a non-decreasing staircase, otherwise we keep the i.i.d. draw; with probability ρ_c we additionally clamp a random number of leading frames to diffusion step zero, so the model is exposed to inference-style clean-context conditioning. We then corrupt the trajectory according to Eq. (3), evaluate the score-matching loss in Eq. (4), and update the network parameters. Pseudocode (Algorithm S3), the DiT-3D backbone with causal temporal attention, frame stacking, noise clipping, and full optimizer settings are in Appendix S2.

Inference. Given a trained model and observations $\{\boldsymbol{y}_k\}$, we initialize $\boldsymbol{x}_k^{(T)} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$ for $k = 1, \ldots, K$ and step through the L scheduled iterations of $\boldsymbol{S}^{(u)}$ in Eq. (7). At iteration ℓ, the active set $\mathcal{K}_{\text{active}}$ collects the frames whose diffusion step decreases between ℓ and $\ell + 1$. For these frames, we (*i*) compute the Tweedie's estimate $\hat{\boldsymbol{x}}_k^{(0)}$ via the denoising network, (*ii*) form the noise-level-aware reweighted observation residual in Eq. (5) aggregated over $\mathcal{K}_{\text{active}}$, (*iii*) backpropagate to obtain a single gradient through the shared denoising network, and (*iv*) apply one DDIM step combined with the gradient correction in Eq. (6). Iterating over $\ell = 0, \ldots, L-1$ produces a clean trajectory $\boldsymbol{x}_{1:K}^{(0)}$ that is consistent with both the trained prior and the observations. Pseudocode (Algorithm S4) and per-experiment hyperparameters (ζ, γ) are in Appendix S2.

4 Experiments

We evaluate ForcingDAS on three benchmarks. 2D Navier–Stokes vorticity (§4.1) is a PDE whose vorticity field is the full state, and for which a classical ensemble filter can run the exact dynamics; SEVIR-VIL precipitation nowcasting (§4.2) and ERA5 global atmospheric state estimation (§4.3) are real-world benchmarks, where the observed sequence is a low-dimensional projection of a higher-dimensional latent state. Existing learned DA methods on ERA5 (Appendix S1.1, Table S1) each commit to a single regime, either filtering or window-based 4D-Var. We evaluate a single trained model in three regimes (filtering, fixed-lag smoothing, and full-sequence smoothing) under our *with-context* protocol; the harder *cold-start* setting is in Appendix S4. The fixed-lag window is $\min(K, \lceil T/u \rceil)$ frames for an assimilation horizon of K frames and $T = 100$ DDIM steps: we use a 10-frame lag ($u = 10$) on NS and ERA5 and a 20-frame lag ($u = 5$) on SEVIR. For all learned methods, sampling hyperparameters are selected on held-out tuning data disjoint from the evaluation set, or taken from the original paper.

4.1 Navier–Stokes Vorticity

We use NS as a controlled benchmark on a PDE-governed dynamical system: the vorticity field is the full state of 2D incompressible flow with known dynamics. Even in this setting, sparse-observation filtering is challenging because per-step error accumulates across long horizons.

Setup, baselines, and metrics. We consider 2D incompressible Navier–Stokes vorticity at 128×128 resolution, evaluated on 16 held-out trajectories. In the main protocol each trajectory has $K = 50$ frames, the first 10 clean as context and the remaining 40 assimilated and scored. The operator is sparse pixel masking at a 5% observed ratio (**SO-5%**, one fixed mask) with additive Gaussian noise of $\sigma_y = 0.05$ in the $[0, 1]$ data space (1.83 in raw vorticity units). Observations are generated once per trajectory and every method assimilates the identical values, converted to its own units. The cold-start protocol ($K = 30$, no clean context) is in Appendix S4.1. We compare against the learned filter FlowDAS (Chen et al., 2025), which is built on a per-step transition model, and the classical Ensemble Kalman Filter (EnKF) (Evensen, 2003), which runs the exact PDE (Appendix S5.5). The appendix tables additionally report the Ensemble Kalman Smoother (EnKS, fixed-lag and full-sequence) and the learned smoother SDA (Rozet & Louppe, 2023). ForcingDAS-Pyr uses a 10-frame lag. We report NRMSE and the relative error of the radially averaged kinetic-energy spectrum: for each frame we

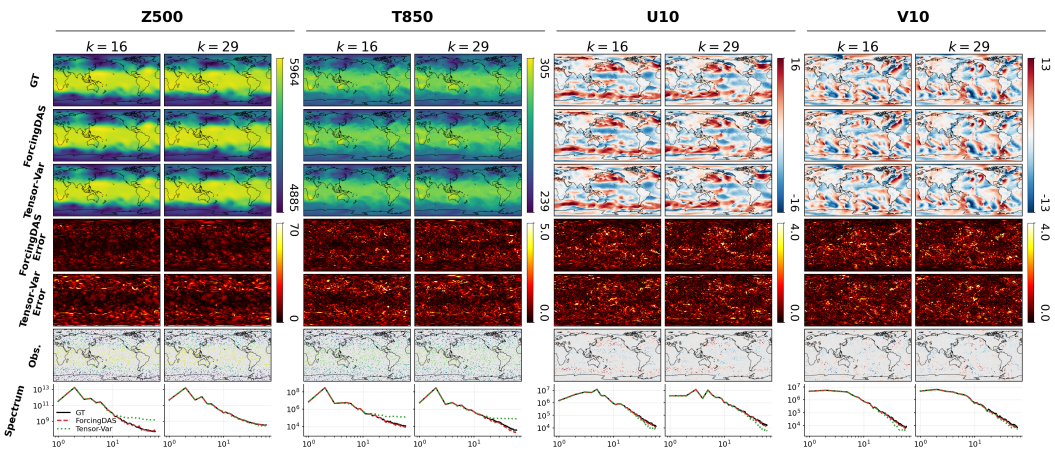

Figure 3: ERA5 SO-10% with-context assimilation on the first of the 16 evaluation trajectories, for all four variables (Z500 in m, T850 in K, U10 and V10 in m s^{-1}), at a mid-window frame ($k = 16$) and the last frame ($k = 29$). ForcingDAS runs as a full-sequence smoother (FS), the regime matched to Tensor-Var. The last row is the per-frame power spectrum along latitude (log–log). Numbers are in Table 2; the cold-start counterpart is Fig. S3.

compute the radial spectrum $E(k)$ of the predicted and ground-truth vorticity and the per-bin relative error $e(k) = |E_{\text{pred}}(k) - E_{\text{gt}}(k)|/\max(E_{\text{gt}}(k), \epsilon)$, averaged over wavenumber bands $\mathcal{K}$; we report mid-k ($\mathcal{K} = [8, 32)$, the inertial range) and the integrated all-k error ($\mathcal{K} = [0.5, 64)$). All numbers are mean $\pm$ std over the 16 trajectories. Dataset details and full hyperparameters are in Appendix S3.1.

Results. Table S3 reports the with-context numbers. In the filtering regime ForcingDAS-AR halves FlowDAS's NRMSE (0.071 vs. 0.141) and cuts the all-k spectrum error from 0.617 to 0.215; it also improves on the EnKF (0.108), which runs the exact PDE. Because every method assimilates identical observations on the same trajectories, the comparison is paired: ForcingDAS-AR is better than both FlowDAS and EnKF on all 16 trajectories and on all three metrics (Wilcoxon signed-rank $p = 3.1 \times 10^{-5}$, the minimum attainable for $n = 16$). Within each smoothing regime, ForcingDAS also improves on the matched smoothers on all 16 trajectories (Table S3): ForcingDAS-Pyr on EnKS-FL over the same 10-frame lag (0.057 vs. 0.113), and ForcingDAS-FS on EnKS-FS (0.067 vs. 0.116) and on SDA given the same clean context (0.067 vs. 0.075). Fig. 1(a) shows why FlowDAS falls behind: its error grows over the first $\sim$10 assimilated frames, after which its fields lose the small-scale eddies, while ForcingDAS-AR stays close to the truth across the full window. EnKF's error is flat but high, and its spectrum decays too fast at high wavenumbers.

4.2 Precipitation Nowcasting (SEVIR-VIL)

Setup. We use the Vertically Integrated Liquid (VIL) product of the Storm Event Imagery (SEVIR) dataset (Veillette et al., 2020), a vertically integrated radar-derived measure of precipitation intensity. We use a SEVIR-LR build that keeps all 49 frames of each event at 5-minute cadence (single channel, 128×128), with a random 90/10 split into training and held-out events. We evaluate on 16 held-out events with enough convective activity (more than 5% of pixels above VIL 16; Appendix S3.2.1). In the main protocol, 6 clean context frames seed a 49-frame trajectory and the remaining 43 frames are assimilated and scored. We evaluate four observation operators with additive Gaussian noise $\sigma_y = 0.05$ in the $[0, 1]$ data space: sparse pixel masks observing 10% and 20% of pixels (**SO-10%**, **SO-20%**) and bilinear down-sampling by $2\times$ and $4\times$ (**SR2x**, **SR4x**). The observations are generated once per event and every method assimilates the identical values. ForcingDAS-Pyr uses a 20-frame lag. The cold-start protocol (30 frames, no clean context) is in Appendix S4.2.

Baselines and metrics. We compare against the learned filter FlowDAS (Chen et al., 2025), retrained on our SEVIR build, and the learned smoother SDA (Rozet & Louppe, 2023). SEVIR has no operationally meaningful classical-DA baseline, since VIL dynamics are not given by a known PDE. SDA has no mechanism for clean context frames, so we give it the same 6 clean frames by replacement conditioning: at every sampling step the context frames are overwritten with the ground truth diffused

Table 1: SEVIR VIL nowcasting with 6 clean context frames and 43 assimilated frames: NRMSE (↓; mean ± std over 16 events) and CSI-mean (↑) over thresholds $\{16, 74, 133, 160, 181, 219\}$ (pooled over all events and frames; ± std over bootstrap resamples of the events). ForcingDAS-Pyr uses a 20-frame lag; SDA receives the same 6 clean frames by replacement conditioning. Best in *Filtering* and best in *Full-sequence Smoothing* per column in **bold**. Cold-start results are in Table S14. CSI-160 and the other per-threshold CSIs are in Table S5.

	SO-10%		SO-20%		SR2x		SR4x	
Method	NRMSE	CSI-m	NRMSE	CSI-m	NRMSE	CSI-m	NRMSE	CSI-m
Filtering								
FlowDAS	0.616 ± 0.213	0.403 ± 0.020	0.479 ± 0.183	0.481 ± 0.023	0.363 ± 0.199	0.637 ± 0.017	0.504 ± 0.189	0.471 ± 0.024
ForcingDAS-AR	**0.330** ± 0.107	**0.557** ± 0.024	**0.258** ± 0.080	**0.625** ± 0.020	**0.199** ± 0.054	**0.684** ± 0.017	**0.313** ± 0.100	**0.558** ± 0.026
Fix-lag Smoothing								
ForcingDAS-Pyr	0.293 ± 0.093	0.584 ± 0.021	0.214 ± 0.063	0.671 ± 0.017	0.164 ± 0.046	0.720 ± 0.014	0.279 ± 0.087	0.559 ± 0.026
Full-sequence Smoothing								
SDA	0.307 ± 0.099	0.573 ± 0.021	0.243 ± 0.079	0.639 ± 0.017	0.215 ± 0.077	0.667 ± 0.015	0.327 ± 0.109	0.551 ± 0.021
ForcingDAS-FS	**0.274** ± 0.080	**0.592** ± 0.021	**0.219** ± 0.062	**0.662** ± 0.017	**0.169** ± 0.046	**0.712** ± 0.014	**0.269** ± 0.076	**0.585** ± 0.022

Table 2: ERA5 SO-10% assimilation with 6 clean context frames (24 assimilated frames): latitude-weighted NRMSE (↓, z-score units) and anomaly correlation coefficient (ACC, ↑) per variable. Mean ± std over 16 held-out trajectories from the 2016 evaluation year; every method assimilates identical observations. ForcingDAS-Pyr uses a 10-frame lag. Best in *Filtering* and best in *Full-sequence Smoothing* per column in **bold**. Cold-start results are in Appendix S4.3.

	Z500		T850		U10		V10	
Method	NRMSE	ACC	NRMSE	ACC	NRMSE	ACC	NRMSE	ACC
Filtering								
3D-Var	0.067 ± 0.003	0.964 ± 0.004	0.114 ± 0.003	0.868 ± 0.020	0.478 ± 0.015	0.792 ± 0.012	0.536 ± 0.014	0.810 ± 0.012
FlowDAS	0.035 ± 0.001	0.990 ± 0.001	0.070 ± 0.001	0.948 ± 0.007	0.206 ± 0.005	0.958 ± 0.002	0.249 ± 0.005	0.956 ± 0.004
ForcingDAS-AR	**0.026** ± 0.000	**0.994** ± 0.001	**0.054** ± 0.001	**0.970** ± 0.004	**0.159** ± 0.003	**0.975** ± 0.001	**0.193** ± 0.004	**0.974** ± 0.002
Fix-lag Smoothing								
ForcingDAS-Pyr	0.023 ± 0.001	0.996 ± 0.000	0.048 ± 0.001	0.976 ± 0.003	0.148 ± 0.003	0.979 ± 0.001	0.181 ± 0.004	0.977 ± 0.002
Full-sequence Smoothing								
Tensor-Var	0.034 ± 0.001	0.990 ± 0.001	0.063 ± 0.002	0.957 ± 0.005	0.155 ± 0.003	0.976 ± 0.001	0.184 ± 0.003	0.976 ± 0.002
ForcingDAS-FS	**0.022** ± 0.001	**0.996** ± 0.000	**0.049** ± 0.001	**0.975** ± 0.003	**0.149** ± 0.003	**0.978** ± 0.001	**0.182** ± 0.004	**0.977** ± 0.002

to the current noise level (Appendix S3.2.5). We report NRMSE (mean ± std over events) and the Critical Success Index (CSI) averaged over the VIL thresholds $\{16, 74, 133, 160, 181, 219\}$ (CSI-mean). Following the SEVIR convention, CSI is pooled: hits, misses, and false alarms are summed over all assimilated frames of all 16 events before taking the ratio. Its ± is the std over 2,000 bootstrap resamples of the 16 events. CSI at each threshold, including the heavy-precipitation threshold 160, is in Table S5.

Results. Table 1 reports the with-context results. *Filtering:* ForcingDAS-AR improves on FlowDAS on every operator and metric, roughly halving the NRMSE (e.g., SO-10% 0.330 vs. 0.616), and it is better on every one of the 16 events for every operator (Wilcoxon $p = 3.1 \times 10^{-5}$). FlowDAS's error grows steadily over the 43 assimilated frames, whereas ForcingDAS stays flat (Fig. 1(b)). *Full-sequence smoothing:* Given the same clean context, ForcingDAS-FS improves on the learned smoother SDA on every operator, in NRMSE (e.g., SR2x 0.169 vs. 0.215) and in CSI-mean, and has the lower NRMSE on 15, 14, 16, and 16 of the 16 events for SO-10%, SO-20%, SR2x, and SR4x. The same ordering holds in the cold-start protocol (Appendix S4.2).

4.3 Global Atmospheric State Estimation (ERA5)

Dataset. ERA5 (Hersbach et al., 2020), ECMWF's fifth-generation global atmospheric reanalysis, is the de facto benchmark for learned DA (Huang et al., 2024; Andry et al., 2025). We use it as the ground-truth atmospheric state and run the same trained model as a filter (real-time forecasting) and as a smoother (retrospective reanalysis), which directly tests the unification claim. Data come from WeatherBench2 (Rasp et al., 2020) at $1.5°$ resolution with a 6-hourly cadence, on a 240×120 longitude–latitude grid. We train on 1979–2015 and evaluate on 16 trajectories of $K = 30$ frames

from the held-out 2016 evaluation year. The 16 windows are spread evenly across the year, one every 23 days, so that the reported std reflects seasonal variability. Four variables span upper-air dynamics, thermodynamics, and surface wind: **Z500** (geopotential at 500 hPa), **T850** (temperature at 850 hPa), and **U10** and **V10** (10 m zonal and meridional wind), stacked as a $4 \times 240 \times 120$ input with per-variable z-score normalization. This is a controlled testbed rather than an operational DA system: operational systems assimilate real instrument observations into 0.25° states with dozens of variables. See Appendix S3.3 for details and Appendix S1.1 for the broader landscape (Table S1).

Observation setup. We simulate sparse observations with one random pixel mask covering 10% of grid points, shared across the 4 channels and all frames, plus additive Gaussian noise of $\sigma_y = 0.05$ in z-score units (**SO-10%**). The observations are generated once and every method assimilates the identical values. In the main protocol the first 6 frames are clean context and the remaining 24 are assimilated and scored. The cold-start protocol (no clean context) is in Appendix S4.3. The equivalent noise in physical units is in Appendix S3.3.3.

Baselines and metrics. We compare against the classical filter **3D-Var**, which runs an independent variational analysis at each step with a spatially correlated background covariance and persistence cycling; the learned filter **FlowDAS** (Chen et al., 2025), which performs per-step posterior sampling with a learned stochastic-interpolant forecast model; and the learned smoother **Tensor-Var** (Yang et al., 2025), which runs 4D-Var in a learned feature space with linear feature dynamics and a transformer inverse-observation network. We retrained Tensor-Var on our variables and grid. Its inverse-observation network is trained on the same fixed mask that is used at evaluation, whereas ForcingDAS never sees the mask during training. Implementation details are in Appendix S5. Per variable we report the latitude-weighted NRMSE (RMSE in z-score space, following WeatherBench2 (Rasp et al., 2020)) and the latitude-weighted anomaly correlation coefficient (ACC) against the WeatherBench2 1990–2019 ERA5 climatology. Both are computed per trajectory over the assimilated frames and reported as mean $\pm$ std over the 16 trajectories; full definitions are in Appendix S3.3.4.

Results. Table 2 reports the with-context results. *Filtering:* ForcingDAS-AR has the lowest NRMSE and highest ACC among the filters on all four variables. Paired over the 16 trajectories, it is better than both FlowDAS and 3D-Var on every trajectory, every variable, and both metrics (Wilcoxon $p = 3.1 \times 10^{-5}$). The largest margins over FlowDAS are on the surface winds (U10 0.159 vs. 0.206; V10 0.193 vs. 0.249). 3D-Var's Gaussian background covariance interpolates the smooth synoptic-scale Z500 field reasonably well (ACC 0.964), but it cannot recover finer structure: its NRMSE is 2.5–3$\times$ that of ForcingDAS-AR on every variable. *Full-sequence smoothing:* ForcingDAS-FS improves on Tensor-Var on all four variables. The margins are large on Z500 (0.022 vs. 0.034) and T850 (0.049 vs. 0.063) and small but consistent on the surface winds: FS is better on 16/16 trajectories for U10 (0.149 vs. 0.155) and on 14/16 for V10 (0.182 vs. 0.184). The same trained model thus serves all three regimes without retraining. Fig. 3 shows ForcingDAS and Tensor-Var on a representative trajectory for all four variables. Its spectrum row shows a difference that NRMSE does not capture: ForcingDAS-FS follows the ground-truth spectrum up to the highest wavenumbers, whereas Tensor-Var carries excess small-scale power on Z500 and T850 and too little on the surface winds.

5 Conclusion and Limitations

We introduced **ForcingDAS**: a single joint-trajectory diffusion prior with per-frame noise levels, trained once, that supports filtering, fixed-lag smoothing, and full-sequence smoothing at inference through a single noise-scheduling matrix and no retraining. Causality-aware training matches the noise patterns the model sees at inference, and noise-level-aware observation guidance scales each frame's correction by its signal quality, together mitigating the long-horizon error accumulation that limits autoregressive learned DA. On 2D Navier–Stokes vorticity, SEVIR radar nowcasting, and ERA5 global atmospheric reanalysis, a single ForcingDAS model is competitive with or outperforms specialized learned baselines (FlowDAS, SDA, Tensor-Var) and the classical 3D-Var and EnKF pipelines, suggesting that the filtering–smoothing choice need not be baked in but can be exposed as a control over inference.

Limitations. The main constraints are causal-only smoothing (future observations reach past states only through backward gradients), compute that scales with the active window for joint-denoising regimes, and a coarse ERA5 setup relative to operational systems. See Appendix S6 for the full discussion.

AI USE STATEMENT

In this work, we used generative AI tools to edit the paper for grammar and readability and to assist with LaTeX formatting. We also used generative AI tools to organize code for our method and baseline experiment pipelines, assist with plotting scripts, and review author-written implementations. The core method and core adaptations of the baselines were implemented by the authors. We did not use generative AI tools to formulate mathematical claims, write proofs, propose or refine hypotheses, design experiments, generate or process data, or interpret results. All AI-assisted code and text were reviewed and verified by the authors. We take responsibility for the final content of this work, including text, claims, code, and other artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work develops a data assimilation method for state estimation in physical systems. All experiments use publicly available or simulated data: ERA5 reanalysis (Hersbach et al., 2020), distributed by ECMWF/Copernicus under an open-access license and accessed through WeatherBench2 (Rasp et al., 2020); the publicly released SEVIR dataset (Veillette et al., 2020); and 2D Navier–Stokes vorticity fields generated by a deterministic spectral solver (Appendix S3.1.1). The work involves no human subjects and no personal or sensitive data. A single trained model that supports filtering, fixed-lag smoothing, and full-sequence smoothing could simplify DA pipelines for nowcasting, weather forecasting, and reanalysis. As with any learned method, its calibration on unseen regimes should be verified before operational use, following the multi-stage validation that is standard for classical DA systems. The trained models are specialized to atmospheric, radar, and vorticity fields, and we do not foresee a direct path to misuse.

REPRODUCIBILITY STATEMENT

The ForcingDAS framework is described in §3, with pseudocode for training (Algorithm S3) and sampling (Algorithm S4) in Appendix S2. The network architecture and training setup are documented in Appendix S2.2 and Appendix S2.2. For each benchmark, the dataset, observation model, noise calibration, training compute, and sampling hyperparameters are given in Appendix S3.1 (Navier–Stokes), Appendix S3.2 (SEVIR), and Appendix S3.3 (ERA5); the ERA5 evaluation metrics are defined in Appendix S3.3.4, and the classical DA baselines are detailed in Appendix S5. All datasets are publicly available, and the Navier–Stokes data are generated by a solver whose configuration is given in Appendix S3.1.1. Anonymized code is provided at `https://anonymous.4open.science/r/ForcingDAS_ICLR-BA50/`, and we will release the code, trained checkpoints, and per-experiment configuration files upon publication.

REFERENCES

Ismail Alkhouri, Shijun Liang, Cheng-Han Huang, Jimmy Dai, Qing Qu, Saiprasad Ravishankar, and Rongrong Wang. SITCOM: Step-wise triple-consistent diffusion sampling for inverse problems. In *Forty-second International Conference on Machine Learning*, 2025. URL `https://openreview.net/forum?id=28Essvtvkw`.

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. *arXiv preprint arXiv:2504.18720*, 2025.

Mark Asch, Marc Bocquet, and Maëlle Nodet. *Data Assimilation: Methods, Algorithms, and Applications*. Number 11 in Fundamentals of Algorithms. Society for Industrial and Applied Mathematics, 2016. doi: 10.1137/1.9781611974546.

Arundhuti Banerjee, Ylona van Dinther, and Femke C. Vossepoel. On parameter bias in earthquake sequence models using data assimilation. *Nonlinear Processes in Geophysics*, 30(2):101–115, 2023. doi: 10.5194/npg-30-101-2023.

Jan-Hendrik Bastek, WaiChing Sun, and Dennis M Kochmann. Physics-informed diffusion models. In *International Conference on Learning Representations*, 2025.

Peter Bickel, Bo Li, and Thomas Bengtsson. Sharp failure rates for the bootstrap particle filter in high dimensions. In *Pushing the Limits of Contemporary Statistics: Contributions in Honor of Jayanta K. Ghosh*, volume 3, pp. 318–329. Institute of Mathematical Statistics, 2008.

Kevin Black, Michael Janner, Yilun Du, Ilya Kostrikov, and Sergey Levine. Training diffusion models with reinforcement learning. In *International Conference on Learning Representations*, 2024.

Marc Bocquet, H Elbern, H Eskes, M Hirtl, R Žabkar, GR Carmichael, J Flemming, A Inness, M Pagowski, JL Pérez Camaño, et al. Data assimilation in atmospheric chemistry models: current status and future prospects for coupled chemistry meteorology models. *Atmospheric Chemistry and Physics*, 15(10):5325–5358, 2015.

Steven L. Brunton and Bernd R. Noack. Closed-loop turbulence control: Progress and challenges. *Applied Mechanics Reviews*, 67(5):050801, 2015. doi: 10.1115/1.4031175.

Gerrit Burgers, Peter Jan van Leeuwen, and Geir Evensen. Analysis scheme in the ensemble Kalman filter. *Monthly Weather Review*, 126(6):1719–1724, 1998.

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

Boyuan Chen, Diego Martí Monsó, Yilun Du, Max Simchowitz, Russ Tedrake, and Vincent Sitzmann. Diffusion Forcing: Next-token prediction meets full-sequence diffusion. In *Advances in Neural Information Processing Systems*, volume 37, pp. 24081–24125, 2024a. doi: 10.52202/079017-0759.

Kun Chen, Lei Bai, Fenghua Ling, Peng Ye, Tao Chen, Hang Fan, Hao Chen, Yi Xiao, Kang Chen, Tao Han, Jing-Jia Luo, and Wanli Ouyang. Towards an end-to-end artificial intelligence driven global weather forecasting system. *arXiv preprint arXiv:2312.12462*, 2023.

Kun Chen, Peng Ye, Hao Chen, Kang Chen, Tao Han, Wanli Ouyang, Tao Chen, and Lei Bai. FNP: Fourier neural processes for arbitrary-resolution data assimilation. In *Advances in Neural Information Processing Systems*, volume 37, pp. 137847–137872, 2024b. doi: 10.52202/079017-4378.

Siyi Chen, Yixuan Jia, Qing Qu, He Sun, and Jeffrey Fessler. Flowdas: A stochastic interpolant-based framework for data assimilation. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 103061–103102. Curran Associates, Inc., 2025. doi: 10.52202/085713-3445. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/951690f8dae19c76394404540b7ca734-Paper-Conference.pdf.

Yifan Chen, Mark Goldstein, Mengjian Hua, Michael Samuel Albergo, Nicholas Matthew Boffi, and Eric Vanden-Eijnden. Probabilistic forecasting with stochastic interpolants and Föllmer processes. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 6728–6756. PMLR, 21–27 Jul 2024c.

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. URL https://openreview.net/forum?id=OnD9zGAGT0k.

Philippe Courtier, J.-N. Thépaut, and Anthony Hollingsworth. A strategy for operational implementation of 4D-Var, using an incremental approach. *Quarterly Journal of the Royal Meteorological Society*, 120(519):1367–1387, 1994. doi: 10.1002/qj.49712051912.

Philippe Courtier, Erik Andersson, William Heckley, Drasko Vasiljevic, Mats Hamrud, Anthony Hollingsworth, Florence Rabier, Mike Fisher, and Jean Pailleux. The ECMWF implementation of three-dimensional variational assimilation (3D-Var). I: Formulation. *Quarterly Journal of the Royal Meteorological Society*, 124(550):1783–1807, 1998. doi: 10.1002/qj.49712455002.

James A. Cummings. Operational multivariate ocean data assimilation. *Quarterly Journal of the Royal Meteorological Society*, 131(613):3583–3604, 2005. doi: 10.1256/qj.05.105.

James A. Cummings and Ole Martin Smedstad. Variational data assimilation for the global ocean. In Seon Ki Park and Liang Xu (eds.), *Data Assimilation for Atmospheric, Oceanic and Hydrologic Applications (Vol. II)*, pp. 303–343. Springer, Berlin, Heidelberg, 2013. doi: 10.1007/978-3-642-35088-7_13.

Karthik Duraisamy, Gianluca Iaccarino, and Heng Xiao. Turbulence modeling in the age of data. *Annual Review of Fluid Mechanics*, 51:357–377, 2019. doi: 10.1146/annurev-fluid-010518-040547.

Geir Evensen. The ensemble Kalman filter: theoretical formulation and practical implementation. *Ocean Dynamics*, 53:343–367, 2003. doi: 10.1007/s10236-003-0036-9.

Geir Evensen and Peter Jan van Leeuwen. An ensemble Kalman smoother for nonlinear dynamics. *Monthly Weather Review*, 128(6):1852–1867, 2000.

Hang Fan, Lei Bai, Ben Fei, Yi Xiao, Kun Chen, Yubao Liu, Yongquan Qu, Fenghua Ling, and Pierre Gentine. Physically consistent global atmospheric data assimilation with machine learning in latent space. *Science Advances*, 12(1):eaea4248, 2026.

Steven J. Fletcher. *Data Assimilation for the Geosciences: From Theory to Application*. Elsevier, 1 edition, 2017. ISBN 978-0-12-804444-5.

Anthony Frion, Lucas Drumetz, Mauro Dalla Mura, Guillaume Tochon, and Abdeldjalil Aissa El Bey. Neural koopman prior for data assimilation. *IEEE Transactions on Signal Processing*, 72: 4191–4206, 2024.

Anthony Frion, Lucas Drumetz, Mauro Dalla Mura, Guillaume Tochon, and Abdeldjalil AISSA EL BEY. Augmented invertible koopman autoencoder for long-term time series forecasting. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856. URL https://openreview.net/forum?id=o6ukhJLzMQ.

Gregory Gaspari and Stephen E Cohn. Construction of correlation functions in two and three dimensions. *Quarterly Journal of the Royal Meteorological Society*, 125(554):723–757, 1999.

Alan J. Geer, Katrin Lonitz, Peter Weston, Masahiro Kazumori, Kozo Okamoto, Yanqiu Zhu, Emily Huichun Liu, Andrew Collard, William Bell, Stefano Migliorini, Philippe Chambon, Nadia Fourrié, Min-Jeong Kim, Christina Köpken-Watts, and Christoph Schraff. All-sky satellite data assimilation at operational weather forecasting centres. *Quarterly Journal of the Royal Meteorological Society*, 144(713):1191–1217, 2018. doi: 10.1002/qj.3202.

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

Kuang Gong, Ciprian Catana, Jinyi Qi, and Quanzheng Li. Direct reconstruction of linear parametric images from dynamic PET using nonlocal deep image prior. *IEEE Transactions on Medical Imaging*, 41(3):680–689, 2022.

Neil J. Gordon, David J. Salmond, and Adrian F. M. Smith. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEE Proceedings F (Radar and Signal Processing)*, 140(2): 107–113, 1993. doi: 10.1049/ip-f-2.1993.0015.

Yutong He, Naoki Murata, Chieh-Hsin Lai, Yuhta Takida, Toshimitsu Uesaka, Dongjun Kim, WeiHsiang Liao, Yuki Mitsufuji, Zico Kolter, Ruslan Salakhutdinov, and Stefano Ermon. Manifold preserving guided diffusion. In *International Conference on Learning Representations*, 2024.

Hans Hersbach. Decomposition of the continuous ranked probability score for ensemble prediction systems. *Weather and Forecasting*, 15(5):559–570, 2000.

Hans Hersbach, Bill Bell, Paul Berrisford, Shoji Hirahara, András Horányi, Joaquín Muñoz-Sabater, Julien Nicolas, Carole Peubey, Raluca Radu, Dinand Schepers, et al. The ERA5 global reanalysis. *Quarterly Journal of the Royal Meteorological Society*, 146(730):1999–2049, 2020.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

Langwen Huang, Lukas Gianinazzi, Yuejiang Yu, Peter Dominik Dueben, and Torsten Hoefler. DiffDA: a diffusion model for weather-scale data assimilation. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 19798–19815. PMLR, 21–27 Jul 2024.

Xun Huang, Zhengqi Li, Guande He, Mingyuan Zhou, and Eli Shechtman. Self forcing: Bridging the train-test gap in autoregressive video diffusion. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=mSiN7i0BYH.

Yixuan Jia, Zongyu Li, Azadeh Akhavanallaf, Jeffrey A Fessler, and Yuni K Dewaraja. ^{90}Y SPECT scatter estimation and voxel dosimetry in radioembolization using a unified deep learning framework. *EJNMMI Physics*, 10(1):82, 2023.

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

William Lahoz, Boris Khattatov, and Richard Ménard (eds.). *Data Assimilation: Making Sense of Observations*. Springer, Berlin, Heidelberg, 2010. doi: 10.1007/978-3-540-74703-1.

Kody Law, Andrew Stuart, and Konstantinos Zygalakis. *Data Assimilation: A Mathematical Introduction*, volume 62 of *Texts in Applied Mathematics*. Springer Cham, 2015. doi: 10.1007/978-3-319-20325-6.

Xiang Li, Yixuan Jia, Xiao Li, Jeffrey A Fessler, Rongrong Wang, and Qing Qu. MCLR: Improving conditional modeling via inter-class likelihood-ratio maximization and unifying classifier-free guidance with alignment objectives. In *ICML 2026 Workshop on Foundations of Deep Generative Models: Understanding Memorization, Generalization, and Reasoning*, 2026a. URL https://openreview.net/forum?id=tma8ubTmy5.

Xiang Li, Soo Min Kwon, Shijun Liang, Ismail R Alkhouri, Saiprasad Ravishankar, and Qing Qu. DCDP: Decoupled data consistency via diffusion purification for inverse problems. In *2026 IEEE International Conference on Computational Photography (ICCP)*, 2026b.

Xiao Li, Zekai Zhang, Xiang Li, Siyi Chen, Zhihui Zhu, Peng Wang, and Qing Qu. Understanding representation dynamics of diffusion models via low-dimensional modeling. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, pp. 107365–107404. Curran Associates, Inc., 2025a.

Zongyu Li, Yixuan Jia, Xiaojian Xu, Jason Hu, Jeffrey A Fessler, and Yuni K Dewaraja. Shorter SPECT scans using self-supervised coordinate learning to synthesize skipped projection views. *EJNMMI Physics*, 12(1):47, 2025b.

Phillip Lippe, Bas Veeling, Paris Perdikaris, Richard Turner, and Johannes Brandstetter. PDE-Refiner: Achieving accurate long rollouts with neural PDE solvers. *Advances in Neural Information Processing Systems*, 36:67398–67433, 2023.

Rambod Mojgani, Ashesh Chattopadhyay, and Pedram Hassanzadeh. Interpretable structural model error discovery from sparse assimilation increments using spectral bias-reduced neural networks: A quasi-geostrophic turbulence test case. *Journal of Advances in Modeling Earth Systems*, 16(3): e2023MS004033, 2024.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4195–4205, 2023.

Shihan Qiu, Shaoyan Pan, Yikang Liu, Lin Zhao, Jian Xu, Qi Liu, Terrence Chen, Eric Z Chen, Xiao Chen, and Shanhui Sun. Spatiotemporal diffusion model with paired sampling for accelerated cardiac cine MRI. *arXiv preprint arXiv:2403.08758*, 2024.

Florence Rabier. Overview of global data assimilation developments in numerical weather-prediction centres. *Quarterly Journal of the Royal Meteorological Society*, 131(613):3215–3233, 2005. doi: 10.1256/qj.05.129.

Stephan Rasp, Peter D Dueben, Sebastian Scher, Jonathan A Weyn, Soukayna Mouatadid, and Nils Thuerey. WeatherBench: A benchmark data set for data-driven weather forecasting. *Journal of Advances in Modeling Earth Systems*, 12(11):e2020MS002203, 2020.

Rolf H. Reichle. Data assimilation methods in the Earth sciences. *Advances in Water Resources*, 31 (11):1411–1418, 2008. doi: 10.1016/j.advwatres.2008.01.001.

François Rozet and Gilles Louppe. Score-based data assimilation. In *Advances in Neural Information Processing Systems*, volume 36, pp. 40521–40541, 2023. doi: 10.52202/075280-1763.

David Ruhe, Jonathan Heek, Tim Salimans, and Emiel Hoogeboom. Rolling diffusion models. In *International Conference on Machine Learning*, 2024.

Ashutosh Singh, Ricardo Borsoi, Deniz Erdogmus, and Tales Imbiriba. Learning semilinear neural operators: A unified recursive framework for prediction and data assimilation. In *International Conference on Learning Representations*, volume 2024, pp. 44092–44109, 2024a.

Imraj RD Singh, Alexander Denker, Riccardo Barbano, Zeljko Kereta, Bangti Jin, K. Thielemans, Peter Maass, and Simon Robert Arridge. Score-based generative models for PET image reconstruction. *Machine Learning for Biomedical Imaging*, 2:547–585, 2024b. doi: 10.59275/j.melba.2024-5d51.

Chris Snyder, Thomas Bengtsson, and Mathias Morzfeld. Performance bounds for particle filters using the optimal proposal. *Monthly Weather Review*, 143(11):4750–4761, 2015.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, pp. 2256–2265. PMLR, 2015.

Bowen Song, Soo Min Kwon, Zecheng Zhang, Xinyu Hu, Qing Qu, and Liyue Shen. Solving inverse problems with latent diffusion models via hard data consistency. In *The Twelfth International Conference on Learning Representations*, 2024.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=St1giarCHLP.

Kiwhan Song, Boyuan Chen, Max Simchowitz, Yilun Du, Russ Tedrake, and Vincent Sitzmann. History-guided video diffusion. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=j8Vr3E3vhy.

Olivier Talagrand and Philippe Courtier. Variational assimilation of meteorological observations with the adjoint vorticity equation. I: Theory. *Quarterly Journal of the Royal Meteorological Society*, 113(478):1311–1328, 1987. doi: 10.1002/qj.49711347812.

Peter Jan van Leeuwen. Particle filtering in geophysical systems. *Monthly Weather Review*, 137(12): 4089–4114, 2009.

Mark Veillette, Siddharth Samsi, and Chris Mattioli. SEVIR : A storm event imagery dataset for deep learning applications in radar and satellite meteorology. *Advances in Neural Information Processing Systems*, 33:22009–22019, 2020.

Bram Wallace, Meihua Dang, Rafael Rafailov, Linqi Zhou, Aaron Lou, Senthil Purushwalkam, Stefano Ermon, Caiming Xiong, Shafiq Joty, and Nikhil Naik. Diffusion model alignment using direct preference optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8228–8238, 2024.

M. J. Werner, K. Ide, and D. Sornette. Earthquake forecasting based on data assimilation: sequential Monte Carlo methods for renewal point processes. *Nonlinear Processes in Geophysics*, 18(1): 49–70, 2011. doi: 10.5194/npg-18-49-2011.

Yanfei Xiang, Weixin Jin, Haiyu Dong, Jonathan Weyn, Mingliang Bai, Zuliang Fang, Pengcheng Zhao, Hongyu Sun, Kit Thambiratnam, Qi Zhang, and Xiaomeng Huang. ADAF: An artificial intelligence data assimilation framework for weather forecasting. *Journal of Advances in Modeling Earth Systems*, 17(9):e2024MS004839, 2025.

Yi Xiao, Lei Bai, Wei Xue, Hao Chen, Kun Chen, Kang Chen, Tao Han, and Wanli Ouyang. Towards a self-contained data-driven global weather forecasting framework. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 54255–54275. PMLR, 21–27 Jul 2024.

Xiaoze Xu, Xiuyu Sun, Wei Han, Xiaohui Zhong, Lei Chen, Zhiqiu Gao, and Hao Li. FuXi-DA: A generalized deep learning data assimilation framework for assimilating satellite observations. *npj Climate and Atmospheric Science*, 8(1):156, 2025.

Yiming Yang, Xiaoyuan Cheng, Daniel Giles, Sibo Cheng, Yi He, Xiao Xue, Boli Chen, and Yukun Hu. Tensor-Var: Efficient four-dimensional variational data assimilation. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 70649–70679. PMLR, 13–19 Jul 2025.

Zekai Zhang, Xiao Li, Xiang Li, Lianghe Shi, Meng Wu, Molei Tao, and Qing Qu. Generalization of diffusion models arises with a balanced representation space. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=57THeGgNAN.

Zhehao Zhang, Jiaming Liu, Deshan Yang, Ulugbek S. Kamilov, and Geoffrey D. Hugo. Deep learning-based motion compensation for four-dimensional cone-beam computed tomography (4D-CBCT) reconstruction. *Medical Physics*, 50(2):808–820, 2023.

Appendices

S1 Related Work **17**

S1.1 Classical and Learned Data Assimilation 17

S1.2 Diffusion Forcing and Per-Frame-Noise Diffusion 18

S1.3 Diffusion-Based Inverse-Problem Solvers 21

S2 Analysis and Implementation of ForcingDAS **22**

S2.1 Asymmetric Past–Future Coupling 22

S2.2 Network Architecture 22

S2.3 Other Design Choices 23

S2.4 Pseudocode 23

S3 Experimental Details and Additional Results **24**

S3.1 Navier–Stokes Vorticity 24

S3.2 SEVIR Radar Nowcasting 27

S3.3 ERA5 Global Atmospheric Reanalysis 29

S3.4 Effect of Causality-Aware Training on Probabilistic Forecasting 32

S4 Cold-Start (No-Context) Data Assimilation **36**

S4.1 Navier–Stokes Vorticity 36

S4.2 Precipitation Nowcasting (SEVIR-VIL) 36

S4.3 Global Atmospheric State Estimation (ERA5) 39

S5 Baseline Implementation Details **41**

S5.1 ERA5: Shared Setup 41

S5.2 3D-Var (Filtering) 41

S5.3 FlowDAS (Filtering) 42

S5.4 Tensor-Var (Smoothing) 42

S5.5 Ensemble Kalman Filter and Smoother Baselines for Navier–Stokes 42

S6 Limitations and Future Work **45**

S6.1 Limitations 45

S6.2 Future Work 45

S1 Related Work

This appendix positions ForcingDAS along three axes: (i) classical and learned data assimilation, the problem ForcingDAS solves (§S1.1); (ii) Diffusion Forcing and other per-frame-noise diffusion models, the family its trajectory prior belongs to (§S1.2); and (iii) diffusion-based inverse-problem solvers, which share its test-time guidance mechanism (§S1.3).

S1.1 Classical and Learned Data Assimilation

We organize related DA methods into classical (variational and ensemble) and learned (per-frame transitions and trajectory priors) approaches.

Classical variational methods (3D-Var, 4D-Var). 3D-Var (Courtier et al., 1998) computes a single-time analysis by minimizing $J(\boldsymbol{x}) = \frac{1}{2}(\boldsymbol{x}-\boldsymbol{x}_b)^\top \boldsymbol{B}^{-1}(\boldsymbol{x}-\boldsymbol{x}_b) + \frac{1}{2}(\boldsymbol{y}-\mathcal{A}\boldsymbol{x})^\top \boldsymbol{R}^{-1}(\boldsymbol{y}-\mathcal{A}\boldsymbol{x})$ with a fixed background $\boldsymbol{x}_b$ and background error covariance $\boldsymbol{B}$. 4D-Var (Talagrand & Courtier, 1987; Courtier et al., 1994) extends this to a window of length W, optimizing an initial state $\boldsymbol{x}_0$ that is propagated by a forward model Ψ and scoring observations against the resulting trajectory. Both formulations require a hand-crafted $\boldsymbol{B}$ and (for 4D-Var) an adjoint of Ψ. We use 3D-Var as the classical filtering baseline on ERA5; implementation details are in Appendix S5.

Classical ensemble methods (EnKF, EnKS). The Ensemble Kalman Filter (Evensen, 2003; Burgers et al., 1998) replaces the explicit $\boldsymbol{B}$ with a sample covariance from a forecast ensemble, which sidesteps the linear-Gaussian assumption of the original Kalman filter (Kalman, 1960) and works well when the forecast model is cheap to ensemble. The Ensemble Kalman Smoother (Evensen & van Leeuwen, 2000) is the smoothing analog, updating an entire window jointly given all observations in that window. Both methods require an explicit dynamical model at inference time; particle filters (Gordon et al., 1993; van Leeuwen, 2009) avoid the Gaussian assumption but are known to degenerate at high state dimension (Bickel et al., 2008; Snyder et al., 2015). We use the EnKF and two EnKS variants as physics-based baselines on Navier–Stokes (Appendix S5.5).

Learned filters: per-frame transition priors (FlowDAS). FlowDAS (Chen et al., 2025) learns a single-step stochastic-interpolant transition and runs DPS-style guidance frame by frame, giving an online filter. Because its prior is a per-step transition, it commits to each frame before the next observation arrives, so per-step errors accumulate over long horizons (§4.1); and when each observed frame is only a partial slice of a higher-dimensional latent state, a short, fixed window of recent frames does not determine the next one (§4.2, §4.3).

Learned smoothers: variational and trajectory priors (Tensor-Var, SDA). Tensor-Var (Yang et al., 2025) learns a linear latent dynamics model coupled with a transformer-based inverse-observation network, and solves a 4D-Var problem in the learned feature space in closed form over a short window; it is a windowed smoother and cannot run as a real-time filter. Score-Based Data Assimilation (SDA) (Rozet & Louppe, 2023) trains a frame-level diffusion score on full trajectories and combines it with observation guidance for offline reanalysis. The score is symmetric in time, which fits batch smoothing, but the same model cannot be used as a real-time filter: there is no causal structure to truncate. ForcingDAS keeps the trajectory prior of SDA but uses the per-frame noise structure of Diffusion Forcing to expose filtering and fixed-lag smoothing as well as full-sequence smoothing from a single trained model.

Learned latent and operator dynamics (Koopman priors, NODA). A related line of learned DA first learns the dynamics, or an operator for them, and then assimilates against that learned model. The neural Koopman prior (Frion et al., 2024) trains an autoencoder whose latent dynamics are linear and uses it as the dynamical prior in variational DA, either as a weak constraint on the full trajectory or as a strong constraint that optimizes only the initial latent state. AIKAE (Frion et al., 2025) augments an invertible Koopman autoencoder with a second, non-invertible encoder and uses the pre-trained model in the same strong-constraint variational scheme. NODA (Singh et al., 2024a) learns a state-space solution operator for semilinear PDEs and, building on nonlinear observer theory, alternates a learned prediction step with an observation-driven correction step, so that one recursive model serves both forecasting and assimilation. These methods learn latent or operator-based dynamics and assimilate

either by variational optimization over a window or by recursive prediction and correction, so the DA regime is fixed by the assimilation scheme. ForcingDAS instead learns a diffusion prior over whole trajectories with per-frame noise levels, assimilates by guided sampling, and selects filtering, fixed-lag smoothing or full-sequence smoothing at inference through the scheduling matrix.

Learned DA on ERA5. ERA5 has become the de facto evaluation benchmark for learned global atmospheric DA, and several recent systems run at full $0.25°$ resolution with 69–82 input channels (Huang et al., 2024; Andry et al., 2025). Table S1 summarizes the main entries, and we briefly characterize each below.

DiffDA (Huang et al., 2024) couples a GraphCast forecast backbone with a diffusion-based analysis head and assimilates simulated column observations through a single-step posterior. **FengWu-Adas** (Chen et al., 2023) integrates the FengWu forecast model with a learned analysis network that consumes a mix of simulated and real GDAS observations, and runs cyclically across many analysis steps. **FuXi-DA** (Xu et al., 2025) is a generalized deep-learning DA framework targeted at real satellite observations: it assimilates FY-4B radiances on top of the FuXi forecast model, demonstrating that learned operators can handle real instrument geometry rather than only synthetic grid masks. **Appa** (Andry et al., 2025) is a latent diffusion model for the global atmospheric state that supports zero-shot inverse problems with arbitrary observation operators at test time, and uses SDA-style score composition for posterior sampling. **L4DVar** (Fan et al., 2026) runs variational DA inside a learned latent space with deterministic encoder / decoder maps, retaining the variational structure of 4D-Var while operating on a much lower-dimensional state. **ADAF** (Xiang et al., 2025) is a regional, high-resolution AI DA framework that combines surface station and GOES-16 satellite observations on a CONUS grid. **FengWu-4DVar** (Xiao et al., 2024) couples the FengWu forecast model with a 4D-Var solver where the adjoint is obtained by automatic differentiation through the learned forward model, replacing the hand-coded operational adjoint.

Each listed method commits to a single DA regime, either (cyclic) filtering or window-based 4D-Var, and the same holds for Tensor-Var, which we use as the ERA5 smoothing baseline. None exposes the filtering–smoothing spectrum from a single trained model; ForcingDAS is, to our knowledge, the first to do so on ERA5. The systems in Table S1 operate at far larger scale than our study, which is a controlled comparison at $1.5°$ with four variables (§4.3).

Table S1: Landscape of learned DA methods on ERA5. Each method commits to a single DA regime; none exposes the filtering–smoothing spectrum. "Vars" counts total input channels (upper-air variables $\times$ pressure levels $+$ surface variables). "Obs" indicates the observation type used at test time. "Params" gives the approximate number of trainable parameters; "N/R" = not reported.

Method	Res.	Grid	Vars	Params	Obs	DA regime
DiffDA	$0.25°$	721×1440	82	$\geq$37M†	Simul. columns	Filtering
FengWu-Adas	$0.25°$	721×1440	69	N/R	Simul. + GDAS	Cyclic filtering
FuXi-DA	$0.25°$	721×1440	70	N/R	FY-4B sat.	Filtering
Appa	$0.25°$	721×1440	71	565M	Arbitrary (0-shot)	Filtering*
L4DVar	$1.4°$	128×256	69	N/R	GDAS real	Latent 4D-Var
ADAF	$0.05°$	512×1280	4	N/R	Sfc + GOES-16	Filtering (CONUS)
FengWu-4DVar	$0.25°$	721×1440	69	N/R	Simulated	4D-Var (AI adj.)
ForcingDAS (ours)	$1.5°$	240×120	4	$\sim$268M	Simul. sparse (10%)	**Filter $\leftrightarrow$ Smoother**

*Appa claims reanalysis capability via SDA score composition but does not study the filtering–smoothing spectrum.
†GraphCast backbone is 36.7M; the adapted diffusion variant is likely larger (not reported).

S1.2 Diffusion Forcing and Per-Frame-Noise Diffusion

ForcingDAS uses Diffusion Forcing (DF) (Chen et al., 2024a) as its trajectory prior. This subsection recaps DF, marks which components ForcingDAS inherits and which it introduces, and positions the per-frame-noise diffusion methods that have appeared since. We follow the notation of §2.2: $k = 1, \ldots, K$ is the frame index, $t = 1, \ldots, T$ is the diffusion-step index, t_k is the per-frame diffusion step, and $\boldsymbol{t} = (t_1, \ldots, t_K)$. The original DF paper instead uses T for sequence length (our K), K for the number of diffusion steps (our T), and k_t for t_k.

S1.2.1 PROBLEM FORMULATION

DF generalizes the standard diffusion forward process (Ho et al., 2020; Sohl-Dickstein et al., 2015) from a single noise variable to a per-frame collection of independent ones. Given a clean trajectory $\boldsymbol{x}_{1:K} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_K) \sim q$, draw independent noise levels $t_k \in \{0, \ldots, T\}$ and Gaussian noise $\boldsymbol{\epsilon}_k \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$, and form

$$\boldsymbol{x}_k^{(t_k)} = \sqrt{\bar{\alpha}_{t_k}}\, \boldsymbol{x}_k + \sqrt{1 - \bar{\alpha}_{t_k}}\, \boldsymbol{\epsilon}_k, \quad k = 1, \ldots, K, \tag{S1}$$

with $\{\bar{\alpha}_t\}_{t=0}^T$ the cumulative signal-retention coefficient of a chosen variance schedule. The two limits $t_k = 0$ (clean) and $t_k = T$ (white noise) recover an unmasked and a fully masked frame, so $\boldsymbol{t}$ acts as a continuous frame-level mask, the "masking along the noise axis" view of Chen et al. (2024a). Teacher-forced next-token prediction ($t_k \in \{0, T\}$) and full-sequence diffusion ($t_k = t$ shared across k) appear as special cases.

DF parameterizes the reverse process by a single joint denoiser $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{x}_{1:K}^{(\boldsymbol{t})}, \boldsymbol{t})$ that predicts the noise on every frame from the full noisy trajectory and the per-frame noise vector. The original DF instantiates this as a causal RNN with a hidden state $\boldsymbol{z}_k$ updated via $\boldsymbol{z}_k \sim p_{\boldsymbol{\theta}}(\boldsymbol{z}_k \mid \boldsymbol{z}_{k-1}, \boldsymbol{x}_k^{(t_k)}, t_k)$ (a Bayes-filter-style posterior that doubles as a conditional diffusion unit): per-frame features from a 2D U-Net (video) or a residual MLP (low-dimensional data) are passed through a GRU along the temporal axis. ForcingDAS uses a causal-attention transformer instead (Appendix S2.2). Both choices are length-invariant in K.

S1.2.2 TRAINING PROCEDURE

Standard DF training samples $t_k \overset{\text{i.i.d.}}{\sim} \mathrm{Uniform}\{0, \ldots, T\}$ across frames, corrupts $\boldsymbol{x}_{1:K}$ via Eq. (S1), and minimizes the per-frame score-matching loss

$$\mathcal{L}_{\mathrm{DF}}(\boldsymbol{\theta}) \;=\; \mathbb{E}_{\boldsymbol{x}_{1:K},\, \boldsymbol{t},\, \boldsymbol{\epsilon}_{1:K}} \sum_{k=1}^{K} \left\| \boldsymbol{\epsilon}_k - \boldsymbol{\epsilon}_{\boldsymbol{\theta}}\big(\boldsymbol{x}_{1:K}^{(\boldsymbol{t})}, \boldsymbol{t}\big)_k \right\|^2. \tag{S2}$$

The defining feature is the per-frame independence of $\boldsymbol{t}$: it exposes the network to all jointly-noised configurations rather than the shared step of full-sequence diffusion or the clean-past / fully-masked-future configuration of teacher forcing, in principle yielding robustness to any test-time scheduling matrix. The validity of this objective is captured by the central theoretical claim of Chen et al. (2024a):

Theorem 1 (Informal, restated from Chen et al. 2024a)**.** *Optimizing $\mathcal{L}_{\mathrm{DF}}$ with $\boldsymbol{t}$ drawn i.i.d. uniformly maximizes a reweighted ELBO on $\ln p_{\boldsymbol{\theta}}(\boldsymbol{x}_{1:K}^{(\boldsymbol{t})})$ averaged over $\boldsymbol{t}$, and under suitable conditions simultaneously lower-bounds the joint log-likelihood for* every fixed $\boldsymbol{t}$*.*

Theorem 1 licenses the use of a single trained DF network under arbitrary per-frame schedules and underpins the ForcingDAS claim that the same prior can serve as filter, fixed-lag smoother, or full-sequence smoother (§3.1). The training pseudocode in our notation is Algorithm S1; the ForcingDAS counterpart, which replaces the i.i.d. draw by the CAT mixture, is Algorithm S3.

Algorithm S1 Diffusion Forcing training (recap of Alg. 1 of Chen et al. 2024a).

Require: Dataset $\{\boldsymbol{x}_{1:K}\}$, total diffusion steps T, schedule $\{\bar{\alpha}_t\}$
1: **repeat**
2: Sample $\boldsymbol{x}_{1:K}$ from the dataset
3: $t_k \sim \mathrm{Uniform}\{0, \ldots, T\}$, $\boldsymbol{\epsilon}_k \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$, $k = 1, \ldots, K$ ▷ *i.i.d. across frames*
4: $\boldsymbol{x}_k^{(t_k)} \leftarrow \sqrt{\bar{\alpha}_{t_k}}\, \boldsymbol{x}_k + \sqrt{1 - \bar{\alpha}_{t_k}}\, \boldsymbol{\epsilon}_k$
5: Update $\boldsymbol{\theta}$ on $\sum_{k=1}^K \|\boldsymbol{\epsilon}_k - \boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{x}_{1:K}^{(\boldsymbol{t})}, \boldsymbol{t})_k\|^2$
6: **until** converged

S1.2.3 SAMPLING PROCEDURE

DF sampling is controlled by a *scheduling matrix* $\boldsymbol{S} \in \{0, \ldots, T\}^{K \times L}$ (the matrix $\mathcal{K} \in [K]^{M \times T}$ of Chen et al. (2024a), with rows and columns transposed): $S_{k,\ell}$ is the diffusion step of frame k

at the ℓ-th of L reverse iterations, with $S_{k,0} = T$ and $S_{k,L} = 0$ for every k. Generation initializes $\boldsymbol{x}_k^{(T)} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$ and iterates $\ell = 0, \dots, L-1$; at each ℓ the active set $\mathcal{K}_{\text{active}} = \{k : S_{k,\ell+1} < S_{k,\ell}\}$ collects the frames that descend, and a single DDIM (Song et al., 2021) step takes each $k \in \mathcal{K}_{\text{active}}$ from $t_k = S_{k,\ell}$ to $t'_k = S_{k,\ell+1}$:

$$\boldsymbol{x}_k^{(t'_k)} = \sqrt{\bar{\alpha}_{t'_k}}\, \hat{\boldsymbol{x}}_k^{(0)} + \sqrt{1 - \bar{\alpha}_{t'_k} - \sigma_{t_k}^2}\, \boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{x}_{1:K}^{(\boldsymbol{t})}, \boldsymbol{t})_k + \sigma_{t_k}\, \boldsymbol{z}_k, \quad \boldsymbol{z}_k \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}), \tag{S3}$$

with Tweedie estimate $\hat{\boldsymbol{x}}_k^{(0)} = (\boldsymbol{x}_k^{(t_k)} - \sqrt{1 - \bar{\alpha}_{t_k}}\, \boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\cdot)_k)/\sqrt{\bar{\alpha}_{t_k}}$; non-active frames are left unchanged. Chen et al. (2024a) highlights three canonical schedules (synchronous full-sequence, autoregressive next-token, and *pyramid*, with the far future kept noisier than the near future); ForcingDAS recasts all three as DA regimes (full-sequence smoothing, filtering, and fixed-lag smoothing) and embeds them into the one-parameter family $\boldsymbol{S}^{(u)}$ of Eq. (7).

DF further supports test-time guidance: any differentiable cost $c(\boldsymbol{x}_{1:K}^{(\boldsymbol{t})})$ contributes a gradient added to Eq. (S3). Because the causal denoiser makes a future frame's $\hat{\boldsymbol{x}}_{k'}^{(0)}$ a differentiable function of all $\boldsymbol{x}_k^{(t_k)}$ with $k \le k'$, the gradient flows backwards in time and refines past frames. ForcingDAS reuses this "long-horizon guidance" mechanism for observation guidance (§3.1, Appendix S2.1). The Monte Carlo Guidance variant of Chen et al. (2024a), which averages gradients over multiple future roll-outs, is not needed for DA and not used. The sampling pseudocode is Algorithm S2; the ForcingDAS counterpart with observation guidance is Algorithm S4.

Algorithm S2 Diffusion Forcing sampling (recap of Alg. 2 of Chen et al. 2024a).

Require: $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}$, scheduling matrix $\boldsymbol{S} \in \{0, \dots, T\}^{K \times L}$, optional cost $c(\cdot)$, scale $\zeta \ge 0$

1: $\boldsymbol{x}_k^{(T)} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$, $k = 1, \dots, K$
2: **for** $\ell = 0, \dots, L-1$ **do**
3: $\quad t_k \leftarrow S_{k,\ell}$, $t'_k \leftarrow S_{k,\ell+1}$; $\mathcal{K}_{\text{active}} \leftarrow \{k : t'_k < t_k\}$
4: $\quad$ Apply Eq. (S3) to each $k \in \mathcal{K}_{\text{active}}$
5: **end for**
6: **Return** $\boldsymbol{x}_{1:K}^{(0)}$

S1.2.4 What ForcingDAS inherits and what it introduces

ForcingDAS adopts from DF, without claim of novelty, the per-frame Gaussian forward process (Eq. (S1)), the joint denoiser conditioned on the noise-level vector $\boldsymbol{t}$, causal temporal dependency across frames, the scheduling-matrix abstraction, and the backward flow of guidance gradients through the causal denoiser. It introduces: (i) the reformulation of per-frame diffusion sampling as a DA solver through observation guidance with noise-level-aware reweighting (Eq. (5)) and windowed aggregation over frames active at staggered diffusion steps; (ii) the one-parameter family $\boldsymbol{S}^{(u)}$ that maps the uncertainty scale u onto filtering, fixed-lag smoothing, and full-sequence smoothing, with DF's canonical schedules as special cases; (iii) causality-aware training (Eq. (4)), which biases the training noise distribution toward the causally monotone configurations used at DA inference; and (iv) the empirical validation of a single trained per-frame-diffusion prior across the DA spectrum against classical and learned DA baselines.

S1.2.5 Other per-frame-noise diffusion methods

Rolling Diffusion (Ruhe et al., 2024). Rolling Diffusion sweeps a window across a sequence with frames at monotonically increasing noise levels, so generation proceeds frame by frame in a streaming fashion. The active-window structure resembles ForcingDAS-Pyr ($0 < u < T$), but the design target is unconditional video generation rather than data assimilation: there is no observation operator and no notion of filtering or smoothing.

Diffusion Forcing Transformer (DFoT) (Song et al., 2025). DFoT replaces the original DF U-Net with a transformer backbone and uses the per-frame noise levels to expose a "history guidance" knob at sampling time, which trades off how strongly clean past frames condition future ones. We share the per-frame noise structure but adapt it for DA rather than long-video generation. ForcingDAS

could in principle use the DFoT backbone in place of our DiT-3D; the filtering–smoothing spectrum and the observation guidance are orthogonal to the choice of backbone.

Self-Forcing (Huang et al., 2025). Self-Forcing trains an autoregressive video diffusion model on its own roll-outs to close the train–test exposure gap. It targets a single inference pattern (autoregressive, frame by frame) and does not expose a continuous schedule. CAT (§3.1) addresses a related but different gap: standard DF samples per-frame noise levels i.i.d., while every DA scheduling matrix imposes a causally monotone pattern at inference; CAT biases training toward the monotone configurations rather than toward roll-out trajectories.

None of these methods has been formulated as a data assimilation solver: there is no observation operator $\mathcal{A}$, no observation guidance, and no filtering / fixed-lag / smoothing distinction.

S1.3 Diffusion-Based Inverse-Problem Solvers

A broader line of work uses pretrained diffusion priors with test-time gradient guidance to solve inverse problems (Chung et al., 2023; Song et al., 2024; Li et al., 2026b; Alkhouri et al., 2025). These methods target single-image restoration (deblurring, inpainting, super-resolution) and do not handle a sequence of partial observations or distinguish filtering from smoothing. ForcingDAS uses the same gradient-guidance idea but extends it to trajectory-level data assimilation; the noise-level-aware reweighting of Eq. (5) arises from the per-frame diffusion-step staircase that single-image solvers do not encounter.

S2 Analysis and Implementation of ForcingDAS

This appendix collects the analysis of how observations couple frames under causal attention (§S2.1) and the implementation details summarized in §3.2: the network backbone (§S2.2), further design choices (§S2.3), and pseudocode for training and inference (§S2.4).

S2.1 Asymmetric Past–Future Coupling

Causal temporal attention introduces a directional asymmetry in how observations couple the trajectory. A frame k directly attends to every past frame $k' < k$: once past frames have been corrected by their observations, those corrections propagate into frame k through the *forward pass*, a strong, direct pathway. Future observations, by contrast, reach past frames only through the *backward gradient* of the measurement loss: since the causal Jacobian $\partial \hat{\boldsymbol{x}}_{0,k'} / \partial \boldsymbol{x}_k^{(t_k)}$ is non-zero for $k \le k'$ and zero otherwise (Eq. (S4) below), guidance at a future frame k' does produce a gradient on frame k, but an *indirect* one that passes through the attention and nonlinear layers between the two frames. Consequently, even in its most smoothing-like mode ($u = 0$), ForcingDAS lets future observations influence past states only through this backward gradient, whereas a model with bidirectional attention would also carry them through the forward pass. We give the formal chain-rule decomposition below (Remark 1).

Forward direction: past → future. When frame k is being denoised, causal attention lets it directly read information from all past frames $k' < k$. If past frames have already been corrected by their observations, those corrections propagate to frame k through the forward pass, a direct information pathway. This is the *filtering* channel.

Backward direction: future → past. When observation guidance is applied, the measurement loss at a future frame k' ($k' > k$) generates gradients $\partial \mathcal{L}(\boldsymbol{y}_{k'}) / \partial \boldsymbol{x}_k^{(t_k)}$ that flow backward through the network. Because causal attention makes $\hat{\boldsymbol{x}}_{0,k'}$ a differentiable function of all inputs $\boldsymbol{x}_k$ with $k \le k'$, these backward gradients are non-zero but *indirect*: they pass through softmax Jacobians and nonlinearities. This is the *smoothing* channel.

Remark 1 (Cross-frame gradient structure). Let $g_k(\boldsymbol{x}_{1:K}^{(\boldsymbol{t})}, \boldsymbol{t}) \triangleq \hat{\boldsymbol{x}}_{0,k}$ denote the Tweedie denoised estimate for frame k, viewed as a function of the full noisy input. Under causal temporal attention, the Jacobian satisfies

$$\frac{\partial\, g_{k'}}{\partial\, \boldsymbol{x}_k^{(t_k)}} \begin{cases} \neq \mathbf{0} & \text{if } k \le k', \\ = \mathbf{0} & \text{if } k > k'. \end{cases} \tag{S4}$$

Consequently, the observation-guidance gradient of frame k''s measurement loss on frame k decomposes via the chain rule as

$$\nabla_{\boldsymbol{x}_k^{(t_k)}} \|\boldsymbol{y}_{k'} - \mathcal{A}(g_{k'})\|^2 = \underbrace{-2\, \boldsymbol{J}_{\mathcal{A}}^{\top}(\boldsymbol{y}_{k'} - \mathcal{A}(g_{k'}))}_{\text{observation residual signal}} \cdot \underbrace{\frac{\partial\, g_{k'}}{\partial\, \boldsymbol{x}_k^{(t_k)}}}_{\text{cross-frame Jacobian}}, \tag{S5}$$

where $\boldsymbol{J}_{\mathcal{A}} = \partial \mathcal{A} / \partial \hat{\boldsymbol{x}}_{0,k'}$. For $k' > k$, this gradient is the *backward smoothing signal*: non-zero by Eq. (S4), and routed through the causal path from frame k to frame k' in the network.

S2.2 Network Architecture

The denoising network $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}$ is a 3D Diffusion Transformer (DiT-3D) (Peebles & Xie, 2023; Chen et al., 2024a) adapted to per-frame noise levels. Each frame is patchified into a sequence of spatial tokens; the per-frame diffusion step t_k is mapped to a sinusoidal time embedding and injected into the corresponding frame's tokens via adaptive layernorm. Spatial self-attention runs over the full image within each frame, while temporal attention is restricted to the causal mask: frame k attends only to frames $k' \le k$. The same backbone is used across all three benchmarks. We use a transformer rather than the convolutional / recurrent backbone of the original DF because the scientific fields targeted here (2D Navier–Stokes vorticity, SEVIR radar, ERA5 weather) have broadband spatial spectra and rapidly evolving structure, for which global attention is empirically a stronger inductive bias.

The training setup is otherwise inherited from DF: the per-frame Gaussian forward process of Eq. (S1), $\boldsymbol{\epsilon}$-parameterization, a cosine variance schedule, $T = 1000$ diffusion steps, AdamW optimization with mixed-precision training, and 100-step DDIM sampling. Per-dataset training data, compute, and sampling hyperparameters are listed in Appendix S3.

S2.3 Other Design Choices

Frame stacking. Following DF, the implementation supports stacking F consecutive frames into a single super-frame with $C \cdot F$ channels, which reduces the temporal dimension fed to the network by a factor F and assigns one noise level per super-frame. All experiments in this paper use $F = 1$, i.e., one noise level per frame.

Noise clipping. For z-score-normalized weather data with a wide dynamic range (ERA5), we clip the standard Gaussian noise to $[-15, 15]$ rather than the default $[-6, 6]$, which would otherwise truncate the heavy tails of the multi-variable normalized distribution.

Sliding window at inference. Frames are generated in chunks: one frame at a time for the autoregressive schedule, and the whole remaining horizon at once for the pyramid and full-sequence schedules. When the trajectory is longer than the number of frames K_{train} the network was trained on, only the most recent K_{train} frames up to the end of the current chunk are fed to the network; earlier frames influence the chunk only through the already-generated frames inside this window.

S2.4 Pseudocode

Algorithm S3 gives the training procedure (sample a CAT-mixture diffusion-step vector, corrupt, score-match), and Algorithm S4 gives the inference procedure (read the scheduling matrix, identify active frames, compute the reweighted observation gradient, apply the DDIM step with the guidance correction). Compared with the DF recaps in Algorithms S1–S2, the new steps are the sorting and context clamping in Algorithm S3 (lines 4–5) and the observation guidance in Algorithm S4 (lines 6a–6c).

Algorithm S3 ForcingDAS Training

Require: Dataset of trajectories $\{\boldsymbol{x}_{1:K}\}$, causal ratio ρ, context clean ratio ρ_c

1: **repeat**
2: Sample trajectory $\boldsymbol{x}_{1:K}$ from dataset
3: Sample diffusion steps $t_k \sim \mathrm{Uniform}\{1, \dots, T\}$ for $k = 1, \dots, K$
4: With probability ρ: sort $\boldsymbol{t} \leftarrow \mathrm{sort}(t_1, \dots, t_K)$ ▷ *Causally-aligned noise*
5: With probability ρ_c: set $t_k \leftarrow 0$ for $k = 1, \dots, C$ where $C \sim \mathrm{Uniform}\{1, \dots, C_{\max}\}$
6: Sample $\boldsymbol{\epsilon}_k \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$ for $k = 1, \dots, K$
7: Corrupt: $\boldsymbol{x}_k^{(t_k)} \leftarrow \sqrt{\bar{\alpha}_{t_k}}\, \boldsymbol{x}_k + \sqrt{1 - \bar{\alpha}_{t_k}}\, \boldsymbol{\epsilon}_k$
8: Update $\boldsymbol{\theta}$ to minimize $\sum_{k=1}^{K} \lambda(t_k) \left\| \boldsymbol{\epsilon}_k - \boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{x}_{1:K}^{(\boldsymbol{t})}, \boldsymbol{t})_k \right\|^2$
9: **until** converged

Algorithm S4 ForcingDAS Sampling (Inference)

Require: Observations $\{\boldsymbol{y}_k\}$, forward operator $\mathcal{A}$, scheduling matrix $\boldsymbol{S}^{(u)}$, trained model $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}$, guidance scale ζ, noise parameter γ

1: Initialize $\boldsymbol{x}_k^{(T)} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$ for $k = 1, \ldots, K$
2: **for** $\ell = 0, 1, \ldots, L-1$ **do**
3: Read diffusion steps: $t_k \leftarrow S_{k,\ell}^{(u)}$, $t'_k \leftarrow S_{k,\ell+1}^{(u)}$ for $k = 1, \ldots, K$
4: Identify active frames: $\mathcal{K}_{\text{active}} \leftarrow \{k : t_k > t'_k\}$
5: Compute $\hat{\boldsymbol{x}}_{0,k} \leftarrow g_k(\boldsymbol{x}_{1:K}^{(\boldsymbol{t})}, \boldsymbol{t})$ via the Tweedie estimate ▷ *Prior step*
6a: Compute per-frame weights: $w(t_k) \leftarrow \left(\sigma_y^2 + \gamma \, \frac{1-\bar{\alpha}_{t_k}}{\bar{\alpha}_{t_k}}\right)^{-1}$ ▷ *Obs. guidance*
6b: Aggregate loss: $\mathcal{L}_{\text{obs}} \leftarrow \left(\sum_{k \in \mathcal{K}_{\text{active}}} w(t_k) \, \|\boldsymbol{y}_k - \mathcal{A}(\hat{\boldsymbol{x}}_{0,k})\|^2\right)^{1/2}$ ▷ *Obs. guidance*
6c: Compute gradient: $\boldsymbol{g} \leftarrow \nabla_{\boldsymbol{x}_{1:K}^{(\boldsymbol{t})}} \mathcal{L}_{\text{obs}}$ ▷ *Obs. guidance*
7: For each $k \in \mathcal{K}_{\text{active}}$: ▷ *DDIM update*
8: $\boldsymbol{x}_k^{(t'_k)} \leftarrow \sqrt{\bar{\alpha}_{t'_k}} \, \hat{\boldsymbol{x}}_{0,k} + \sqrt{1 - \bar{\alpha}_{t'_k} - \sigma_{t_k}^2} \, \boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\cdot)_k + \sigma_{t_k} \boldsymbol{z}_k - \zeta \, \boldsymbol{g}_k$
9: **end for**
10: **return** Estimated trajectory $\boldsymbol{x}_{1:K}^{(0)}$

S3 Experimental Details and Additional Results

This appendix collects, for each of the three benchmarks of §4: (i) the dataset and its splits; (ii) the observation model and the conversion between the data-space noise level and its physical-units counterpart; (iii) training compute and the sampling hyperparameters behind the reported numbers; (iv) baseline configurations; and (v) additional results and visualizations. It closes with an ablation of causality-aware training on probabilistic forecasting (§S3.4). Cold-start results are in Appendix S4.

Convention: data space vs. raw data space. Throughout the main text, we report the additive observation noise as a Gaussian of standard deviation $\sigma_y = 0.05$ in the dataset's *data space*, i.e. the space in which both the diffusion model and the observation operator $\mathcal{A}$ act at training and inference. This is the space produced by the dataset loader: $[0, 1]$-rescaled vorticity for NS (§S3.1), $[0, 1]$-scaled VIL for SEVIR (§S3.2), and per-channel z-score-normalized fields for ERA5 (§S3.3). For each system, the dataset normalization is an invertible affine map $x^{\text{data}} = (x^{\text{raw}} - a)/b$, where b is a positive scalar (NS, SEVIR) or a per-channel scale (σ_c^{clim} for ERA5). Because every observation operator $\mathcal{A}$ we evaluate (sparse pixel masking; bilinear down-sampling) is *linear*, applying $\mathcal{A}$ in data space and adding noise of std σ_y is exactly equivalent to applying the same $\mathcal{A}$ (same mask / same kernel) directly to the raw field and adding Gaussian noise of standard deviation

$$\sigma_y^{\text{raw}} \;=\; b \cdot \sigma_y^{\text{data}}, \qquad b \;=\; \begin{cases} x_{\max} - x_{\min} & \text{(min–max scaling, NS \& SEVIR),} \\ \sigma_c^{\text{clim}} & \text{(per-channel } z\text{-score, ERA5).} \end{cases} \tag{S6}$$

That is, at every observed grid point i (and channel c for ERA5), $y_i^{\text{data}} = x_i^{\text{data}} + \sigma_y^{\text{data}} \varepsilon_i = (x_i^{\text{raw}} - a)/b + \sigma_y^{\text{data}} \varepsilon_i$, so that $b\, y_i^{\text{data}} + a = x_i^{\text{raw}} + (b\, \sigma_y^{\text{data}})\, \varepsilon_i$, with the *same* Gaussian sample ε_i. Eq. (S6) is therefore an exact restatement, not an approximation, and the per-benchmark subsections below tabulate the resulting σ_y^{raw} in physical units.

S3.1 Navier–Stokes Vorticity

S3.1.1 Dataset and physical setting

We use 2D incompressible Navier–Stokes vorticity trajectories generated from the stochastic vorticity equation on the torus $\mathbb{T}^2 = [0, 2\pi]^2$ (Chen et al., 2024c; Rozet & Louppe, 2023):

$$\mathrm{d}\omega + (\boldsymbol{v} \cdot \nabla \omega)\, \mathrm{d}t = \nu \, \Delta \omega \, \mathrm{d}t - \alpha \, \omega \, \mathrm{d}t + \varepsilon \, \mathrm{d}\eta, \tag{S7}$$

where $\boldsymbol{v} = \nabla^{\perp} \psi = (-\partial_y \psi, \, \partial_x \psi)$ is the velocity field recovered from the stream function ψ satisfying $-\Delta \psi = \omega$. The parameters are viscosity $\nu = 10^{-3}$, linear drag $\alpha = 0.1$, and noise

amplitude $\varepsilon = 1.0$. The stochastic forcing $\mathrm{d}\eta$ is white-in-time and acts on eight fixed sinusoidal modes driven by independent Wiener processes $\{W_i(t)\}_{i=1}^{8}$:

$$\begin{aligned}\eta(t,x,y) = {} & W_1(t)\sin(6x) + W_2(t)\cos(7x) + W_3(t)\sin\big(5(x{+}y)\big) + W_4(t)\cos\big(8(x{+}y)\big) \\ & + W_5(t)\cos(6x) + W_6(t)\sin(7x) + W_7(t)\cos\big(5(x{+}y)\big) + W_8(t)\sin\big(8(x{+}y)\big).\end{aligned} \tag{S8}$$

The data are generated using a pseudo-spectral solver with Euler–Maruyama time stepping on a 256×256 grid at $\Delta t = 10^{-4}$, with snapshots stored every 0.5 time units. The fields are then bilinearly downsampled to 128×128 for training and evaluation, and every second snapshot is kept, so consecutive frames are 1.0 time units apart. The trajectories are split 950/50 into training and held-out evaluation trajectories. We evaluate on 16 of the 50 held-out trajectories, which are not used for training or model selection and are disjoint from the 950 training trajectories of the ForcingDAS, FlowDAS, and SDA checkpoints alike. Four further held-out trajectories, disjoint from these 16, form a tuning set, used to select the ForcingDAS sampling hyperparameters (Appendix S3.1.4). We use $K = 50$ frames per trajectory for the with-context protocol (10 clean context frames, 40 assimilated) and $K = 30$ for the cold-start protocol (Appendix S4.1). Because the underlying PDE is fully specified, this experiment also admits strong physics-based DA references (the Ensemble Kalman Filter and Smoother), whose implementation is described in Appendix S5.5.

The training data have empirical extrema $[\,x_{\min},\, x_{\max}\,] = [\,-19.16,\, +17.42\,]$ and standard deviation $\sigma^{\text{raw}} = 3.07$ (rad/s), giving a min–max width of $b = x_{\max} - x_{\min} = 36.58$. The dataset loader maps raw vorticity into the data space by min–max rescaling to $[0, 1]$, $x^{\text{data}} = (x^{\text{raw}} - x_{\min})/b$; this is the input the diffusion model receives, and the space in which $\mathcal{A}$ is evaluated at test time.

S3.1.2 Observation model and noise calibration

Operator. For NS we evaluate the 5% sparse observation operator (cf. Setup in §4.1): a random binary pixel mask retaining 5% of grid points. The mask is a fixed (seeded) sample, shared across frames and trajectories, that observes 795 of the 16,384 pixels.

Noise. The clean operator output is corrupted by additive Gaussian noise of std $\sigma_y^{\text{data}} = 0.05$ (the value we report in the main text), i.e. $\boldsymbol{y}_k = \mathcal{A}(\boldsymbol{x}_k^{\text{data}}) + \boldsymbol{\varepsilon}_k$, $\boldsymbol{\varepsilon}_k \sim \mathcal{N}(\mathbf{0}, 0.05^2 \boldsymbol{I})$, in the $[0, 1]$-rescaled vorticity space.

Equivalent noise in raw vorticity units. By Eq. (S6) with $b = 36.58$, this is exactly equivalent to applying the same operator to the *raw* vorticity field and adding Gaussian noise of standard deviation

$$\sigma_y^{\text{raw}} \;=\; 0.05 \times 36.58 \;=\; 1.829 \;\text{(rad/s)} \;\;\approx\;\; 0.60 \times \sigma^{\text{raw}}. \tag{S9}$$

That is, the observation noise is roughly 60% of the climatological standard deviation of the field, a moderate noise level relative to the dynamic range of the vorticity.

Shared observations. The observations are drawn once per trajectory from a seeded generator and stored, and every method (ForcingDAS, FlowDAS, SDA, EnKF, EnKS) assimilates these identical values; the evaluation script checks this and refuses to score a run otherwise. The baselines operate in different units (FlowDAS in raw vorticity divided by 3.07; SDA, EnKF, and EnKS in raw vorticity), so the stored observations are converted into each method's units by the exact affine map of Eq. (S6). Every method therefore sees the same noise level, $\sigma_y^{\text{raw}} = 1.829$. Since all methods see the same observations on the same trajectories, per-trajectory differences between two methods are exactly paired, and we test them with the Wilcoxon signed-rank test.

Metrics. NRMSE is computed per frame as $\|\hat{\boldsymbol{x}}_k - \boldsymbol{x}_k\|_F / \|\boldsymbol{x}_k\|_F$ in the $[0, 1]$ data space, averaged over the scored frames of a trajectory, and then reported as mean $\pm$ std over the 16 trajectories. The spectrum errors are the mean per-bin relative errors of the radially averaged kinetic-energy spectrum (computed from the vorticity through the stream function) over mid-k $[8, 32)$ and over all 64 bins $[0.5, 64)$. Only assimilated frames are scored, since the clean context frames are copies of the truth. Because the min–max map adds a constant offset, NRMSE in the $[0, 1]$ space is smaller than NRMSE of the raw vorticity by a fixed factor (≈ 11.9 on these trajectories). The factor is the same for every method, so rankings are unaffected.

S3.1.3 TRAINING COMPUTE

We train the ForcingDAS DiT-B backbone on the NS training set of shape $(950, 50, 128, 128)$ (trajectories $\times$ frames per trajectory $\times$ height $\times$ width) in float32 across 4 NVIDIA A800 GPUs for 500 epochs, with per-GPU batch size 8 (total batch size 32). Total training time is approximately 24 hours.

S3.1.4 SAMPLING HYPERPARAMETERS

ForcingDAS sampling on NS uses the DiT-B backbone with 100 DDIM steps, per-frame variance reweighting (Eq. 5) with parameter γ, and guidance scale ζ. Table S2 reports the values behind every number in Table S3 and Table S13. ForcingDAS-Pyr uses a 10-frame lag ($u = 10$; with $T = 100$ DDIM steps the active window is $\min(K, \lceil T/u \rceil)$ frames). For all three schedules, (γ, ζ) were selected on the 4 held-out tuning trajectories, which are disjoint from the 16 evaluation trajectories.

Table S2: NS sampling hyperparameters (DiT-B backbone, 100 DDIM steps, SO-5%, $\sigma_y^{\text{data}} = 0.05$). AR = autoregressive (filtering), Pyr = pyramid (fixed-lag smoothing, $u = 10$), FS = full-sequence (smoothing). "Lag" is the number of frames that are denoised jointly. All (γ, ζ) are selected on the 4 held-out tuning trajectories.

Protocol	K (frames)	Schedule	Lag	γ	ζ
With context (10 clean)	50	AR	1	0.01	2.0
	50	Pyr	10	0.01	4.0
	50	FS	40	0.005	12.0
Cold start (0 clean)	30	AR	1	0.05	4.0
	30	Pyr	10	0.05	8.0
	30	FS	30	0.01	12.0

S3.1.5 BASELINE CONFIGURATIONS

FlowDAS (Chen et al., 2025) uses the original public implementation and its public NS checkpoint, whose training data lie within the 950 training trajectories and do not overlap the evaluation set. It uses a 10-frame conditioning window, 500 Euler–Maruyama steps per assimilated frame, 25 Monte Carlo samples for the guidance estimate, and guidance scale 1.0, the value selected in the original FlowDAS paper. With context, the window is seeded with the 10 clean frames; in the cold start, the released code has no such mode, so we seed the window with standard Gaussian noise. **SDA** (Rozet & Louppe, 2023) uses a score network trained on the same 950 trajectories with a Markov blanket of window 5, sampled with 256 steps, 1 Langevin correction, $\tau = 0.5$, and $\gamma = 0.01$, selected on the 4 held-out tuning trajectories. SDA has no clean-context mode. Passing the context frames as fully observed frames makes its unnormalized Gaussian likelihood dominated by those frames, and sampling diverged at every likelihood scale we tried. We therefore give SDA the context by replacement conditioning: after every predictor and corrector update, the context frames are overwritten with the ground truth diffused to the current noise level, while the likelihood term is unchanged. The returned context frames are within 1.4×10^{-4} of the truth, so SDA has the same information as ForcingDAS. **EnKF** and **EnKS** have no trained parameters; their configuration is in Appendix S5.5. All four baselines assimilate the shared observations described above.

S3.1.6 ADDITIONAL RESULTS

Table S3 gives the with-context numbers behind Fig. 1(a); the cold-start counterpart is in Appendix S4.1. EnKF and EnKS have access to the *exact* PDE dynamics (with the resolution-induced model error discussed in Appendix S5.5), whereas ForcingDAS uses only a learned prior and no knowledge of the governing equations. Their results therefore indicate what a physics-based filter or smoother with the true governing equations achieves on this benchmark.

Table S3: Navier–Stokes vorticity assimilation under SO-5% with 10 clean context frames ($K = 50$, scored on the 40 assimilated frames): NRMSE (↓) and the radially averaged kinetic-energy spectrum relative error (↓) over mid-k ($k \in [8, 32)$, the inertial range) and all-k ($k \in [0.5, 64)$). Mean ± std over 16 held-out trajectories; every method assimilates identical observations. SDA receives the 10 clean context frames by replacement conditioning (Appendix S3.1.5). EnKS-FL uses a 10-frame lag, matching ForcingDAS-Pyr. Cold-start results are in Table S13.

	Filtering			Fix-lag Smoothing		Full-sequence Smoothing		
Metric	FlowDAS	EnKF	ForcingDAS-AR	EnKS-FL	ForcingDAS-Pyr	EnKS-FS	SDA	ForcingDAS-FS
NRMSE	0.141 ± 0.006	0.108 ± 0.003	**0.071** ± 0.004	0.113 ± 0.003	**0.057** ± 0.003	0.116 ± 0.003	0.075 ± 0.002	**0.067** ± 0.003
mid-k	0.607 ± 0.044	0.684 ± 0.009	**0.160** ± 0.019	0.530 ± 0.013	**0.218** ± 0.015	0.430 ± 0.016	0.193 ± 0.017	**0.139** ± 0.004
all-k	0.617 ± 0.041	0.824 ± 0.019	**0.215** ± 0.020	0.657 ± 0.022	**0.279** ± 0.016	0.550 ± 0.023	0.251 ± 0.027	**0.189** ± 0.008

S3.2 SEVIR Radar Nowcasting

S3.2.1 Dataset and physical setting

SEVIR-LR (Veillette et al., 2020) provides spatio-temporal radar imagery sampled from severe-weather events. We use the Vertically Integrated Liquid (VIL) channel, which encodes precipitation intensity as a single-channel field at 128×128 resolution and 5-minute temporal cadence. We use a SEVIR-LR build that keeps all 49 frames of each event. This differs from the standard 25-frame SEVIR-LR build and date-based split that FlowDAS's own pipeline uses. The 19,207 VIL events with no missing data are shuffled (seed 42) and split 90/10 into 17,287 training and 1,920 held-out events. From the held-out events, 30 were sampled (seed 2026) and 6 were discarded for insufficient convective activity (fewer than 5% of pixels above VIL 16 on the uint8 scale over all 49 frames), leaving 24 curated events. We evaluate on 16 of them (4 inspected during early development plus the 12 most active of the remaining 20), which are not used for model selection. Their activity ranges from 9% to 81% of pixels above VIL 16. Four further held-out events, disjoint from the 16 and from all training data, form the tuning set. Trajectories have $K = 49$ frames with context (6 clean, 43 assimilated) and $K = 30$ frames in the cold start (the first 30 frames of the same events). SEVIR has no closed-form physical model for VIL evolution, so no PDE-based DA baseline exists; FlowDAS (Chen et al., 2025) is the only learned filter we are aware of that has been applied at this resolution and modality.

The raw VIL data are stored as 8-bit integer pixel intensities and are loaded as floating-point values rescaled to the unit interval $[0, 1]$ by dividing by 255. This unit interval is the data space used by both the diffusion model and the observation operator: in the convention where "raw data" means $[0, 1]$ VIL, no further rescaling is applied (i.e. $b = 1$). For comparability with the standard SEVIR CSI thresholds, which are reported on the 0–255 uint8 scale, we give both conventions below.

S3.2.2 Observation model and noise calibration

Operators. For SEVIR we evaluate four observation operators (cf. Setup in §4.2): SO-10% and SO-20% (random pixel masks retaining 10% resp. 20% of pixels; one seed-42 mask shared across frames and events, observing 1,637 and 3,295 of the $16{,}384$ pixels) and SR2x, SR4x (bilinear down-sampling by 2 resp. 4).

Noise. Gaussian noise of standard deviation $\sigma_y^{\text{data}} = 0.05$ is added to the clean operator output, identically to NS. The noisy observations are generated once per event (seed $12345 + i$ for event i) and stored; every method (ForcingDAS, FlowDAS, SDA) assimilates these identical values, and the evaluation script checks this before scoring. Cold-start runs use the first 30 frames of the same observations.

Equivalent noise in physical units. Because the dataset's data space coincides with the $[0, 1]$ VIL convention ($b = 1$), Eq. (S6) gives $\sigma_y^{\text{raw, [0,1]}} = 0.05$: the data-space and raw-space noise levels coincide. Translated into the 0–255 uint8 SEVIR convention (multiply by 255):

$$\sigma_y^{\text{raw, [0,255]}} \;=\; 0.05 \times 255 \;=\; 12.75. \tag{S10}$$

For reference, the lowest CSI threshold in the SEVIR evaluation suite is 16 on the same uint8 scale (Table S5), so the noise level is $\approx 80\%$ of the smallest nowcasting bin width and $\sim 6\%$ of the largest.

S3.2.3 TRAINING COMPUTE

We train the ForcingDAS DiT-B backbone on the SEVIR-VIL training set of shape $(17287, 49, 128, 128)$ (trajectories $\times$ frames $\times$ height $\times$ width) in float32 across 4 NVIDIA A800 GPUs for 25 epochs, with per-GPU batch size 8 (total batch size 32). Total training time is approximately 22 hours.

S3.2.4 SAMPLING HYPERPARAMETERS

Table S4 lists the (γ, ζ) behind every ForcingDAS number. ForcingDAS-Pyr runs with a 20-frame lag ($u = 5$). For all three schedules, (γ, ζ) were selected on the 4 held-out tuning events, which are disjoint from the 16 evaluation events, by the lowest mean NRMSE over the assimilated frames.

Table S4: SEVIR sampling hyperparameters (γ, ζ) of ForcingDAS (DiT-B backbone, 100 DDIM steps, $\sigma_y^{\text{data}} = 0.05$). AR = autoregressive (filtering), Pyr = pyramid (fixed-lag smoothing, 20-frame lag, $u = 5$), FS = full-sequence (smoothing).

Protocol	Schedule	SO-10%	SO-20%	SR2x	SR4x
With context	AR	(0.005, 1.0)	(0.007, 0.7)	(0.015, 0.5)	(0.010, 0.7)
($K = 49$, 6 clean)	Pyr	(0.010, 8.0)	(0.005, 4.0)	(0.005, 2.0)	(0.010, 2.0)
	FS	(0.005, 8.0)	(0.015, 8.0)	(0.015, 4.0)	(0.005, 4.0)
Cold start	AR	(0.010, 1.0)	(0.005, 0.7)	(0.005, 0.5)	(0.015, 0.7)
($K = 30$, 0 clean)	Pyr	(0.010, 8.0)	(0.005, 4.0)	(0.015, 2.0)	(0.015, 4.0)
	FS	(0.005, 8.0)	(0.015, 8.0)	(0.015, 4.0)	(0.005, 4.0)

S3.2.5 BASELINE CONFIGURATIONS

FlowDAS (Chen et al., 2025) was retrained on our 49-frame SEVIR build and training split (99,250 steps, 23 h on one A40 GPU) and samples with 500 Euler–Maruyama steps and 25 Monte Carlo draws for the guidance estimate. Its guidance scale was first selected on the 4 tuning events from two values per setting, then re-searched over $\zeta \in \{0.05, 0.1, 0.2, 0.5, 1.0, 1.5, 2.0\}$. The re-search confirmed every original value: $\zeta = 0.1/0.05/0.1/0.1$ with context and $0.2/0.1/0.2/0.2$ in the cold start (SO-10%/SO-20%/SR2x/SR4x). Using short 3-step rollouts as a cheap tuning proxy picked larger steps in four settings, and each of these was worse at full length on the tuning events: a large step corrects quickly over a few frames but accumulates error over 30–43 frames. **SDA** (Rozet & Louppe, 2023) uses a score network trained on our training split (Markov blanket window 5, channels $(96, 192, 384)$, 3 blocks per level, embedding dimension 64, AdamW with learning rate 2×10^{-4} and weight decay 10^{-3}, batch size 32; 119 epochs, 10.4 h on one A40), sampled with 256 steps, 1 Langevin correction, $\tau = 0.5$, and $\gamma = 0.01$, selected on the 4 held-out tuning events. SDA reconstructs the full trajectory jointly and has no clean-context mode. As on NS (Appendix S3.1.5), passing the context frames as fully observed frames makes sampling diverge, so with context we use replacement conditioning: after every predictor and corrector update, the 6 context frames are overwritten with the ground truth diffused to the current noise level, and the likelihood term is unchanged. Because the last replacement still carries a small amount of diffusion noise, SDA's returned context frames are within 5×10^{-3} of the truth, whereas the other methods copy them exactly; this slightly disadvantages SDA. No method's prediction is clamped; values are clipped to $[0, 255]$ only when computing CSI. For reference, ForcingDAS was trained for 25 epochs ($\approx$88 GPU-hours on A800; Appendix S3.2.3).

S3.2.6 PER-THRESHOLD CSI

Table S5 reports the per-threshold Critical Success Index (CSI) of all methods on SEVIR VIL across the four observation settings with context; the cold-start counterpart is Table S15. Thresholds are SEVIR VIL pixel intensities on the uint8 scale; higher thresholds indicate heavier precipitation, which is increasingly rare and harder to reconstruct. CSI is pooled: hits, misses, and false alarms

are summed over all assimilated frames of all 16 events, following the SEVIR convention, and the ± is the std over 2,000 bootstrap resamples of the events. Per-event CSI is noisy at high thresholds because several events contain almost no pixels above VIL 133.

Table S5: SEVIR per-threshold pooled CSI (↑; ± std over bootstrap resamples of the 16 events) with 6 clean context frames, grouped by DA regime. ForcingDAS-Pyr uses a 20-frame lag; SDA receives the context frames by replacement conditioning. Cold-start results are in Table S15.

Obs	Regime	Method	CSI-16	CSI-74	CSI-133	CSI-160	CSI-181	CSI-219	Mean
SO-10%	Filtering	FlowDAS	0.668 ± 0.049	0.682 ± 0.041	0.502 ± 0.027	0.305 ± 0.013	0.196 ± 0.029	0.066 ± 0.040	0.403 ± 0.020
		ForcingDAS-AR	**0.820** ± 0.023	**0.787** ± 0.032	**0.624** ± 0.032	**0.457** ± 0.024	**0.394** ± 0.035	**0.257** ± 0.037	**0.557** ± 0.024
	Fix-lag	ForcingDAS-Pyr	0.840 ± 0.020	0.809 ± 0.028	0.649 ± 0.026	0.490 ± 0.021	0.426 ± 0.030	0.293 ± 0.031	0.584 ± 0.021
	Full-seq.	SDA	0.820 ± 0.025	0.797 ± 0.031	0.629 ± 0.025	0.477 ± 0.021	0.425 ± 0.031	**0.294** ± 0.037	0.573 ± 0.021
		ForcingDAS-FS	**0.853** ± 0.018	**0.815** ± 0.028	**0.659** ± 0.027	**0.498** ± 0.021	**0.437** ± 0.032	0.291 ± 0.033	**0.592** ± 0.021
SO-20%	Filtering	FlowDAS	0.711 ± 0.047	0.736 ± 0.040	0.564 ± 0.032	0.384 ± 0.021	0.300 ± 0.029	0.190 ± 0.032	0.481 ± 0.023
		ForcingDAS-AR	**0.850** ± 0.020	**0.834** ± 0.026	**0.692** ± 0.025	**0.543** ± 0.021	**0.493** ± 0.032	**0.340** ± 0.038	**0.625** ± 0.020
	Fix-lag	ForcingDAS-Pyr	0.884 ± 0.015	0.861 ± 0.022	0.730 ± 0.020	0.595 ± 0.017	0.546 ± 0.027	0.411 ± 0.034	0.671 ± 0.017
	Full-seq.	SDA	0.849 ± 0.022	0.837 ± 0.027	0.691 ± 0.020	0.557 ± 0.018	0.516 ± 0.027	0.384 ± 0.033	0.639 ± 0.017
		ForcingDAS-FS	**0.879** ± 0.015	**0.853** ± 0.023	**0.721** ± 0.021	**0.585** ± 0.018	**0.537** ± 0.027	**0.399** ± 0.032	**0.662** ± 0.017
SR2x	Filtering	FlowDAS	0.798 ± 0.039	0.841 ± 0.028	0.720 ± 0.020	0.575 ± 0.020	0.514 ± 0.029	0.373 ± 0.030	0.637 ± 0.017
		ForcingDAS-AR	**0.872** ± 0.017	**0.865** ± 0.022	**0.745** ± 0.019	**0.616** ± 0.022	**0.575** ± 0.032	**0.431** ± 0.035	**0.684** ± 0.017
	Fix-lag	ForcingDAS-Pyr	0.898 ± 0.014	0.886 ± 0.019	0.774 ± 0.015	0.657 ± 0.017	0.618 ± 0.025	0.488 ± 0.031	0.720 ± 0.014
	Full-seq.	SDA	0.853 ± 0.022	0.850 ± 0.026	0.715 ± 0.019	0.592 ± 0.019	0.559 ± 0.026	0.433 ± 0.031	0.667 ± 0.015
		ForcingDAS-FS	**0.893** ± 0.014	**0.881** ± 0.020	**0.766** ± 0.017	**0.645** ± 0.018	**0.609** ± 0.027	**0.476** ± 0.032	**0.712** ± 0.014
SR4x	Filtering	FlowDAS	0.703 ± 0.045	0.726 ± 0.039	0.558 ± 0.033	0.376 ± 0.023	0.286 ± 0.028	0.178 ± 0.030	0.471 ± 0.024
		ForcingDAS-AR	**0.830** ± 0.022	**0.792** ± 0.031	**0.623** ± 0.033	**0.460** ± 0.027	**0.392** ± 0.038	**0.252** ± 0.036	**0.558** ± 0.026
	Fix-lag	ForcingDAS-Pyr	0.833 ± 0.018	0.811 ± 0.028	0.645 ± 0.032	0.458 ± 0.028	0.370 ± 0.042	0.237 ± 0.044	0.559 ± 0.026
	Full-seq.	SDA	0.802 ± 0.028	0.782 ± 0.033	0.608 ± 0.026	0.454 ± 0.020	0.398 ± 0.031	0.261 ± 0.034	0.551 ± 0.021
		ForcingDAS-FS	**0.847** ± 0.019	**0.816** ± 0.028	**0.653** ± 0.026	**0.491** ± 0.022	**0.426** ± 0.035	**0.279** ± 0.037	**0.585** ± 0.022

S3.3 ERA5 Global Atmospheric Reanalysis

S3.3.1 Dataset and physical setting

We use a 4-channel multi-variable subset of WeatherBench2 ERA5 (Rasp et al., 2020; Hersbach et al., 2020) at $\sim 1.5^\circ$ resolution with 6-hourly temporal resolution. The native grid is 240×121 (longitude $\times$ latitude); we drop the North-Pole row, so every model and metric works on 240×120. The four channels are Z500, T850, U10, V10 (Appendix S3.3.2). We train on 1979–2015. 2016 is held out from training; the 16 evaluation windows and the 4 tuning windows below are disjoint. We evaluate on 16 trajectories of $K = 30$ frames (7.5 days) from 2016, one window every 92 time steps (23 days) starting at 00 UTC, so the windows cover all seasons. A separate tuning set of 4 windows from 2016 (starting at time steps 136, 500, 864, 1228) is verified not to overlap the 16 evaluation windows, and it is the only data used for the hyperparameter selections described in Appendix S3.3.6. Appendix S1.1 surveys related learned-DA work on ERA5, and Appendix S5 describes the 3D-Var, FlowDAS, and Tensor-Var baselines.

The dataset stores values that are z-score normalized per channel using climatological statistics from the training period: $\tilde{x}_{t,c} = (x^{\text{raw}}_{t,c} - \mu_c)/\sigma_c$. Hence the data space loaded by the model is the z-score-normalized space (per-channel mean 0, std 1), with no further rescaling inside the algorithm. The per-channel climatological scales used to convert between the data space and the raw physical space are summarized in Table S6.

S3.3.2 Variable descriptions

Table S7 summarizes the four ERA5 variables used in §4.3.

Why these four variables. Z500 and T850 are the standard upper-air evaluation pair in the DA literature, testing both dynamical (pressure) and thermodynamic (temperature) fidelity. U10 and V10 are surface wind components that are directly observable by weather stations, buoys, and satellite scatterometers, making them natural candidates for the observation operator in DA experiments. The surface wind field is also physically coupled to the upper-air geopotential via *geostrophic balance*:

Table S6: Per-channel climatological mean μ_c and standard deviation σ_c (in raw physical units), computed from the ERA5 1979–2015 training period. These are the conversion factors between the data space (where the model and the observation operator act) and the raw physical space, and every method and metric uses them.

Variable	μ_c	σ_c	Units
Z500	54 085.09	3,363.11	$\mathrm{m^2\,s^{-2}}$
T850	274.23	15.72	K
U10	−0.053	5.440	$\mathrm{m\,s^{-1}}$
V10	0.188	4.671	$\mathrm{m\,s^{-1}}$

Table S7: ERA5 variables used in the multi-variable experiment. Typical ranges are approximate global values across seasons.

Variable	**Level**	**Units**	**Typical range**	**Physical meaning**
Z500	500 hPa	$\mathrm{m^2\,s^{-2}}$	48,000–58,000	Geopotential at 500 hPa. Encodes the height of the mid-tropospheric pressure surface, capturing synoptic-scale Rossby waves, troughs, ridges, and the jet stream. The single most-reported variable in DA evaluation.
T850	850 hPa	K	250–300	Temperature at 850 hPa (≈1.5 km altitude). Captures low-level thermal structure including warm/cold air advection and frontal zones. Strongly coupled to surface weather.
U10	10 m	$\mathrm{m\,s^{-1}}$	−20 to +20	Zonal (east–west) component of 10-meter wind. Positive values indicate eastward flow. Exhibits sharper spatial gradients than upper-air fields, particularly near coastlines and orography.
V10	10 m	$\mathrm{m\,s^{-1}}$	−20 to +20	Meridional (north–south) component of 10-meter wind. Together with U10, completes the surface wind vector.

large-scale winds flow approximately parallel to geopotential contours, with speed proportional to the geopotential gradient, so the four channels are not independent.

Normalization. The four variables span three orders of magnitude in raw values (Table S7). We apply per-variable z-score standardization:

$$\tilde{x}_{t,c} = \frac{x_{t,c} - \mu_c}{\sigma_c}, \quad c \in \{\text{Z500, T850, U10, V10}\}, \tag{S11}$$

where μ_c and σ_c are the climatological mean and standard deviation of variable c, computed over the training period (Table S6). After normalization, all channels are approximately $\mathcal{N}(0,1)$-distributed, allowing the diffusion model to treat them on equal footing. This follows standard practice in the learned weather prediction literature (Huang et al., 2024; Andry et al., 2025; Xu et al., 2025).

S3.3.3 Observation model and noise calibration

Operator. We use a sparse pixel observation (SO) operator: a single random binary mask of shape $H \times W$ is shared across all 4 channels, modeling co-located meteorological measurements. The mask observes 10% of grid points (seed 42), and it is the same mask for every frame, trajectory, and method.

Noise. The data-space noise level is $\sigma_y^{\text{data}} = 0.05$. The noise is i.i.d. in space and across channels. The observations $\boldsymbol{y} = \boldsymbol{m} \odot \boldsymbol{x} + 0.05\,\boldsymbol{\varepsilon}$ are generated once per trajectory and stored, and every method (ForcingDAS, 3D-Var, FlowDAS, Tensor-Var) assimilates these identical values; the evaluation script checks this, and checks that context frames equal the ground truth, before scoring. Per-trajectory differences between methods are therefore exactly paired.

Equivalent per-channel noise in raw physical units. For ERA5, Eq. (S6) applies *per channel*, with $b = \sigma_c$ taken from Table S6: $\sigma_y^{\text{raw},c} = \sigma_c \cdot \sigma_y^{\text{data}}$. The operator (the same shared mask) is unchanged in raw space; the noise becomes anisotropic, with a different physical std per variable. Table S8 reports the resulting raw-space noise standard deviations. For example, $\sigma_y^{\text{data}} = 0.05$ corresponds to a noise of $\sim 168\,\text{m}^2\,\text{s}^{-2}$ on Z500 (a small fraction of typical synoptic anomalies, which are $\mathcal{O}(10^3)\,\text{m}^2\,\text{s}^{-2}$) and $\sim 0.79\,\text{K}$ on T850 (well below typical regional temperature variability).

Table S8: ERA5 observation noise standard deviation in raw physical units, derived from $\sigma_y^{\text{data}} = 0.05$ via $\sigma_y^{\text{raw},c} = \sigma_c \cdot \sigma_y^{\text{data}}$ using the climatological standard deviations of Table S6.

σ_y^{data}	σ_y^{Z500} ($\text{m}^2\,\text{s}^{-2}$)	σ_y^{T850} (K)	σ_y^{U10} ($\text{m}\,\text{s}^{-1}$)	σ_y^{V10} ($\text{m}\,\text{s}^{-1}$)
0.05	168.16	0.786	0.272	0.234

S3.3.4 Evaluation metrics: NRMSE and ACC

We report two per-variable evaluation metrics for ERA5 assimilation, both evaluated on the equiangular 240×120 longitude–latitude grid (after dropping the North-Pole row) and both applying latitude weights that compensate for grid-cell area shrinking toward the poles. All metrics are computed identically for every method in Table 2.

Notation. Let $\hat{s}_{k,b,i,j}$ denote the assimilated value at frame k of trajectory b, longitude index $j \in \{1, \dots, J\}$, latitude index $i \in \{1, \dots, I\}$; let $s_{k,b,i,j}$ denote the ERA5 reference; and let $c_{k,b,i,j}$ denote the per-grid-point climatology at the valid time of frame k (varying with latitude, longitude, day-of-year, and hour-of-day). Metrics are computed over the scored frames $k \in \mathcal{S}$ only: the 24 assimilated frames with context (the 6 context frames are copies of the truth), and frames 1–29 in the cold start (Appendix S4.3). All quantities are in z-score-normalized data space (§S3.3.1).

Latitude weights. We use cosine-of-latitude weights normalized so that their mean over the I latitude rows equals one:

$$w(i) = \frac{\cos\theta_i}{\frac{1}{I}\sum_{i'=1}^{I}\cos\theta_{i'}}, \tag{S12}$$

where θ_i is the latitude of row i. This is the standard WeatherBench2 weighting; it is numerically equivalent (relative error $<0.005\%$) to the exact cell-area formula $w(i) \propto \sin\theta_i^u - \sin\theta_i^l$ on our 1.5° equiangular grid.

Latitude-weighted NRMSE. Because the data is z-score normalized per channel (per-channel std ≈ 1), the latitude-weighted RMSE in data space equals the normalized RMSE. For each variable we compute the per-frame RMSE, average it over the scored frames of trajectory b, and report the mean and standard deviation of these per-trajectory values over the $B = 16$ trajectories:

$$\text{NRMSE}_b = \frac{1}{|\mathcal{S}|}\sum_{k\in\mathcal{S}}\sqrt{\frac{1}{IJ}\sum_{i=1}^{I}\sum_{j=1}^{J} w(i)\left(\hat{s}_{k,b,i,j} - s_{k,b,i,j}\right)^2}. \tag{S13}$$

Anomaly Correlation Coefficient (ACC). We subtract the per-grid-point, per-time climatology to obtain anomalies $a_{k,b,i,j} = \hat{s}_{k,b,i,j} - c_{k,b,i,j}$ and $r_{k,b,i,j} = s_{k,b,i,j} - c_{k,b,i,j}$, and then compute the latitude-weighted Pearson correlation between predicted and reference anomalies jointly over the scored frames and spatial points of each trajectory b, again reporting mean $\pm$ std over trajectories:

$$\text{ACC}_b = \frac{\sum_{k\in\mathcal{S},i,j} w(i)\, a_{k,b,i,j}\, r_{k,b,i,j}}{\sqrt{\sum_{k\in\mathcal{S},i,j} w(i)\, a_{k,b,i,j}^2}\,\sqrt{\sum_{k\in\mathcal{S},i,j} w(i)\, r_{k,b,i,j}^2}}. \tag{S14}$$

ACC takes values in $[-1, 1]$; in operational practice, $\text{ACC} > 0.6$ is considered useful skill and $\text{ACC} > 0.8$ high skill. ACC is dimensionless and invariant to per-variable affine rescaling, so it is unaffected by the choice of normalization. Pooling the sums of Eq. (S14) over all trajectories instead changes the reported values by at most 0.001 for the learned methods and by up to 0.004 for 3D-Var (T850).

Climatology source. The climatology $c_{k,b,i,j}$ is the WeatherBench2 hourly ERA5 climatology (1990–2019 base period, 240×121 equiangular grid, cropped to 240×120), indexed by the day-of-year and hour-of-day matching each frame's valid time. It varies with latitude, longitude, and calendar time (the full latitude–longitude climatology, as in WeatherBench2). The same fixed climatology is used for all methods; in z-score data space it is $c^{(\mathrm{z})}_{k,b,i,j} = (c^{(\mathrm{raw})}_{k,b,i,j} - \mu_c)/\sigma_c$ using the per-channel scales of Table S6.

S3.3.5 Training compute

We train the ForcingDAS DiT-B backbone on the ERA5 training set of shape $(1351, 40, 4, 240, 120)$ (trajectories $\times$ frames $\times$ channels $\times$ longitude $\times$ latitude) in float32 across 4 NVIDIA A800 GPUs for 300 epochs, with per-GPU batch size 4 (total batch size 16). Total training time is approximately 33 hours.

S3.3.6 Sampling hyperparameters

For ERA5 we use the same per-frame variance-reweighted guidance as on NS and SEVIR. Table S9 lists the values behind every ForcingDAS number. ForcingDAS-Pyr runs with a 10-frame lag ($u = 10$). For all three schedules, (γ, ζ) were selected on the 4 held-out tuning windows; no value was selected on the 16 evaluation trajectories.

Table S9: ERA5 sampling hyperparameters (DiT-B backbone, 100 DDIM steps, SO-10%, $\sigma_y^{\mathrm{data}} = 0.05$). AR = autoregressive (filtering), Pyr = pyramid (fixed-lag smoothing, $u = 10$), FS = full-sequence (smoothing). "Lag" is the number of frames that are denoised jointly.

Protocol	K (frames)	Schedule	Lag	γ	ζ
With context (6 clean)	30	AR	1	0.005	1.0
	30	Pyr	10	0.005	2.0
	30	FS	24	0.001	4.0
Cold start (0 clean)	30	AR	1	0.005	1.0
	30	Pyr	10	0.005	2.0
	30	FS	30	0.005	8.0

S3.4 Effect of Causality-Aware Training on Probabilistic Forecasting

This subsection isolates the effect of *Causality-Aware Training* (CAT, §3.1) on free-running probabilistic forecasting, conditioning on a clean context window but *without* observation guidance. Standard Diffusion Forcing samples per-frame noise levels i.i.d. uniformly at training time, while every DA scheduling matrix imposes a causally monotone pattern at inference; the resulting train–test gap exists already in pure generation, so probabilistic forecasting is the cleanest setting in which to isolate it.

S3.4.1 Setup

Protocol. We run free-running probabilistic forecasting with ForcingDAS-AR ($u = T$): the first C frames of each test trajectory are clean context, and the model autoregressively samples the next $H = 3$ frames from the per-frame diffusion prior, with no observation guidance applied. Following the conventions of the main experiments, $C = 10$ for NS, $C = 6$ for SEVIR, and $C = 6$ for ERA5. We evaluate on 4 held-out trajectories per benchmark, a smaller set than the 16 of §4. For each starting condition we draw an ensemble of $M = 16$ samples by re-running the reverse process with independent random seeds; all samples share the same clean context.

Models compared. For each benchmark we train ForcingDAS networks that are identical in architecture, optimizer, learning-rate schedule, and total training steps; the *only* difference is the per-frame noise distribution at training time:

- **Without CAT** ($\rho = 0$): the per-frame noise vector $\boldsymbol{t}$ is drawn i.i.d. from $\mathrm{Uniform}\{1, \ldots, T\}$, recovering the original DF training scheme.

- **With CAT** ($\rho > 0$): the CAT mixture of Eq. (4) is used. We sweep $\rho \in \{0.25, 0.75\}$ on NS and SEVIR and report both rows in Tables S10–S11; on ERA5 we report $\rho = 0.25$. The remaining mixture parameters $(\rho_c, C_{\max})$ are shared across the compared models.

Any difference in forecasting quality is therefore attributable to the training noise distribution alone.

Metrics. We report two metrics per dataset: the Continuous Ranked Probability Score (CRPS, §S3.4.2) for the full forecast distribution, and NRMSE on the ensemble mean for deterministic skill. Both are reported per lead time $h \in \{1, 2, 3\}$ and as a mean over the three lead times. For ERA5 we report both metrics per variable (Z500, T850, U10, V10) with cosine-of-latitude weights, matching the main-text protocol (Appendix S3.3.4).

S3.4.2 Continuous Ranked Probability Score

CRPS (Gneiting & Raftery, 2007) is the standard strictly proper scoring rule for univariate probabilistic forecasts. Given a predictive cumulative distribution function (CDF) F and an observation $y \in \mathbb{R}$, it is defined as

$$\mathrm{CRPS}(F, y) = \int_{-\infty}^{\infty} \big(F(z) - \mathbf{1}\{y \le z\}\big)^2 \, \mathrm{d}z, \tag{S15}$$

the integrated squared distance between the forecast CDF and the step CDF of the truth. Lower values are better; CRPS reduces to the mean absolute error when F is a point mass, so it generalizes deterministic skill in a calibration-aware way.

Ensemble form. We have access to F only through an ensemble of M samples $\{x^{(1)}, \ldots, x^{(M)}\}$. The standard sample-form estimator (Gneiting & Raftery, 2007; Hersbach, 2000) is

$$\widehat{\mathrm{CRPS}}\Big(\{x^{(i)}\}_{i=1}^{M}, y\Big) = \frac{1}{M}\sum_{i=1}^{M} \big|x^{(i)} - y\big| - \frac{1}{2M^2}\sum_{i=1}^{M}\sum_{j=1}^{M} \big|x^{(i)} - x^{(j)}\big|, \tag{S16}$$

a difference between an accuracy term (mean absolute deviation of samples from the truth) and a sharpness term (mean pairwise spread). For $M = 1$, the second term vanishes and Eq. (S16) reduces to $|x^{(1)} - y|$, the deterministic absolute error. We use $M = 16$ throughout.

Aggregation across pixels and trajectories. For a 2D field, we compute Eq. (S16) per pixel and per channel and then average over pixels and held-out trajectories at fixed lead time h. For ERA5 the spatial average uses the cosine-of-latitude weights of Eq. (S12), so the reported per-variable CRPS is a latitude-weighted mean analogous to the latitude-weighted NRMSE of Eq. (S13).

S3.4.3 Results

Table S10: Effect of CAT on probabilistic forecasting on Navier–Stokes vorticity. ForcingDAS-AR ($u = T$), $C = 10$ clean context frames, $H = 3$-frame forecast horizon, ensemble size $M = 16$, averaged over 4 held-out trajectories. CRPS and ensemble-mean NRMSE are reported per lead time and as a mean over $h \in \{1, 2, 3\}$; lower is better for both. Two CAT mixing ratios are reported.

	CRPS (↓)				NRMSE on ensemble mean (↓)			
Method	h=1	h=2	h=3	mean	h=1	h=2	h=3	mean
Without CAT ($\rho = 0$)	0.033	**0.046**	**0.049**	**0.043**	0.106	**0.149**	**0.158**	**0.137**
With CAT ($\rho = 0.25$)	0.035	0.048	0.050	0.044	0.111	0.155	0.160	0.142
With CAT ($\rho = 0.75$)	**0.032**	0.048	0.050	**0.043**	**0.102**	0.154	0.162	0.140

The train–test gap. Standard DF training samples the per-frame diffusion-step vector $\boldsymbol{t} = (t_1, \ldots, t_K)$ i.i.d. from $\mathrm{Uniform}\{1, \ldots, T\}$. Inference, in contrast, follows a scheduling matrix of the form Eq. (7), in which earlier frames are always at no-higher diffusion steps than later ones. The model is therefore trained on noise configurations that are, with overwhelming probability, very different from the configurations it sees at test time. This gap matters for scientific dynamical systems, where the future is tightly determined by the past: forecasting quality depends on the model's ability

Table S11: Effect of CAT on probabilistic forecasting on SEVIR-VIL. ForcingDAS-AR ($u = T$), $C = 6$ clean context frames, $H = 3$-frame forecast horizon, ensemble size $M = 16$, averaged over 4 held-out trajectories. Same column convention as Table S10.

	CRPS (↓)				NRMSE on ensemble mean (↓)			
Method	h=1	h=2	h=3	mean	h=1	h=2	h=3	mean
Without CAT ($\rho = 0$)	**0.013**	0.019	0.025	0.019	**0.175**	0.258	0.339	0.257
With CAT ($\rho = 0.25$)	0.014	0.020	0.026	0.020	**0.175**	0.251	0.314	0.247
With CAT ($\rho = 0.75$)	**0.013**	**0.018**	**0.022**	**0.018**	0.177	**0.243**	**0.294**	**0.238**

Table S12: Effect of CAT on probabilistic forecasting on ERA5. ForcingDAS-AR ($u = T$), $C = 6$ clean context frames, $H = 3$-frame forecast horizon, ensemble size $M = 16$, averaged over 4 held-out length-30 trajectories from 2016. Per-variable, latitude-weighted CRPS and ensemble-mean NRMSE in z-score-normalized data space; lower is better for both. Per lead time $h \in \{1, 2, 3\}$ and as a mean over the three lead times.

		CRPS (↓)				NRMSE on ensemble mean (↓)			
Method	Variable	h=1	h=2	h=3	mean	h=1	h=2	h=3	mean
Without CAT ($\rho = 0$)	Z500	0.030	0.052	0.075	0.053	0.050	0.086	0.122	0.086
	T850	0.039	0.057	0.076	0.058	0.065	0.090	0.116	0.091
	U10	0.091	0.119	0.152	0.121	0.165	0.217	0.279	0.220
	V10	0.120	0.154	0.199	0.158	0.215	0.278	0.358	0.284
With CAT ($\rho = 0.25$)	Z500	**0.018**	**0.027**	**0.037**	**0.027**	**0.033**	**0.049**	**0.067**	**0.050**
	T850	**0.031**	**0.038**	**0.046**	**0.038**	**0.056**	**0.069**	**0.084**	**0.070**
	U10	**0.086**	**0.107**	**0.135**	**0.109**	**0.160**	**0.203**	**0.260**	**0.208**
	V10	**0.109**	**0.137**	**0.168**	**0.138**	**0.201**	**0.257**	**0.321**	**0.260**

to use a clean past as a reliable conditioning signal, and training with i.i.d. noise levels spends most of its capacity on configurations with a noisy past.

CAT noise distribution. With probability ρ (the *causal ratio*) the i.i.d. noise vector is sorted into a non-decreasing staircase, $\boldsymbol{t}^{\text{sorted}} = \text{sort}(t_1, \ldots, t_K)$ with $t_1^{\text{sorted}} \leq \cdots \leq t_K^{\text{sorted}}$; with probability $1 - \rho$ the i.i.d. draw is kept. With probability ρ_c a random number of leading frames are additionally clamped to diffusion step zero, exposing the model to the clean-context conditioning it sees at inference. The mixture preserves robustness to arbitrary noise patterns while biasing capacity toward the causally monotone configurations that dominate inference; setting $\rho = 0$ recovers standard DF training.

Results. Tables S10–S12 report CRPS and ensemble-mean NRMSE per lead time on the three benchmarks. CAT is most effective on ERA5 (Table S12): every per-variable, per-lead-time cell improves over the i.i.d.-trained baseline, with the largest reductions on the synoptic-scale fields Z500 and T850 (mean CRPS on Z500 roughly halves, $0.053 \to 0.027$; T850, $0.058 \to 0.038$). On SEVIR (Table S11), $\rho = 0.75$ gives consistent gains that grow with the forecast horizon: NRMSE at $h = 3$ drops from 0.339 to 0.294 (a 13% relative reduction) and mean CRPS from 0.019 to 0.018. On NS (Table S10), CAT is essentially flat: the i.i.d. baseline and the two CAT settings track each other within a few percent on every metric and lead time.

Why CAT helps where the observed sequence is non-Markovian. The pattern across the three benchmarks mirrors the Markovian / non-Markovian distinction of §4. On NS, the vorticity field is the *full* state of 2D incompressible flow with known dynamics: once a clean current frame is available, future frames are determined by it alone, and additional past frames carry no extra information. The clean-past conditioning that AR inference relies on is therefore no more informative than what an i.i.d.-trained network already learns to use, leaving little headroom for CAT. SEVIR and ERA5 are different: the observed sequences are low-dimensional projections of much higher-dimensional latent states (a vertically integrated radar field; a 4-channel slice of a 37-level atmosphere). Past frames carry information about hidden degrees of freedom that the current frame alone does not, so the

inference-time pattern of clean past and noisy future is exactly the regime in which the model needs to be trained, and matching it through CAT translates into clear forecasting gains. That the gain on SEVIR widens with the forecast horizon, while ERA5 already shows large gains at $h = 1$, is consistent with this picture: SEVIR's per-step dependence on hidden state is moderate, so the gap compounds; ERA5's dependence is strong from the first step, since most of the atmospheric state is unobserved.

S4 COLD-START (NO-CONTEXT) DATA ASSIMILATION

The main text reports results under the with-context protocol, where the model receives a short clean prefix to seed assimilation. This matches how learned DA is typically deployed in operational pipelines, in which a recent reanalysis or forecast supplies the initial condition. We additionally evaluate ForcingDAS in the more demanding *cold-start* setting: no clean frames are provided, and the entire trajectory is assimilated from observations alone. Many learned DA methods cannot operate in this regime, or degrade severely; we use it as additional evidence of the framework's versatility.

We report cold-start results on the same three benchmarks as the main text. The evaluation protocol (16 held-out trajectories of length $K = 30$, observation operators, noise levels, and metrics) is unchanged from §4; only the context length is set to 0. Sampling hyperparameters are in Tables S2, S4, and S9.

S4.1 NAVIER–STOKES VORTICITY

Table S13 reports cold-start NS results at SO-5% ($K = 30$, all 30 frames assimilated and scored), with the same eight methods and three regimes as Table S3. Setup, shared observations, and metrics are as in Appendix S3.1; Fig. S1 visualizes a representative trajectory.

Table S13: Navier–Stokes vorticity, *cold-start* at SO-5% (no clean context, $K = 30$): NRMSE (↓) and the radially averaged kinetic-energy spectrum relative error (↓) over mid-k ($k \in [8, 32)$) and all-k ($k \in [0.5, 64)$). Mean $\pm$ std over 16 held-out trajectories; every method assimilates identical observations.

	Filtering			Fix-lag Smoothing		Full-sequence Smoothing		
Metric	FlowDAS	EnKF	ForcingDAS-AR	EnKS-FL	ForcingDAS-Pyr	EnKS-FS	SDA	ForcingDAS-FS
NRMSE	0.145 ± 0.004	0.108 ± 0.003	**0.080** ± 0.005	0.112 ± 0.003	**0.069** ± 0.003	0.114 ± 0.003	0.077 ± 0.003	**0.073** ± 0.003
mid-k	0.820 ± 0.028	0.686 ± 0.008	**0.319** ± 0.038	0.541 ± 0.013	**0.254** ± 0.021	0.479 ± 0.016	0.213 ± 0.019	**0.161** ± 0.012
all-k	0.869 ± 0.031	0.822 ± 0.023	**0.323** ± 0.033	0.665 ± 0.023	**0.307** ± 0.040	0.598 ± 0.025	0.280 ± 0.025	**0.208** ± 0.015

Among the full-sequence smoothers, ForcingDAS-FS is best on all three metrics. Its NRMSE margin over SDA is small in absolute terms (0.073 vs. 0.077) but systematic: paired over the 16 trajectories, ForcingDAS-FS is better on every trajectory for NRMSE, mid-k, and all-k (Wilcoxon $p = 3.1 \times 10^{-5}$; mean NRMSE difference -0.004, 95% paired-bootstrap CI $[-0.005, -0.003]$), and its spectrum errors are clearly lower (0.161 vs. 0.213 mid-k; 0.208 vs. 0.280 all-k). EnKS-FS, which runs the exact PDE, is well behind (0.114), consistent with the severely rank-deficient sample covariance of its 100-member ensemble over a 30-frame augmented state (Appendix S5.5). In filtering, ForcingDAS-AR again roughly halves FlowDAS's NRMSE and improves on the EnKF (0.080 vs. 0.108). In fixed-lag smoothing, ForcingDAS-Pyr improves on EnKS-FL over the same 10-frame lag (0.069 vs. 0.112). Removing the clean context costs ForcingDAS little (AR $0.071 \to 0.080$, FS $0.067 \to 0.073$ against Table S3).

Fig. S1 shows the cold-start behavior qualitatively. Without a clean prefix, FlowDAS never locks onto the flow: its fields show smooth periodic patterns unrelated to the truth, and its NRMSE stays near 0.15 over the whole window. EnKS-FS produces grainy fields, and its spectrum decays too fast at high wavenumbers. SDA and ForcingDAS-FS both track the truth. The ForcingDAS regimes need a few frames to settle when no context is given. This is most visible for AR, whose first few frames are noticeably worse; after the first 5–8 frames all three regimes hold a flat, low error.

S4.2 PRECIPITATION NOWCASTING (SEVIR-VIL)

Table S14 reports cold-start SEVIR results on the first 30 frames of the same 16 events, with the same shared observations; Table S15 gives the per-threshold CSI and Fig. S2 visualizes a representative event. Here SDA runs in its native mode, without any clean context.

Under matched context, ForcingDAS-FS improves on SDA on every operator, in NRMSE and in pooled CSI-mean and CSI-160 (Table S15). Because the spread across events (± 0.08–0.13 NRMSE) is much larger than the gap between methods, the paired per-event comparison is the informative one:

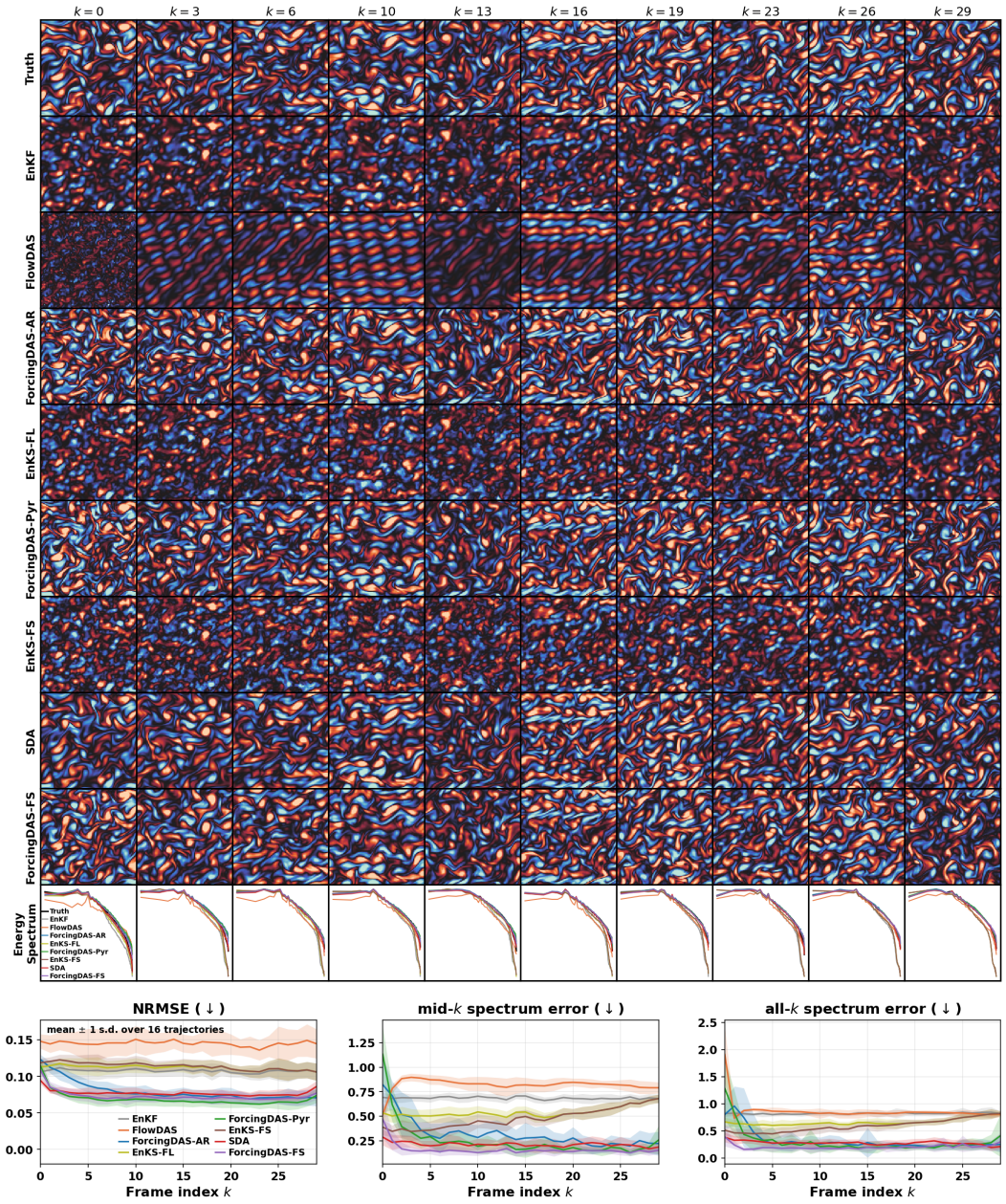

Figure S1: NS assimilation in the *cold-start* protocol (SO-5%, no clean context, $K = 30$). **Top nine rows:** ground-truth vorticity and the estimates of every method, grouped by regime: the filters EnKF, FlowDAS, and ForcingDAS-AR; the fixed-lag smoothers EnKS-FL and ForcingDAS-Pyr; and the full-sequence smoothers EnKS-FS, SDA (Rozet & Louppe, 2023), and ForcingDAS-FS, on a representative held-out trajectory, at $k \in \{0, 3, 6, 10, 13, 16, 19, 23, 26, 29\}$. **Tenth row:** per-frame radially averaged kinetic-energy spectrum on log–log axes. **Bottom:** per-frame NRMSE, mid-k, and all-k spectrum relative error (↓), mean ± 1 s.d. over all 16 held-out trajectories. Numbers are in Table S13.

ForcingDAS-FS has the lower NRMSE on 11, 13, 16, and 15 of the 16 events (SO-10%, SO-20%, SR2x, SR4x; Wilcoxon $p = 0.05, 0.02, 3 \times 10^{-5}, 9 \times 10^{-5}$). The gain is therefore consistent on SO-20%, SR2x, and SR4x and only a trend on SO-10%. Without a clean prefix, FlowDAS degrades sharply (NRMSE $0.616 \to 1.107$ on SO-10%, with a large spread across events, because it diverges on several of them), while ForcingDAS changes little (Pyr $0.293 \to 0.313$). Fig. S2 shows this on

Table S14: SEVIR VIL nowcasting, *cold-start* (no clean context, $K = 30$): NRMSE (↓; mean ± std over 16 events) and CSI-mean (↑) over thresholds $\{16, 74, 133, 160, 181, 219\}$ (pooled over all events and frames; ± std over bootstrap resamples of the events). ForcingDAS-Pyr uses a 20-frame lag. Best in *Filtering* and best in *Full-sequence Smoothing* per column in **bold**. Per-threshold CSI is in Table S15.

	SO-10%		SO-20%		SR2x		SR4x	
Method	NRMSE	CSI-m	NRMSE	CSI-m	NRMSE	CSI-m	NRMSE	CSI-m
Filtering								
FlowDAS	1.107 ± 0.721	0.334 ± 0.025	0.709 ± 0.492	0.451 ± 0.029	0.379 ± 0.262	0.652 ± 0.020	0.736 ± 0.463	0.420 ± 0.027
ForcingDAS-AR	**0.357** ± 0.132	**0.540** ± 0.032	**0.268** ± 0.096	**0.626** ± 0.025	**0.200** ± 0.062	**0.692** ± 0.020	**0.356** ± 0.137	**0.535** ± 0.035
Fix-lag Smoothing								
ForcingDAS-Pyr	0.313 ± 0.116	0.580 ± 0.026	0.225 ± 0.077	0.672 ± 0.022	0.167 ± 0.051	0.721 ± 0.018	0.318 ± 0.130	0.572 ± 0.028
Full-sequence Smoothing								
SDA	0.322 ± 0.115	0.575 ± 0.024	0.252 ± 0.089	0.643 ± 0.020	0.218 ± 0.083	0.672 ± 0.017	0.343 ± 0.126	0.553 ± 0.024
ForcingDAS-FS	**0.300** ± 0.105	**0.582** ± 0.026	**0.233** ± 0.077	**0.661** ± 0.021	**0.173** ± 0.051	**0.716** ± 0.017	**0.301** ± 0.101	**0.569** ± 0.029

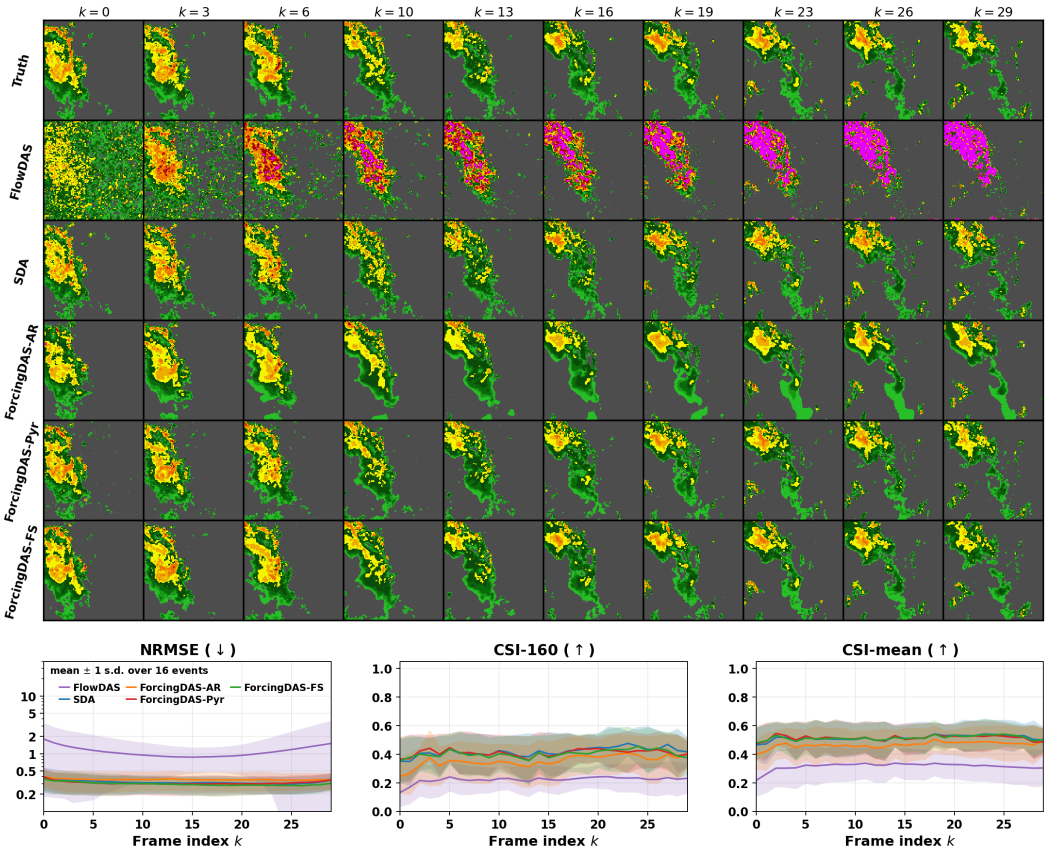

Figure S2: SEVIR VIL assimilation in the *cold-start* protocol (SO-10%, no clean context, $K = 30$). **Top six rows:** ground truth and the estimates of the learned filter FlowDAS, the learned smoother SDA (Rozet & Louppe, 2023), and ForcingDAS in its three regimes (AR, Pyr with a 20-frame lag, FS) on the same event as Fig. 1(b), at $k \in \{0, 3, 6, 10, 13, 16, 19, 23, 26, 29\}$. **Bottom:** per-frame NRMSE (↓, log scale), CSI-160 (↑), and CSI-mean (↑), mean ± 1 s.d. over all 16 events. Numbers are in Table S14.

the event of Fig. 1(b): FlowDAS starts from noise and then saturates the storm core, while SDA and the three ForcingDAS regimes all recover the storm from the first frame. On SEVIR-VIL, VIL is a vertically integrated radar product, a low-dimensional projection of a much higher-dimensional atmospheric state, which favors a joint trajectory prior over a frame-to-frame transition model.

Table S15: SEVIR per-threshold pooled CSI (↑; ± std over bootstrap resamples of the 16 events) in the *cold-start* protocol (no clean context, $K = 30$), grouped by DA regime. ForcingDAS-Pyr uses a 20-frame lag. With-context results are in Table S5.

Obs	Regime	Method	CSI-16	CSI-74	CSI-133	CSI-160	CSI-181	CSI-219	Mean
SO-10%	Filtering	FlowDAS	0.576 ± 0.047	0.559 ± 0.053	0.436 ± 0.031	0.253 ± 0.018	0.143 ± 0.032	0.039 ± 0.029	0.334 ± 0.025
		ForcingDAS-AR	**0.814** ± 0.028	**0.784** ± 0.035	**0.602** ± 0.043	**0.424** ± 0.035	**0.368** ± 0.046	**0.247** ± 0.047	**0.540** ± 0.032
	Fix-lag	ForcingDAS-Pyr	0.837 ± 0.026	0.814 ± 0.030	0.642 ± 0.035	0.478 ± 0.028	0.418 ± 0.038	0.290 ± 0.037	0.580 ± 0.026
	Full-seq.	SDA	0.821 ± 0.029	0.805 ± 0.031	0.628 ± 0.031	0.473 ± 0.024	**0.426** ± 0.035	**0.299** ± 0.040	0.575 ± 0.024
		ForcingDAS-FS	**0.849** ± 0.023	**0.815** ± 0.030	**0.640** ± 0.035	**0.476** ± 0.029	0.420 ± 0.040	0.293 ± 0.039	**0.582** ± 0.026
SO-20%	Filtering	FlowDAS	0.642 ± 0.052	0.661 ± 0.051	0.560 ± 0.038	0.386 ± 0.023	0.288 ± 0.045	0.170 ± 0.064	0.451 ± 0.029
		ForcingDAS-AR	**0.856** ± 0.022	**0.841** ± 0.026	**0.686** ± 0.032	**0.528** ± 0.029	**0.486** ± 0.042	**0.359** ± 0.045	**0.626** ± 0.025
	Fix-lag	ForcingDAS-Pyr	0.884 ± 0.018	0.866 ± 0.022	0.726 ± 0.026	0.589 ± 0.024	0.546 ± 0.036	0.420 ± 0.039	0.672 ± 0.022
	Full-seq.	SDA	0.851 ± 0.025	0.845 ± 0.026	0.690 ± 0.025	0.554 ± 0.022	0.523 ± 0.033	0.393 ± 0.037	0.643 ± 0.020
		ForcingDAS-FS	**0.877** ± 0.019	**0.857** ± 0.023	**0.712** ± 0.027	**0.574** ± 0.024	**0.533** ± 0.036	**0.413** ± 0.037	**0.661** ± 0.021
SR2x	Filtering	FlowDAS	0.795 ± 0.038	0.849 ± 0.025	0.722 ± 0.022	0.588 ± 0.023	0.543 ± 0.037	0.411 ± 0.040	0.652 ± 0.020
		ForcingDAS-AR	**0.880** ± 0.018	**0.875** ± 0.022	**0.745** ± 0.024	**0.617** ± 0.027	**0.581** ± 0.039	**0.453** ± 0.038	**0.692** ± 0.020
	Fix-lag	ForcingDAS-Pyr	0.899 ± 0.016	0.893 ± 0.018	0.773 ± 0.021	0.651 ± 0.023	0.616 ± 0.033	0.492 ± 0.037	0.721 ± 0.018
	Full-seq.	SDA	0.856 ± 0.025	0.860 ± 0.024	0.715 ± 0.023	0.593 ± 0.022	0.567 ± 0.032	0.443 ± 0.032	0.672 ± 0.017
		ForcingDAS-FS	**0.894** ± 0.016	**0.887** ± 0.019	**0.763** ± 0.021	**0.645** ± 0.022	**0.614** ± 0.033	**0.496** ± 0.033	**0.716** ± 0.017
SR4x	Filtering	FlowDAS	0.634 ± 0.048	0.667 ± 0.047	0.536 ± 0.036	0.345 ± 0.026	0.232 ± 0.035	0.103 ± 0.037	0.420 ± 0.027
		ForcingDAS-AR	**0.823** ± 0.028	**0.776** ± 0.037	**0.585** ± 0.047	**0.420** ± 0.038	**0.360** ± 0.048	**0.246** ± 0.048	**0.535** ± 0.035
	Fix-lag	ForcingDAS-Pyr	0.806 ± 0.030	0.815 ± 0.029	0.637 ± 0.034	0.475 ± 0.029	0.409 ± 0.042	0.288 ± 0.047	0.572 ± 0.028
	Full-seq.	SDA	0.803 ± 0.032	0.791 ± 0.032	0.607 ± 0.033	0.450 ± 0.024	0.394 ± 0.035	0.270 ± 0.040	0.553 ± 0.024
		ForcingDAS-FS	**0.833** ± 0.024	**0.811** ± 0.031	**0.629** ± 0.036	**0.467** ± 0.031	**0.403** ± 0.044	**0.274** ± 0.048	**0.569** ± 0.029

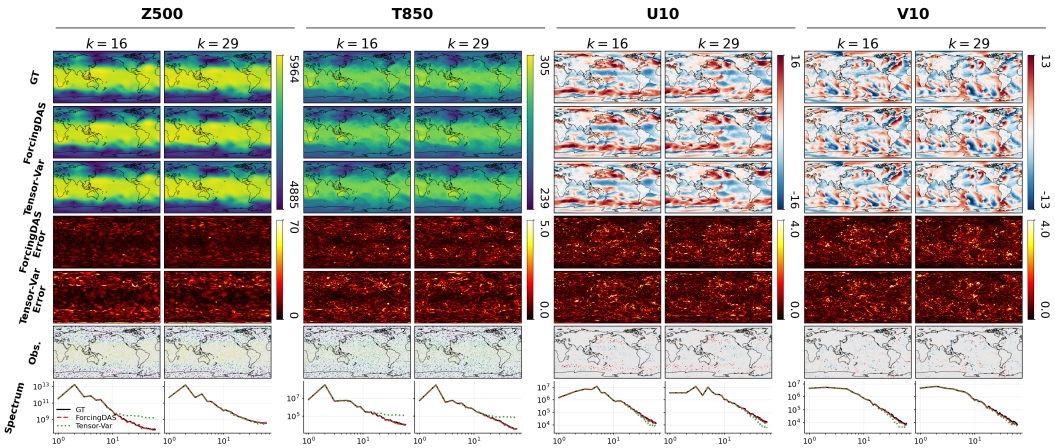

Figure S3: ERA5 SO-10% assimilation in the *cold-start* protocol (no clean context, $K = 30$), on the same trajectory and in the same layout as Fig. 3; ForcingDAS runs as a full-sequence smoother (FS). Numbers are in Table S16.

Per-event CSI is noisy at high thresholds. Under a per-event mean instead of pooling, SDA is level with or slightly ahead of ForcingDAS-FS on CSI-160 for cold-start SO-10% and SR4x (by at most 0.013), while ForcingDAS-FS remains ahead on the other two operators; pooled, ForcingDAS-FS is ahead on all four.

S4.3 Global Atmospheric State Estimation (ERA5)

Table S16 reports cold-start ERA5 results (no clean context, $K = 30$) on the same 16 trajectories and shared observations as the main text. A cold-start run begins with no clean state, so we treat the first assimilated frame as a spin-up period and exclude it from every metric, identically for all methods; scores cover frames 1–29.

Without a clean context, ForcingDAS-AR again improves on 3D-Var on all four variables and both metrics, on every trajectory. The other learned filter, FlowDAS, is the method that suffers most from losing the clean prefix: its NRMSE rises from $0.035/0.070/0.206/0.249$ with context (Table 2) to

Table S16: ERA5 SO-10% assimilation, *cold-start* (no clean context, $K = 30$, frame 0 excluded as spin-up): latitude-weighted NRMSE (↓, z-score units) and ACC (↑) per variable. Mean ± std over 16 held-out trajectories from 2016. Best in *Filtering* and best in *Full-sequence Smoothing* per column in **bold**.

	Z500		T850		U10		V10	
Method	NRMSE	ACC	NRMSE	ACC	NRMSE	ACC	NRMSE	ACC
Filtering								
3D-Var	0.058 ± 0.007	0.971 ± 0.009	0.098 ± 0.007	0.896 ± 0.030	0.380 ± 0.008	0.853 ± 0.010	0.417 ± 0.010	0.875 ± 0.007
FlowDAS	0.084 ± 0.004	0.860 ± 0.026	0.143 ± 0.004	0.699 ± 0.046	0.273 ± 0.006	0.915 ± 0.005	0.320 ± 0.006	0.921 ± 0.007
ForcingDAS-AR	**0.027** ± 0.001	**0.994** ± 0.001	**0.056** ± 0.001	**0.968** ± 0.004	**0.165** ± 0.003	**0.973** ± 0.002	**0.199** ± 0.003	**0.972** ± 0.002
Fix-lag Smoothing								
ForcingDAS-Pyr	0.025 ± 0.003	0.995 ± 0.002	0.050 ± 0.002	0.974 ± 0.005	0.153 ± 0.004	0.977 ± 0.001	0.187 ± 0.004	0.975 ± 0.002
Full-sequence Smoothing								
Tensor-Var	0.035 ± 0.001	0.990 ± 0.001	0.064 ± 0.002	0.956 ± 0.006	0.156 ± 0.003	0.976 ± 0.001	**0.186** ± 0.003	**0.976** ± 0.002
ForcingDAS-FS	**0.022** ± 0.000	**0.996** ± 0.000	**0.050** ± 0.001	**0.974** ± 0.004	**0.154** ± 0.003	**0.977** ± 0.001	0.187 ± 0.003	0.975 ± 0.002

$0.084/0.143/0.273/0.320$, a factor of 2.4 on Z500 and 2.0 on T850, against 1.03 on every variable for ForcingDAS-AR. That leaves it behind even the classical 3D-Var on Z500 (0.084 vs. 0.058) and T850 (0.143 vs. 0.098), although it stays well ahead on the surface winds (0.273 vs. 0.380, 0.320 vs. 0.417); a per-step transition model has no clean state to propagate at the start of a cold-start run, whereas a joint trajectory prior conditions on the whole window. Among the smoothers, ForcingDAS-FS improves on Tensor-Var on Z500 (0.022 vs. 0.035), T850 (0.050 vs. 0.064), and U10 (0.154 vs. 0.156), in each case on all 16 trajectories. V10 is the exception: Tensor-Var is slightly better (0.186 vs. 0.187 NRMSE, better on 14 of 16 trajectories). The Z500 spread of ForcingDAS-Pyr comes from a few trajectories whose first frames after the spin-up frame are still unsettled. Removing the clean context costs ForcingDAS little: FS changes from $0.022/0.049/0.149/0.182$ with context (Table 2) to $0.022/0.050/0.154/0.187$. The two protocols score different frames (24 vs. 29), so this comparison is indicative only. As on the other benchmarks, the four ERA5 channels are a small slice of a much higher-dimensional atmospheric state, and a joint trajectory prior can use past frames to infer what the current frame does not show.

S5 BASELINE IMPLEMENTATION DETAILS

This appendix describes the ERA5 baselines (3D-Var, FlowDAS, Tensor-Var; §4.3) and the Navier–Stokes ensemble baselines (EnKF, EnKS; §4.1). The learned NS and SEVIR baselines are described in Appendices S3.1.5 and S3.2.5. Every baseline assimilates the same stored observations as ForcingDAS and is scored by the same evaluation code.

S5.1 ERA5: SHARED SETUP

Background state. In the cold-start setting the background state is the climatological mean, which is $\boldsymbol{x}_b = \mathbf{0}$ in normalized space. With context, the 6 clean frames are their own background, and the last context frame is the first-pass background for every assimilated frame.

Background error covariance. We model the background error covariance $\boldsymbol{B}$ using the *control variable transform* (CVT) (Courtier et al., 1998), the standard operational approach for introducing spatial correlations into $\boldsymbol{B}$ without forming the full $\mathbb{R}^{d\times d}$ matrix. A control variable $\boldsymbol{v}$ is optimized instead of the state increment directly; the state is recovered via $\boldsymbol{x} = \boldsymbol{x}_b + \sigma_b\, K * \boldsymbol{v}$, where K is a Gaussian smoothing kernel applied per channel in the Fourier domain and $\sigma_b = 1$. This implicitly specifies $\boldsymbol{B} = \sigma_b^2\, \boldsymbol{K}\boldsymbol{K}^\top$, a spatially correlated covariance whose off-diagonal structure allows observations to inform nearby unobserved grid points through the correlation length scale of K. We use per-variable isotropic Gaussian kernels with length scales $\ell = (8, 6, 5, 5)$ grid points for (Z500, T850, U10, V10) respectively, reflecting the decreasing spatial coherence from synoptic-scale geopotential to sub-synoptic surface winds.

Observation model. Observations follow the shared binary spatial mask $\boldsymbol{m} \in \{0,1\}^{H\times W}$ (identical across all C channels):

$$\boldsymbol{y}_{k,c} = \boldsymbol{x}_{k,c} \odot \boldsymbol{m} + \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \sigma_y^2\, \boldsymbol{I}), \tag{S17}$$

where $\sigma_y = 0.05$ in normalized space. The baselines read the stored observations rather than drawing their own noise, so they assimilate exactly the values ForcingDAS assimilates.

Spatial grid. All data are cropped from 240×121 to 240×120 by dropping the North-Pole row, exactly as for ForcingDAS.

S5.2 3D-VAR (FILTERING)

3D-Var performs an independent analysis at each time step. Using the CVT parameterization from the shared setup above, we optimize the control variable $\boldsymbol{v}_k$ at each time step:

$$J_{\mathrm{3D}}(\boldsymbol{v}_k) = \frac{1}{2}\|\boldsymbol{v}_k\|^2 + \frac{1}{2\sigma_y^2}\sum_{c=1}^{C} \|\boldsymbol{y}_{k,c}[\boldsymbol{m}] - \boldsymbol{x}_{k,c}[\boldsymbol{m}]\|^2, \quad \boldsymbol{x}_k = \boldsymbol{x}_{b,k} + \sigma_b\, K_c * \boldsymbol{v}_{k,c}, \tag{S18}$$

where K_c is the per-channel Gaussian kernel and $[\boldsymbol{m}]$ denotes restriction to observed grid points. The background penalty $\frac{1}{2}\|\boldsymbol{v}_k\|^2$ in control space is equivalent to $\frac{1}{2}(\boldsymbol{x}_k - \boldsymbol{x}_{b,k})^\top \boldsymbol{B}^{-1}(\boldsymbol{x}_k - \boldsymbol{x}_{b,k})$ in state space.

Cycling. Although each frame is analyzed independently (no temporal coupling in the cost function), we cycle the background sequentially in two passes: the first pass uses the background of the shared setup above; the second uses persistence of the first-pass analysis, $\boldsymbol{x}_{b,k} = \boldsymbol{x}_{a,k-1}$. This propagates information across time via the background field, allowing observations at time $k-1$ to inform the prior at time k.

Optimization. We use L-BFGS with 80 outer iterations (each with 50 inner line-search steps, history size 50, strong Wolfe conditions). The Gaussian kernel K_c is applied in the Fourier domain via `rfft2`/`irfft2`, making each L-BFGS step $O(CHW\log(HW))$. For efficiency, all frames are optimized jointly in a single batch; this is equivalent to per-frame optimization since the cost decomposes across time.

Interpretation. 3D-Var with a Gaussian CVT produces a spatially smoothed analysis: the correlation kernel fills unobserved locations by interpolating from nearby observed pixels. This works reasonably for the smooth synoptic-scale Z500 field (ACC 0.964 with context), but the Gaussian smoothing removes structure at scales shorter than the correlation length. For the surface winds, whose spatial scales approach the kernel width, its ACC drops to 0.79–0.81 and its NRMSE is about $3\times$ that of ForcingDAS-AR. Since 3D-Var has no dynamical model, the clean context helps it only through the persistence background, and its with-context errors are in fact higher than its cold-start errors (the two protocols score different frames; Tables 2 and S16).

S5.3 FlowDAS (Filtering)

FlowDAS (Chen et al., 2025) learns a one-step stochastic-interpolant forecast model and assimilates each frame by guided sampling from it. On ERA5 it conditions on the 6 clean context frames and rolls out 24 autoregressive steps, assimilating the shared observation of each frame. In the cold-start protocol the conditioning window is replaced by Gaussian noise and all 30 frames are assimilated, using the same checkpoint and sampler. The drift model is a U-Net with 128 base channels and multipliers $1{-}2{-}2{-}2$, 30.2M parameters, taking 28 input channels (6 conditioning frames $\times$ 4 variables, plus the noisy frame) and predicting 4; it was trained for $300{,}000$ steps on the same `era5_full` 1979–2015 training set and normalization as ForcingDAS. Sampling uses 500 Euler–Maruyama steps per frame with 25 Monte-Carlo guidance samples and guidance scale 1.0, selected on the 4 held-out tuning windows.

S5.4 Tensor-Var (Smoothing)

Tensor-Var (Yang et al., 2025) has four components: a CNN encoder K_S that maps a state to a 512-dimensional feature, and a matching decoder K_S^+ with skip connections; a linear feature dynamics matrix $C \in \mathbb{R}^{512\times512}$, estimated by ridge regression on encoded consecutive states; a Restormer-style inverse-observation network K_O that maps a stack of 10 observation frames to a state estimate; and a 4D-Var in feature space with a soft dynamics penalty, solved in closed form over non-overlapping 3-frame windows and decoded with K_S^+. We re-implemented it from the public code and retrained it on our setting: the four variables on the 240×120 grid (replicate-padded to 240×128 internally), the 1979–2015 training years with the same normalization, and the SO-10% observations with $\sigma_y = 0.05$. Training takes 100 epochs for K_S, K_S^+, and C, then 100 epochs for K_O with the others frozen, on 4 A40 GPUs ($\approx$113 GPU-hours in total). The model has 33.6M parameters (32.3M for K_S and K_S^+, 1.2M for K_O). We use the published feature-space 4D-Var weights ($B = 0.1$, $R = 1$, $Q = 0.1$). Inference takes about 1 s per 30-frame trajectory on one GPU. The observation window of K_O for frame k is the ten most recent observations $\boldsymbol{y}_{k-9}, \ldots, \boldsymbol{y}_k$, including the current one; with context, the context slots of that window hold the masked clean states. One detail favors Tensor-Var: K_O is trained on the same fixed observation mask used at evaluation, so the model is specific to that mask, whereas ForcingDAS never sees the mask during training.

Fairness considerations. The comparisons differ in what each method is given. 3D-Var has a fixed Gaussian spatial prior and no dynamics. ForcingDAS learns its trajectory prior from the 1979–2015 training years and never sees the observation mask during training; FlowDAS likewise learns its forecast model from ERA5 training data and applies the observations only at inference. Tensor-Var learns its dynamics and its inverse-observation network from the same training years, but its inverse network is trained on the evaluation mask. All methods assimilate identical observations and are scored by the same code on the same frames.

S5.5 Ensemble Kalman Filter and Smoother Baselines for Navier–Stokes

We implement classical ensemble data assimilation baselines for the 2D Navier–Stokes experiment (§4.1): a stochastic Ensemble Kalman Filter (EnKF) (Evensen, 2003) and two variants of the Ensemble Kalman Smoother (EnKS): a fixed-lag variant (EnKS-FL) and a full-sequence variant (EnKS-FS). All three share the same forward model, observation operator, and core analysis update; they differ only in whether and how future observations are used to refine past state estimates.

S5.5.1 FORWARD MODEL

The EnKF/EnKS forecast step integrates the same stochastic vorticity equation (S7) with the same forcing structure (S8) used to generate the truth data (Appendix S3.1.1). The forecast model uses a pseudo-spectral discretization on a 128×128 grid with a semi-implicit Euler–Maruyama time integrator at $\Delta t_{\text{sim}} = 10^{-3}$: linear terms (viscosity and drag) are treated implicitly, eliminating CFL restrictions from diffusion; the nonlinear advection term is treated explicitly with $\frac{2}{3}$-rule dealiasing; stochastic forcing is discretized via the standard Euler–Maruyama scheme (the appropriate choice for SDEs, unlike higher-order deterministic integrators such as RK4). We use $\Delta t_{\text{sim}} = 10^{-3}$ because the scheme is not stable at 2×10^{-3} over a full observation interval: integrated freely for $\Delta t_{\text{obs}} = 1.0$, the explicit advection term overflows within about 7 cycles. The smaller step is also closer to the $\Delta t = 10^{-4}$ at which the truth was generated.

Model error. As described in Appendix S3.1.1, the truth data were generated at 256×256 resolution ($\Delta t = 10^{-4}$) and bilinearly downsampled to 128×128 (Chen et al., 2024c). The EnKF forecast runs natively at 128×128, introducing an unavoidable resolution-dependent model error: the downsampled dynamics differ from dynamics simulated natively at the coarser resolution due to missing sub-grid energy transfer. The exact PDE parameters (viscosity, drag, forcing structure) are known and matched.

S5.5.2 ENKF CONFIGURATION

Ensemble. We use $N_e = 100$ ensemble members, each initialized as a random snapshot drawn from the training trajectories only, so that no member is taken from the trajectory being assimilated. The EnKF does not use the clean context frames: it starts from this ensemble and assimilates from frame 10 onward.

Forecast. Each ensemble member is integrated forward by $\Delta t_{\text{obs}} = 1.0$ time units (1000 steps at $\Delta t_{\text{sim}} = 10^{-3}$), the spacing between consecutive stored frames, using the stochastic NS solver with independent noise realizations per member. This ensures the ensemble spread captures the inherent forecast uncertainty from the unknown stochastic forcing.

Analysis. We use the batch stochastic EnKF formulation. Let $\boldsymbol{X}^f \in \mathbb{R}^{N_e \times n}$ denote the forecast ensemble ($n = 128^2 = 16{,}384$), $\boldsymbol{H}$ the observation operator that selects observed grid points, and $\boldsymbol{R} = \sigma_y^2 \boldsymbol{I}_m$ the observation error covariance. The Kalman gain is computed as

$$\boldsymbol{K} = \boldsymbol{P}^f \boldsymbol{H}^\top \left(\boldsymbol{H} \boldsymbol{P}^f \boldsymbol{H}^\top + \boldsymbol{R} \right)^{-1}, \tag{S19}$$

where $\boldsymbol{P}^f$ is the sample covariance from the ensemble. Each member is updated with perturbed observations (Burgers et al., 1998): $\boldsymbol{\omega}_i^a = \boldsymbol{\omega}_i^f + \boldsymbol{K}(\tilde{\boldsymbol{y}}_i - \boldsymbol{H}\boldsymbol{\omega}_i^f)$, where $\tilde{\boldsymbol{y}}_i = \boldsymbol{y} + \boldsymbol{\epsilon}_i$, $\boldsymbol{\epsilon}_i \sim \mathcal{N}(\boldsymbol{0},\, \boldsymbol{R})$, with the perturbations mean-centered across the ensemble. The analysis ensemble mean is the state estimate at each observation time. Before the forecast–analysis cycling begins, one analysis step is performed at t_0 using sparse observations of the initial frame, so that the output frame count matches the data-driven methods.

Localization. Covariance localization via the Gaspari–Cohn function (Gaspari & Cohn, 1999) is applied element-wise to both $\boldsymbol{P}^f \boldsymbol{H}^\top$ and $\boldsymbol{H} \boldsymbol{P}^f \boldsymbol{H}^\top$, with a cutoff radius of 15 grid points. Periodic boundary conditions are respected in the distance computation.

Inflation. Multiplicative prior inflation is applied before each analysis step: $\boldsymbol{\omega}_i \leftarrow \bar{\boldsymbol{\omega}} + \lambda(\boldsymbol{\omega}_i - \bar{\boldsymbol{\omega}})$ with $\lambda = 1.10$, to counteract ensemble collapse from undersampling.

S5.5.3 ENKS ALGORITHM

Both EnKS variants extend the EnKF by updating past state estimates using future observations, via an augmented-state approach.

Mechanism. At each analysis step, the current ensemble is concatenated with ensembles stored from past observation times into an augmented state vector:

$$\boldsymbol{x}_{\text{aug}} = \begin{bmatrix} \boldsymbol{x}_{t-L} \\ \boldsymbol{x}_{t-L+1} \\ \vdots \\ \boldsymbol{x}_t \end{bmatrix} \in \mathbb{R}^{(L+1)\cdot n}, \tag{S20}$$

where L is the lag and $n = 128^2 = 16{,}384$ is the state dimension. The observation operator $\boldsymbol{H}$ acts only on the current-time block (last n components). The standard EnKF analysis is applied to the full augmented state; because the ensemble sample covariance naturally contains cross-time correlations, the Kalman update propagates observation information backward to the earlier blocks. After analysis, the augmented state is split back into individual time-step ensembles. Localization is applied by tiling the spatial Gaspari–Cohn matrix across all time blocks in the augmented state.

EnKS-FL (fixed-lag, $L = 10$). The buffer holds the most recent $L + 1$ ensembles (current + L past). Each state estimate is refined by up to L future observations before being finalized and removed from the buffer. We use $L = 10$, the same window as ForcingDAS-Pyr, so that the two fixed-lag smoothers are compared over the same lag. It is evaluated in both protocols.

EnKS-FS (full-sequence). The buffer holds the ensembles of the entire assimilation window, and each state is refined by all subsequent observations with no early finalization. With $N_e = 100$ ensemble members and an augmented state of up to $30 \times 16{,}384 = 491{,}520$ dimensions in the cold start, the sample covariance is severely rank-deficient, which limits the smoother's effectiveness and can degrade results compared to smaller windows.

S5.5.4 Observation setup

EnKF, EnKS-FL, and EnKS-FS assimilate the shared observations of Appendix S3.1.2: the same 795 observed grid points (5% of the 128×128 grid, fixed across time steps) and the same noise realization, converted to raw vorticity, so that $\boldsymbol{R} = \sigma_y^2 \boldsymbol{I}$ with $\sigma_y = 1.829$. Consecutive observations are $\Delta t_{\text{obs}} = 1.0$ time units apart.

S5.5.5 Protocol

The 16 evaluation trajectories are assimilated independently, and metrics are averaged across trajectories. In the with-context protocol ($K = 50$), the first 10 frames (indices 0–9) are the context frames of the other methods and are not scored; assimilation runs for 39 cycles over frames 10–49 (40 output frames including the initial analysis). In the cold-start protocol ($K = 30$), assimilation covers all 30 frames (29 cycles, no frames skipped). EnKS-FS uses a lag spanning the whole assimilation window ($L = 39$ with context, $L = 29$ in the cold start) with no temporal taper, so every state is updated by all later observations.

S6 LIMITATIONS AND FUTURE WORK

This appendix expands the brief discussion in the main-text Limitations and Conclusion (§5), and closes with directions for future work.

S6.1 LIMITATIONS

Causal-only smoothing. Because the denoising network uses causal temporal attention, future observations can only refine past states through the backward gradient of the observation loss; the forward pass carries no future-to-past signal (Appendix S2.1). A model with bidirectional attention could also pass this information through the forward pass, but would no longer support causal filtering from the same weights. Removing this restriction, e.g., through a hybrid causal/bidirectional backbone, while preserving the single-model filtering–smoothing interpolation is an open direction.

Compute at long horizons. Pyramid and full-sequence scheduling denoise multiple frames jointly at every DDIM step, with memory and compute that scale with the active window, $\min(K, \lceil T/u \rceil)$ frames for the pyramid schedule and the whole horizon for the full-sequence schedule. For very long trajectories, this cost can become prohibitive relative to the purely autoregressive regime, and blockwise or hierarchical schedules may be needed.

Simulated observations. All observations in our experiments are synthetic: a fixed random pixel mask or a down-sampling operator applied to the reference field, plus i.i.d. Gaussian noise. Real observing systems have irregular and time-varying geometry, instrument-specific and often nonlinear operators (e.g., satellite radiances), and correlated, non-Gaussian errors. On ERA5 in particular, the reference is itself a reanalysis, so we assimilate synthetic observations of a model-consistent state rather than raw measurements. Assimilating real instrument observations is left to future work.

Resolution and variables on ERA5. Our ERA5 study operates at 1.5° with four variables, sufficient to compare against the 3D-Var, FlowDAS, and Tensor-Var baselines on the same footing but coarser than operational weather systems (0.25° with 60+ variables). Scaling the framework to native WeatherBench2-style resolutions will likely require a stronger diffusion backbone, possibly a latent-diffusion variant, and improved memory handling.

Probabilistic calibration. We evaluate a single estimated trajectory per run; while the per-frame noise levels provide an *internal* uncertainty representation, we have not systematically evaluated the calibration of ensemble statistics obtained by running multiple seeds of the reverse process. A careful probabilistic evaluation with spread–error diagnostics is left to future work.

S6.2 FUTURE WORK

Beyond the limitations above, we see three concrete directions for extending ForcingDAS.

Physics constraints and reward-driven fine-tuning. The current trajectory prior is purely data-driven and may produce trajectories that are statistically plausible yet inconsistent with the conservation laws of the underlying PDE. Adding soft physics-residual penalties during training (Bastek et al., 2025), using diffusion-style multi-step refinement to stabilize long PDE rollouts (Lippe et al., 2023), and projecting sampling updates onto a constraint manifold at inference (He et al., 2024) are all natural ways to close this gap. A complementary direction is reward-driven fine-tuning of the trajectory prior on assimilation-specific objectives such as long-horizon NRMSE or spectral fidelity, via policy-gradient reinforcement learning (Black et al., 2024) or preference-based alignment (Wallace et al., 2024).

Distillation, latent diffusion, and representation learning. Pyramid and full-sequence sampling are expensive at long horizons. Distilling the reverse process into a few-step student, and lifting the trajectory prior into a latent space, would both reduce cost and provide a clean setting in which to study the geometry of the learned representation (Li et al., 2025a; Zhang et al., 2026; Li et al., 2026a).

Spatio-temporal medical imaging. ForcingDAS is, at its core, a joint denoising framework for sequences of partial observations of an evolving state, which matches the structure of many dynamic medical imaging problems. Natural targets include accelerated cardiac cine MRI (Qiu et al., 2024), motion-compensated 4D cone-beam CT for radiotherapy (Zhang et al., 2023), SPECT (Li et al., 2025b; Jia et al., 2023; Singh et al., 2024b), and dynamic PET imaging (Gong et al., 2022). The trajectory-prior and scheduling-matrix machinery transfer directly; only the observation operator (k-space, projection geometries, Poisson noise) is modality-specific.